

Edda Klipp, Wolfram Liebermeister,
Christoph Wierling and Axel Kowald

Systems Biology

A Textbook

Second Edition



Edda Klipp
Wolfram Liebermeister
Christoph Wierling
Axel Kowald

Systems Biology

Edda Klipp
Wolfram Liebermeister
Christoph Wierling
Axel Kowald

Systems Biology

A Textbook

Second, Completely Revised
and Enlarged Edition

WILEY-VCH
Verlag GmbH & Co. KGaA

Authors

Prof. Dr. h.c. Edda Klipp

Theoretical Biophysics
Humboldt-Universität zu Berlin
Invalidenstr. 42
10115 Berlin
Germany

Dr. Wolfram Liebermeister

Institute of Biochemistry
Charité - Universitätsmedizin Berlin
Charitéplatz 1
10117 Berlin
Germany

Dr. Christoph Wierling

Alacris Theranostics GmbH
Fabeckstr. 60-62
14195 Berlin
Germany

and

Max Planck Institute for Molecular Genetics
Ihnestr. 63-73
14195 Berlin
Germany

Dr. Axel Kowald

Theoretical Biophysics
Humboldt University Berlin
Invalidenstr. 42
10115 Berlin
Germany

Cover

Cover design by Wolfram Liebermeister. The cover picture was provided with kind permission by Jörg Bernhardt.

All books published by Wiley-VCH are carefully produced. Nevertheless, authors, editors, and publisher do not warrant the information contained in these books, including this book, to be free of errors. Readers are advised to keep in mind that statements, data, illustrations, procedural details or other items may inadvertently be inaccurate.

Library of Congress Card No.: applied for

British Library Cataloguing-in-Publication Data

A catalogue record for this book is available from the British Library.

Bibliographic information published by the Deutsche Nationalbibliothek

The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available on the Internet at <<http://dnb.d-nb.de>>.

© 2016 Wiley-VCH Verlag GmbH & Co. KGaA, Boschstr. 12, 69469 Weinheim, Germany

All rights reserved (including those of translation into other languages). No part of this book may be reproduced in any form – by photoprinting, microfilm, or any other means – nor transmitted or translated into a machine language without written permission from the publishers. Registered names, trademarks, etc. used in this book, even when not specifically marked as such, are not to be considered unprotected by law.

Print ISBN: 978-3-527-33636-4

ePDF ISBN: 978-3-527-67566-1

ePub ISBN: 978-3-527-67567-8

Mobi ISBN: 978-3-527-67568-5

Typesetting Thomson Digital, Noida, India

Printed on acid-free paper

Contents

Preface xi

Guide to Different Topics of the Book xiii

About the Authors xv

Part One Introduction to Systems Biology 1

1 Introduction 3

1.1 Biology in Time and Space 3

1.2 Models and Modeling 4

1.2.1 What Is a Model? 4

1.2.2 Purpose and Adequateness of Models 5

1.2.3 Advantages of Computational Modeling 5

1.3 Basic Notions for Computational Models 6

1.3.1 Model Scope 6

1.3.2 Model Statements 6

1.3.3 System State 6

1.3.4 Variables, Parameters, and Constants 6

1.3.5 Model Behavior 7

1.3.6 Model Classification 7

1.3.7 Steady States 7

1.3.8 Model Assignment Is Not Unique 7

1.4 Networks 8

1.5 Data Integration 8

1.6 Standards 9

1.7 Model Organisms 9

1.7.1 *Escherichia coli* 9

1.7.2 *Saccharomyces cerevisiae* 11

1.7.3 *Caenorhabditis elegans* 11

1.7.4 *Drosophila melanogaster* 11

1.7.5 *Mus musculus* 12

References 12

Further Reading 14

2 Modeling of Biochemical Systems 15

2.1 Overview of Common Modeling Approaches for Biochemical Systems 15

2.2 ODE Systems for Biochemical Networks 17

2.2.1 Basic Components of ODE Models 18

2.2.2 Illustrative Examples of ODE Models 18

References 21

Further Reading 21

3 Structural Modeling and Analysis of Biochemical Networks 23

3.1 Structural Analysis of Biochemical Systems 24

3.1.1 System Equations 24

3.1.2 Information Encoded in the Stoichiometric Matrix N 25

3.1.3 The Flux Cone 27

3.1.4 Elementary Flux Modes and Extreme Pathways 27

3.1.5 Conservation Relations – Null Space of N^T 29

3.2 Constraint-Based Flux Optimization 30

3.2.1 Flux Balance Analysis 31

3.2.2 Geometric Interpretation of Flux Balance Analysis 31

3.2.3 Thermodynamic Constraints 31

3.2.4 Applications and Tests of the Flux Optimization Paradigm 32

3.2.5 Extensions of Flux Balance Analysis 33

Exercises 35

References 36

Further Reading 37

4 Kinetic Models of Biochemical Networks: Introduction 39

4.1 Reaction Kinetics and Thermodynamics 39

4.1.1 Kinetic Modeling of Enzymatic Reactions 39

4.1.2 The Law of Mass Action 40

4.1.3 Reaction Thermodynamics 40

4.1.4 Michaelis–Menten Kinetics 42

4.1.5 Regulation of Enzyme Activity by Effectors 44

4.1.6 Generalized Mass Action Kinetics 48

4.1.7 Approximate Kinetic Formats 48

4.1.8 Convenience Kinetics and Modular Rate Laws 49

4.2 Metabolic Control Analysis 50

4.2.1 The Coefficients of Control Analysis 51

4.2.2	Theorems of Metabolic Control Theory	53	6.4	Coupled Systems and Emergent Behavior	110
4.2.3	Matrix Expressions for Control Coefficients	55	6.4.1	Modeling of Coupled Systems	111
4.2.4	Upper Glycolysis as Realistic Model Example	58	6.4.2	Combining Rate Laws into Models	113
4.2.5	Time-Dependent Response Coefficients	59	6.4.3	Modular Response Analysis	113
	Exercises	61	6.4.4	Emergent Behavior in Coupled Systems	114
	References	61	6.4.5	Causal Interactions and Global Behavior	115
	Further Reading	62		Exercises	116
5	Data Formats, Simulation Techniques, and Modeling Tools	63		References	117
5.1	Simulation Techniques and Tools	63		Further Reading	119
5.1.1	Differential Equations	63	7	Discrete, Stochastic, and Spatial Models	121
5.1.2	Stochastic Simulations	64	7.1	Discrete Models	122
5.1.3	Simulation Tools	65	7.1.1	Boolean Networks	122
5.2	Standards and Formats for Systems Biology	72	7.1.2	Petri Nets	124
5.2.1	Systems Biology Markup Language	72	7.2	Stochastic Modeling of Biochemical Reactions	127
5.2.2	BioPAX	74	7.2.1	Chance in Biochemical Reaction Systems	127
5.2.3	Systems Biology Graphical Notation	74	7.2.2	The Chemical Master Equation	129
5.3	Data Resources for Modeling of Cellular Reaction Systems	75	7.2.3	Stochastic Simulation	129
5.3.1	General-Purpose Databases	75	7.2.4	Chemical Langevin Equation and Chemical Noise	130
5.3.2	Pathway Databases	76	7.2.5	Dynamic Fluctuations	132
5.3.3	Model Databases	77	7.2.6	From Stochastic to Deterministic Modeling	133
5.4	Sustainable Modeling and Model Semantics	78	7.3	Spatial Models	133
5.4.1	Standards for Systems Biology Models	78	7.3.1	Types of Spatial Models	134
5.4.2	Model Semantics and Model Comparison	78	7.3.2	Compartment Models	135
5.4.3	Model Combination	80	7.3.3	Reaction-Diffusion Systems	136
5.4.4	Model Validity	82	7.3.4	Robust Pattern Formation in Embryonic Development	138
	References	83	7.3.5	Spontaneous Pattern Formation	139
	Further Reading	85	7.3.6	Linear Stability Analysis of the Activator-Inhibitor Model	140
6	Model Fitting, Reduction, and Coupling	87		Exercises	142
6.1	Parameter Estimation	88		References	143
6.1.1	Regression, Estimators, and Maximal Likelihood	88		Further Reading	144
6.1.2	Parameter Identifiability	90	8	Network Structure, Dynamics, and Function	145
6.1.3	Bootstrapping	91	8.1	Structure of Biochemical Networks	146
6.1.4	Bayesian Parameter Estimation	92	8.1.1	Random Graphs	147
6.1.5	Probability Distributions for Rate Constants	94	8.1.2	Scale-Free Networks	148
6.1.6	Optimization Methods	97	8.1.3	Connectivity and Node Distances	149
6.2	Model Selection	99	8.1.4	Network Motifs and Significance Tests	150
6.2.1	What Is a Good Model?	99	8.1.5	Explanations for Network Structures	151
6.2.2	The Problem of Model Selection	100	8.2	Regulation Networks and Network Motifs	152
6.2.3	Likelihood Ratio Test	102	8.2.1	Structure of Transcription Networks	153
6.2.4	Selection Criteria	102	8.2.2	Regulation Edges and Their Steady-State Response	156
6.2.5	Bayesian Model Selection	103	8.2.3	Negative Feedback	156
6.3	Model Reduction	104	8.2.4	Adaptation Motif	157
6.3.1	Model Simplification	104	8.2.5	Feed-Forward Loops	158
6.3.2	Reduction of Fast Processes	105			
6.3.3	Quasi-Equilibrium and Quasi-Steady State	107			
6.3.4	Global Model Reduction	108			

8.3 Modularity and Gene Functions 160	10 Variability, Robustness, and Information 209
8.3.1 Cell Functions Are Reflected in Structure, Dynamics, Regulation, and Genetics 160	10.1 Variability and Biochemical Models 210
8.3.2 Metabolics Pathways and Elementary Modes 162	10.1.1 Variability and Uncertainty Analysis 210
8.3.3 Epistasis Can Indicate Functional Modules 163	10.1.2 Flux Sampling 212
8.3.4 Evolution of Function and Modules 163	10.1.3 Elasticity Sampling 213
8.3.5 Independent Systems as a Tacit Model Assumption 165	10.1.4 Propagation of Parameter Variability in Kinetic Models 214
8.3.6 Modularity and Biological Function Are Conceptual Abstractions 165 Exercises 166	10.1.5 Models with Parameter Fluctuations 216
References 167	10.2 Robustness Mechanisms and Scaling Laws 217
Further Reading 169	10.2.1 Robustness in Biochemical Systems 218
9 Gene Expression Models 171	10.2.2 Robustness by Backup Elements 219
9.1 Mechanisms of Gene Expression Regulation 171	10.2.3 Feedback Control 219
9.1.1 Transcription Factor-Initiated Gene Regulation 171	10.2.4 Perfect Robustness by Structure 222
9.1.2 General Promoter Structure 173	10.2.5 Scaling Laws 224
9.1.3 Prediction and Analysis of Promoter Elements 174	10.2.6 Time Scaling, Summation Laws, and Robustness 227
9.1.4 Posttranscriptional Regulation through microRNAs 176	10.2.7 The Role of Robustness in Evolution and Modeling 228
9.2 Dynamic Models of Gene Regulation 180	10.3 Adaptation and Exploration Strategies 229
9.2.1 A Basic Model of Gene Expression and Regulation 180	10.3.1 Information Transmission in Signaling Pathways 230
9.2.2 Natural and Synthetic Gene Regulatory Networks 183	10.3.2 Adaptation and Fold-Change Detection 230
9.2.3 Gene Expression Modeling with Stochastic Equations 186	10.3.3 Two Adaptation Mechanisms: Sensing and Random Switching 231
9.3 Gene Regulation Functions 187	10.3.4 Shannon Information and the Value of Information 232
9.3.1 The Lac Operon in <i>E. coli</i> 187	10.3.5 Metabolic Shifts and Anticipation 233
9.3.2 Gene Regulation Functions Derived from Equilibrium Binding 188	10.3.6 Exploration Strategies 234 Exercises 236
9.3.3 Thermodynamic Models of Promoter Occupancy 189	References 237
9.3.4 Gene Regulation Function of the Lac Promoter 191	Further Reading 239
9.3.5 Inferring Transcription Factor Activities from Transcription Data 192	11 Optimality and Evolution 241
9.3.6 Network Component Analysis 194	11.1 Optimality in Systems Biology Models 243
9.3.7 Correspondences between mRNA and Protein Levels 196	11.1.1 Mathematical Concepts for Optimality and Compromise 245
9.4 Fluctuations in Gene Expression 196	11.1.2 Metabolism Is Shaped by Optimality 248
9.4.1 Stochastic Model of Transcription and Translation 197	11.1.3 Optimality Approaches in Metabolic Modeling 250
9.4.2 Intrinsic and Extrinsic Variability 200	11.1.4 Metabolic Strategies 252
9.4.3 Temporal Fluctuations in Gene Cascades 202	11.1.5 Optimal Metabolic Adaptation 253
Exercises 203	11.2 Optimal Enzyme Concentrations 255
References 205	11.2.1 Optimization of Catalytic Properties of Single Enzymes 255
Further Reading 207	11.2.2 Optimal Distribution of Enzyme Concentrations in a Metabolic Pathway 257
	11.2.3 Temporal Transcription Programs 259
	11.3 Evolution and Self-Organization 261
	11.3.1 Introduction 261
	11.3.2 Selection Equations for Biological Macromolecules 263
	11.3.3 The Quasispecies Model: Self-Replication with Mutations 265

11.3.4	The Hypercycle	267	13.2.1	Chemical Bonds and Forces Important in Biological Molecules	336
11.3.5	Other Mathematical Models of Evolution: Spin Glass Model	269	13.2.2	Functional Groups in Biological Molecules	338
11.3.6	The Neutral Theory of Molecular Evolution	270	13.2.3	Major Classes of Biological Molecules	338
11.4	Evolutionary Game Theory	271	13.3	Structural Cell Biology	345
11.4.1	Social Interactions	272	13.3.1	Structure and Function of Biological Membranes	347
11.4.2	Game Theory	273	13.3.2	Nucleus	349
11.4.3	Evolutionary Game Theory	274	13.3.3	Cytosol	349
11.4.4	Replicator Equation for Population Dynamics	274	13.3.4	Mitochondria	350
11.4.5	Evolutionarily Stable Strategies	275	13.3.5	Endoplasmic Reticulum and Golgi Complex	350
11.4.6	Dynamical Behavior in the Rock–Scissors–Paper Game	276	13.3.6	Other Organelles	351
11.4.7	Evolution of Cooperative Behavior	276	13.4	Expression of Genes	351
11.4.8	Compromises between Metabolic Yield and Efficiency	278	13.4.1	Transcription	351
	Exercises	279	13.4.2	Processing of the mRNA	353
	References	280	13.4.3	Translation	353
	Further Reading	283	13.4.4	Protein Sorting and Posttranslational Modifications	355
12	Models of Biochemical Systems	285	13.4.5	Regulation of Gene Expression	355
12.1	Metabolic Systems	285		Exercises	356
12.1.1	Basic Elements of Metabolic Modeling	286		References	356
12.1.2	Toy Model of Upper Glycolysis	286		Further Reading	356
12.1.3	Threonine Synthesis Pathway Model	289	14	Experimental Techniques	357
12.2	Signaling Pathways	291	14.1	Restriction Enzymes and Gel Electrophoresis	358
12.2.1	Function and Structure of Intra- and Intercellular Communication	292	14.2	Cloning Vectors and DNA Libraries	359
12.2.2	Receptor–Ligand Interactions	293	14.3	1D and 2D Protein Gels	361
12.2.3	Structural Components of Signaling Pathways	295	14.4	Hybridization and Blotting Techniques	362
12.2.4	Analysis of Dynamic and Regulatory Features of Signaling Pathways	304	14.4.1	Southern Blotting	363
12.3	The Cell Cycle	307	14.4.2	Northern Blotting	363
12.3.1	Steps in the Cycle	309	14.4.3	Western Blotting	363
12.3.2	Minimal Cascade Model of a Mitotic Oscillator	310	14.4.4	<i>In Situ</i> Hybridization	364
12.3.3	Models of Budding Yeast Cell Cycle	311	14.5	Further Protein Separation Techniques	364
12.4	The Aging Process	314	14.5.1	Centrifugation	364
12.4.1	Evolution of the Aging Process	316	14.5.2	Column Chromatography	364
12.4.2	Using Stochastic Simulations to Study Mitochondrial Damage	318	14.6	Polymerase Chain Reaction	365
12.4.3	Using Delay Differential Equations to Study Mitochondrial Damage	323	14.7	Next-Generation Sequencing	366
	Exercises	327	14.8	DNA and Protein Chips	367
	References	327	14.8.1	DNA Chips	367
Part Two Reference Section	331		14.8.2	Protein Chips	367
13	Cell Biology	333	14.9	RNA-Seq	368
13.1	The Origin of Life	334	14.10	Yeast Two-Hybrid System	368
13.2	Molecular Biology of the Cell	336	14.11	Mass Spectrometry	369
			14.12	Transgenic Animals	370
			14.12.1	Microinjection and ES Cells	370
			14.12.2	Genome Editing Using ZFN, TALENs, and CRISPR	370
			14.13	RNA Interference	371
			14.14	ChIP-on-Chip and ChIP-PET	372
			14.15	Green Fluorescent Protein	374
			14.16	Single-Cell Experiments	375

14.17	Surface Plasmon Resonance	376			
	Exercises	377	15.7.5 Clustering Algorithms	430	
	References	377	15.7.6 Cluster Validation	435	
			15.7.7 Overrepresentation and Enrichment		
			Analyses	436	
			15.7.8 Classification Methods	438	
			Exercises	441	
			References	443	
15	Mathematical and Physical Concepts	381			
15.1	Linear Algebra	381	16	Databases	445
15.1.1	Linear Equations	381	16.1	General-Purpose Data Resources	445
15.1.2	Matrices	384	16.1.1	PathGuide	445
15.2	Dynamic Systems	386	16.1.2	BioNumbers	446
15.2.1	Describing Dynamics with Ordinary Differential		16.2	Nucleotide Sequence Databases	446
	Equations	386	16.2.1	Data Repositories of the National	
15.2.2	Linearization of Autonomous Systems	388		Center for Biotechnology	
15.2.3	Solution of Linear ODE Systems	388		Information	446
15.2.4	Stability of Steady States	388	16.2.2	GenBank/RefSeq/UniGene	446
15.2.5	Global Stability of Steady States	390	16.2.3	Entrez	447
15.2.6	Limit Cycles	390	16.2.4	EMBL Nucleotide Sequence Database	447
15.3	Statistics	391	16.2.5	European Nucleotide Archive	447
15.3.1	Basic Concepts of Probability Theory	391	16.2.6	Ensembl	447
15.3.2	Descriptive Statistics	396	16.3	Protein Databases	448
15.3.3	Testing Statistical Hypotheses	399	16.3.1	UniProt/Swiss-Prot/TrEMBL	448
15.3.4	Linear Models	401	16.3.2	Protein Data Bank	448
15.3.5	Principal Component Analysis	404	16.3.3	PANTHER	448
15.4	Stochastic Processes	405	16.3.4	InterPro	448
15.4.1	Chance in Physical Theories	405	16.3.5	iHOP	449
15.4.2	Mathematical Random Processes	406	16.4	Ontology Databases	449
15.4.3	Brownian Motion as a Random Process	407	16.4.1	Gene Ontology	449
15.4.4	Markov Processes	409	16.5	Pathway Databases	449
15.4.5	Markov Chains	410	16.5.1	KEGG	450
15.4.6	Jump Processes in Continuous Time	410	16.5.2	Reactome	450
15.4.7	Continuous Random Processes	411	16.5.3	ConsensusPathDB	451
15.4.8	Moment-Generating Functions	412	16.5.4	WikiPathways	451
15.5	Control of Linear Dynamical Systems	412	16.6	Enzyme Reaction Kinetics	
15.5.1	Linear Dynamical Systems	413		Databases	451
15.5.2	System Response and Linear Filters	414	16.6.1	BRENDA	451
15.5.3	Random Fluctuations and Spectral Density	415	16.6.2	SABIO-RK	452
15.5.4	The Gramian Matrices	415	16.7	Model Collections	452
15.5.5	Model Reduction	416	16.7.1	BioModels	452
15.5.6	Optimal Control	416	16.7.2	JWS Online	452
15.6	Biological Thermodynamics	417	16.8	Compound and Drug Databases	452
15.6.1	Microstate and Statistical Ensemble	417	16.8.1	ChEBI	453
15.6.2	Boltzmann Distribution and Free Energy	418	16.8.2	Guide to PHARMACOLOGY	453
15.6.3	Entropy	419	16.9	Transcription Factor Databases	453
15.6.4	Equilibrium Constant and Energies	421	16.9.1	JASPAR	453
15.6.5	Chemical Reaction Systems	422	16.9.2	TRED	453
15.6.6	Nonequilibrium Reactions	424	16.9.3	Transcription Factor Encyclopedia	454
15.6.7	The Role of Thermodynamics in Systems		16.10	Microarray and Sequencing	
	Biology	425		Databases	454
15.7	Multivariate Statistics	426	16.10.1	Gene Expression Omnibus	454
15.7.1	Planning and Designing Experiments for		16.10.2	ArrayExpress	454
	Case-Control Studies	426		References	455
15.7.2	Tests for Differential Expression	427			
15.7.3	Multiple Testing	428			
15.7.4	ROC Curve Analysis	429			

17	Software Tools for Modeling	457
17.1	13C-Flux2	458
17.2	Antimony	458
17.3	Berkeley Madonna	459
17.4	BIOCHAM	459
17.5	BioNetGen	459
17.6	Biopython	459
17.7	BioTapestry	460
17.8	BioUML	460
17.9	CellDesigner	460
17.10	CellNetAnalyzer	460
17.11	Copasi	461
17.12	CPN Tools	461
17.13	Cytoscape	461
17.14	E-Cell	461
17.15	EvA2	461
17.16	FEniCS Project	462
17.17	Genetic Network Analyzer (GNA)	462
17.18	Jarnac	462
17.19	JDesigner	463
17.20	JSim	463
17.21	KNIME	463
17.22	libSBML	464
17.23	MASON	464
17.24	Mathematica	464
17.25	MathSBML	465
17.26	Matlab	465
17.27	MesoRD	465
17.28	Octave	465
17.29	Omix Visualization	466
17.30	OpenCOR	466
17.31	Oscill8	466
17.32	PhysioDesigner	466
17.33	PottersWheel	467
17.34	PyBioS	467
17.35	PySCeS	467
17.36	R	468
17.37	SAAM II	468
17.38	SBMLEditor	468
17.39	SemanticSBML	468
17.40	SBML-PET-MPI	469
17.41	SBMLsimulator	469
17.42	SBMLSqueezer	469
17.43	SBML Toolbox	470
17.44	SBtoolbox2	470
17.45	SBML Validator	470
17.46	SensA	470
17.47	SmartCell	471
17.48	STELLA	471
17.49	STEPS	471
17.50	StochKit2	471
17.51	SystemModeler	472
17.52	Systems Biology Workbench	472
17.53	Taverna	472
17.54	VANTED	473
17.55	Virtual Cell (VCell)	473
17.56	xCellerator	473
17.57	XPPAUT	473
	Exercises	474
	References	474
	Index	475

Preface

Systems biology is the scientific discipline that studies the systemic properties and dynamic interactions in a biological object, be it a cell, an organism, a virus, or an infected host, in a qualitative and quantitative manner and by combining experimental studies with mathematical modeling. Scientists can describe the inner processes of stars a thousand light years away with great accuracy. But how a tiny cell under our microscope grows and divides remains puzzling in many ways. We see kids growing, people aging, plants blooming, and microbes degrading their remains. We use yeast for brewery and bakery, and doctors prescribe drugs to cure diseases. But do we understand how processes of life work?

Starting in the nineteenth century, such processes have no longer been explained by referring to special “life forces,” but by laws of physics and chemistry. By studying the structure and dynamics of living systems in finer and finer details, researchers from different disciplines have revealed how life processes arise from the structure and functional organization of cells, how tens of thousands of biochemical components interact in orchestrated ways, and how these systems are regulated by genetic information and continuously adapted through mutations and selection. With this conceptual shift, new questions became central in biology: How does an organism’s phenotype emerge from the genotype, as encoded in the organism’s DNA, and how is it shaped by environmental factors? Initially, such questions were approached by statistics, for example, by studying what mutations are associated with specific inheritable diseases. But the task, now, is to understand the mechanistic details.

We can easily understand the effects of gene disruptions when gene products have simple, specific functions. However, most gene mutations have only weak or quantitative effects on physiology, and many genetic diseases are multifactorial. Tracing the effects of multiple mutations, of mutations affecting gene regulation, or of

drugs requires a deep, quantitative, and dynamical understanding of cell physiology. In recent years, high-throughput experiments, time series experiments, and imaging techniques with high resolution have provided us with a detailed picture of the cellular machinery. We can observe how physical structures are built, maintained, and reproduced, how the metabolic state is changing, and how signaling and regulation systems allow cells to adapt to their environment. However, to understand how all these systems act together – and how cells can work as complex, robust systems – cataloging and understanding single-cell components is not enough. Instead, we need to capture the global dynamics between these components. This is where mathematical models come into play.

Mathematical modeling has a long, though relatively marginal, tradition in biology, and has influenced the field in many ways. Models can be used to test hypotheses and to yield quantitative predictions or reveal gaps or inconsistencies in previous arguments, thus helping us to improve our understanding of biochemical processes. Inspired by the ideas of cybernetics in the sixties and seventies, dynamical systems theory and control theory have been increasingly applied to biochemical pathways. Thanks to powerful experimental techniques in genomics and proteomics, a wealth of biological data has accumulated and computational models of cells are now within reach. Systems biology, the discipline devoted to developing such models, uses biochemical networks as a main concept. It studies biological systems by investigating the network components and their interactions with the help of experimental high-throughput techniques and dedicated small-scale investigations and by integrating these data into networks and dynamical simulation models.

Like many new fields of research, systems biology started out with great expectations. High-throughput data and computational models were hoped to provide

answers to basic yet difficult biological questions: Why do we age? What processes control cell proliferation, and how? How do neurodegenerative disorders or diseases such as cancer develop? How can we engineer microbes more efficiently to produce valuable chemicals, fuels, or specific drugs? Only few of these goals have been achieved until now, and most of these questions remain on our agenda. Nevertheless, systems biology has greatly contributed to our understanding of cells and is increasingly becoming a standard part of biological research. It has fostered the formulation of new concepts and methods, such as statistical network analysis, the analysis of the robustness and fragility of dynamical systems, and the analysis of molecular noise. Even more importantly, it has enabled experimental biologists to realize that some scientific ideas cannot be easily expressed by words only. Inspired by electrical engineering, biologists now communicate the structure of biochemical systems by network graphics, which can then be translated into dynamical models.

This book gives an overview of systems biology as a rapidly developing field and provides readers with established and emerging tools and methods. You will learn how to formulate mathematical models of biological processes, how to analyze them, how to use experimental data and other types of knowledge to make models more precise, and how to interpret their simulation results. Based on our own experiences in teaching undergraduate and graduate students, the book is designed as an introductory course for students of biology, biophysics, and bioinformatics. It is as well useful for senior scholars who approach systems biology for the first time or seek more information about specific concepts and techniques. In the first chapters, we introduce stoichiometric and kinetic models, the main theoretical frameworks for metabolism and signaling pathways. We continue with methods for model construction (including model fitting, data handling, and model reduction) and related formalisms (spatial, discrete, and stochastic models). Then, we move on to experimental high-throughput techniques and to cellular networks. The analysis of regulation networks leads us to more general perspectives on cell physiology, including modularity, robustness, and optimality. The main part of the book

ends with a chapter on case studies. Addressing readers with different scientific backgrounds, we have added a reference section summarizing some basic knowledge of cell biology and mathematics, followed by a survey of popular biological databases and software tools. Further material is available on an accompanying Web site (<http://www.wiley-vch.de/home/systemsbiology>), which also contains solutions to the exercises presented in the book.

For the second edition of this book, we have updated and expanded the text to reflect advances in the field, and have reorganized the chapters to improve readability. Many of the changes reflect current developments in systems biology. On the one hand, the development of software tools is a very active area, where many new tools are developed, while others drop into oblivion. In the meantime, SBML has become an established exchange format for computational models in systems biology. We also notice that systems biology as a whole has become a mainstream discipline: High-throughput measurements have become an integral part of cell biology, computational models are used in research and teaching, and collaborations between experimentalists and theoreticians are increasingly common. Today, systems biology is perceived as what it is: the endeavor to understand complex processes in living organisms. Not more, but also not less!

We thank our friends and colleagues who helped us write this book. We are especially grateful to Mariapaola Gritti, Bernd Binder, Andreas Hoppe, Dagmar Waltemath, Elad Noor, Avi Flamholz, Terence Hwa, Ron Milo, Jonathan Karr, Ulrich Liebermeister, David Jesinghaus, Martina Fröhlich, and Severin Ehret for reading and commenting on the text. We thank the Max Planck Society for support and encouragement. We are grateful to the European Commission for funding via different European projects (UniCellSys, SysteMTb and FinSysB to EK, HeCaToS 602156), the German Ministry for Education and Research, BMBF (ViroSign, OncoPath, SysToxChip to EK), and the German Research Foundation (GRK 1772 to EK, L1 1676/2-1 to WL).

The book is dedicated to our teacher Prof. Dr. Reinhart Heinrich (1946–2006) whose work on metabolic control theory in the 1970s paved the way to systems biology and who greatly inspired our minds.

Guide to Different Topics of the Book**Biological systems and processes**

Metabolism (2, 3, 4, 11.1, 11.4, 12.1)
 Gene regulatory network (7, 8.2, 9)
 Gene expression regulation (2, 9)
 Signaling systems (8.2, 12.2)
 Cell cycle (12.3)
 Development (7.3)
 Aging (12.4)

Concepts for biological function

Qualitative behavior (3, 7.1)
 Parameter sensitivity and robustness (4.2, 10.2)
 Modularity and functional subsystems (6.4, 8.3)
 Robustness against failure (10.2)
 Information (10.3, 15.6)
 Population heterogeneity (10.3)
 Optimality (3.2, 11.1, 11.2)
 Evolution (11.3)
 Population dynamics and game theory (11.4)

Model types with different levels of abstraction

Statistical particle models (15.6)
 Stochastic biochemical models (5.1.2, 7.2, 9.4, 15.4)
 Kinetic models (4, 5.1.1, 11, 12, 16.7)
 Constraint-based models (3.2)
 Discrete models (7.1)
 Spatial models (7.3)

Modeling skills

Model building (2, 3, 4, 5.1, 7)
 Model annotation (5.4)
 Parameter estimation (6.1)
 Model testing and selection (6.2)
 Local sensitivity analysis/control theory (4.2, 10.1, 10.2)
 Global sensitivity/uncertainty analysis (10.1)
 Model reduction (6.3)
 Model combination (5.4, 6.4)
 Network theory (8)
 Statistics (15.3, 15.7)
 Optimization of model outputs and structure (11.1)
 Optimal temporal control (11.2, 15.5)

Mathematical frameworks to describe cell states

Topological (network structures) (3.1, 8)
 Structural stoichiometric models (3)
 Dynamical systems (4, 12, 15.2)
 Deterministic linear models (6.3)
 Deterministic kinetic models (4, 9.1, 12)
 Uncertain parameters (10.1)
 Optimization and control theory (4.2, 11.1, 11.2, 15.5)

Practical issues in modeling

Use of databases (5.3, 16)
 Data formats (5.2, 5.4)
 Data sources (5.3, 16)
 Modeling software (5.1, 17)
 Simulation techniques and tools (5.1)
 Model visualization (5.1)
 Data visualization (8.3)

Experimental Techniques

Experimental techniques (14)

About the Authors

Edda Klipp (born 1965) studied biophysics at the Humboldt-Universität zu Berlin and obtained a PhD of theoretical biophysics at the HU Berlin and an honorary doctor at Gothenburg University. She is professor for theoretical biophysics at HU Berlin. A founding member of the International Society for Systems Biology, her research interests include mathematical modeling of cellular systems, application of physical principles in biology, systems biology, and development of modeling tools.

Wolfram Liebermeister (born 1972) studied physics in Tübingen and Hamburg and obtained a PhD of theoretical biophysics at the Humboldt University of Berlin. In his work on complex biological systems, he highlights functional aspects such as variability, information, and optimality.

Christoph Wierling (born 1973) studied biology in Münster and holds a PhD in biochemistry obtained from the Free University Berlin. He was leading a research group for systems biology at the Max Planck Institute for Molecular Genetics, Berlin. Currently he is heading the bioinformatics and modeling unit at Alacris Theranostics, a Berlin-based company applying NGS and systems biology approaches for translational research and personalized medicine. His research interests focus on modeling and simulation of biological systems and the development of systems biology software.

Axel Kowald (born 1963) holds a PhD in mathematical biology from the National Institute for Medical Research, London. His current research interests focus on the mathematical modeling of processes involved in the biology of aging and systems biology.

Part One

Introduction to Systems Biology

Introduction

1.1

Biology in Time and Space

Biological systems such as organisms, cells, or biomolecules are highly organized in their structure and function. They have developed during evolution and can only be fully understood in this context. To study them and to apply mathematical, computational, or theoretical concepts, we have to be aware of the following circumstances.

The continuous reproduction of cell compounds necessary for living and the respective flow of information is captured by the central dogma of molecular biology, which can be summarized as follows: genes code for mRNA, mRNA serves as template for proteins, and proteins perform cellular work. Although information is stored in the genes encoded by the DNA sequence, it is made available only through the cellular machinery that can decode this sequence and can translate it into structure and function. In this book, we will explain that from various perspectives.

A description of biological entities and their properties encompasses different levels of organization and different time scales. We can study biological phenomena at the level of populations, individuals, tissues, organs, cells, and compartments down to molecules and atoms. Length scales range from the order of meter (e.g., the size of whale or human) to micrometer for many cell types, down to picometer for atom sizes. Time scales include millions of years for evolutionary processes, annual and daily cycles, seconds for many biochemical reactions, and femtoseconds for molecular vibrations. Figure 1.1 gives an overview about scales.

In a unified view of cellular networks, each action of a cell involves different levels of cellular organization, including genes, proteins, metabolism, or signaling pathways. Therefore, the current description of the individual networks must be integrated into a larger framework.

1.1 Biology in Time and Space

1.2 Models and Modeling

- What Is a Model?
- Purpose and Adequateness of Models
- Advantages of Computational Modeling

1.3 Basic Notions for Computational Models

- Model Scope
- Model Statements
- System State
- Variables, Parameters, and Constants
- Model Behavior
- Model Classification
- Steady States
- Model Assignment Is Not Unique

1.4 Networks

1.5 Data Integration

1.6 Standards

1.7 Model Organisms

- *Escherichia coli*
- *Saccharomyces cerevisiae*
- *Caenorhabditis elegans*
- *Drosophila melanogaster*
- *Mus musculus*

References

Further Reading

Many current approaches pay tribute to the fact that biological items are subject to evolution. The structure and organization of organisms and their cellular machinery has developed during evolution to fulfill major functions such as growth, proliferation, and survival under changing conditions. If parts of the organism or of the cell fail to perform their function, the individual might become unable to survive or replicate.

One consequence of evolution is the similarity of biological organisms of different species. This similarity

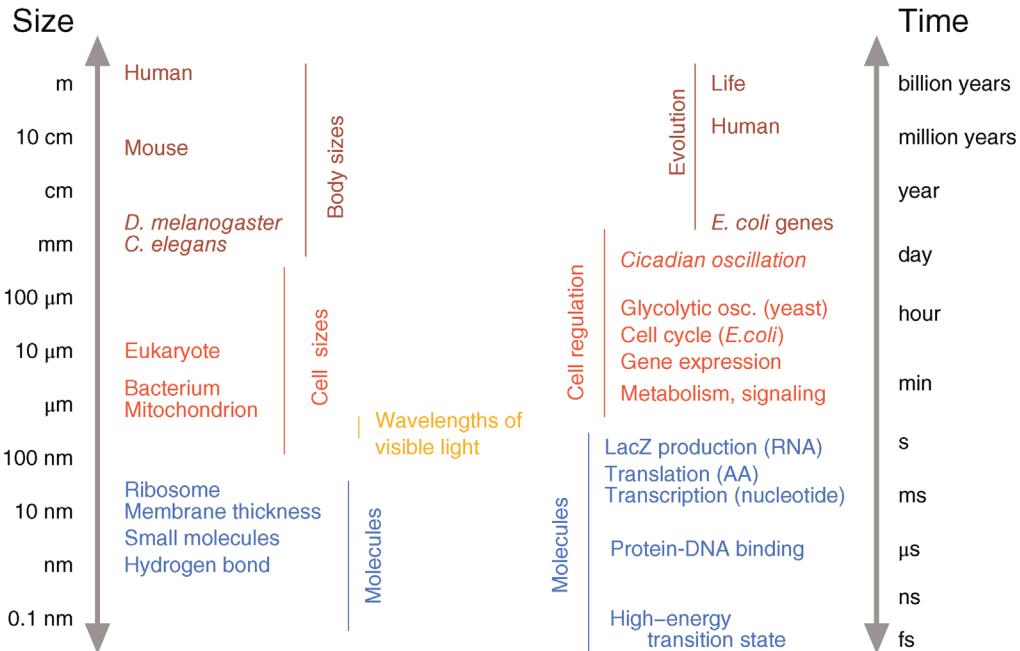


Figure 1.1 Length and time scales in biology. (Data from the BioNumbers database at biornumbers.hms.harvard.edu.)

allows for the use of model organisms and for the critical transfer of insights gained from one cell type to other cell types. Applications include, for example, prediction of protein function from similarity, prediction of network properties from optimality principles, reconstruction of phylogenetic trees, or the identification of regulatory DNA sequences through cross-species comparisons. However, the evolutionary process also leads to genetic variations within species. Therefore, personalized medicine and research is an important new challenge for biomedical research.

1.2 Models and Modeling

If we observe biological phenomena, we are confronted with various complex processes that often cannot be explained from first principles and the outcome of which cannot reliably be foreseen from intuition. Even if general biochemical principles are well established (e.g., the central dogma of transcription and translation or the biochemistry of enzyme-catalyzed reactions), the biochemistry of individual molecules and systems is often unknown and can vary considerably between species. Experiments lead to biological hypotheses about individual processes, but it often remains unclear whether these hypotheses can be combined into a larger coherent picture because it is often difficult to foresee the global

behavior of a complex system from knowledge of its parts. Mathematical modeling and computer simulations can help us to understand the internal nature and dynamics of these processes and to arrive at predictions about their future development and the effect of interactions with the environment.

1.2.1 What Is a Model?

The answer to this question will differ among communities of researchers. In a broad sense, a model is an abstract representation of objects or processes that explains features of these objects or processes (Figure 1.2). A biochemical reaction network can be represented by a graphical sketch showing dots for metabolites and arrows for reactions; the same network could also be described by a system of differential equations, which allows simulating and predicting the dynamic behavior of that network. If a model is used for simulations, it needs to be ensured that it faithfully predicts the system's behavior – at least those aspects that are supposed to be covered by the model. Systems biology models are often based on well-established physical laws that justify their general form, for instance, the thermodynamics of chemical reactions. Besides this, a computational model needs to make specific statements about a system of interest – which are partially justified by experiments and biochemical knowledge, and partially by mere extrapolation from other systems. Such a model can

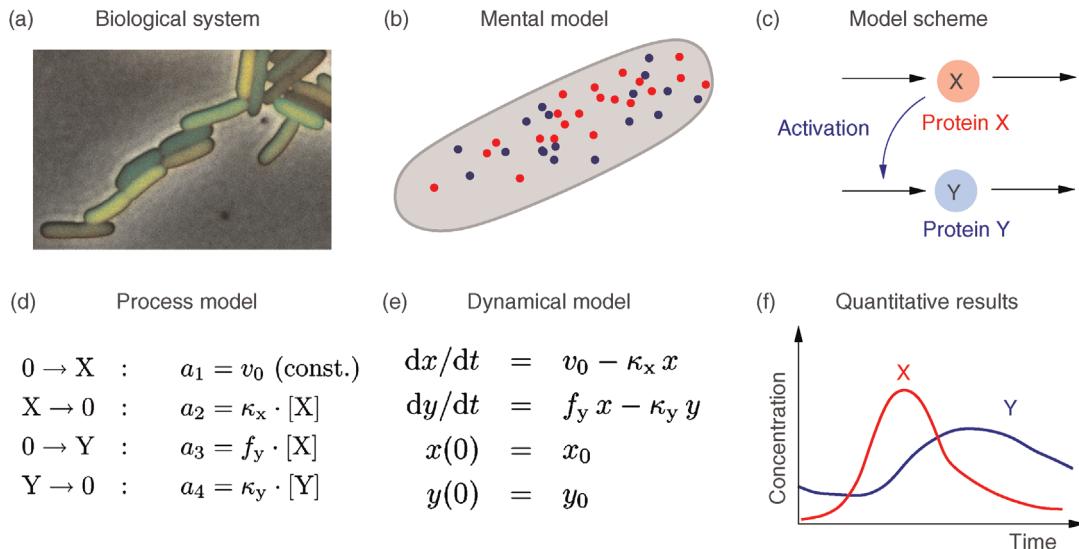


Figure 1.2 Typical abstraction steps in mathematical modeling. (a) *E. coli* bacteria produce thousands of different proteins. If a specific protein type is labeled with a fluorescent marker, cells glow under the microscope according to the concentration of this marker. (Courtesy of M. Elowitz.) (b) In a simplified mental model, we assume that cells contain two enzymes of interest, X (red) and Y (blue), and that the molecules (dots) can freely diffuse within the cell. All other substances are disregarded for the sake of simplicity. (c) The interactions between the two protein types can be drawn in a wiring scheme: each protein can be produced or degraded (black arrows). In addition, we assume that proteins of type X can increase the production of protein Y. (d) All individual processes to be considered are listed together with their rates a (occurrence per time). The mathematical expressions for the rates are based on a simplified picture of the actual chemical processes. (e) The list of processes can be translated into different sorts of dynamic models, in this case, deterministic rate equations for the protein concentrations x and y . (f) By solving the model equations, predictions for the time-dependent concentrations can be obtained. If the predictions do not agree with experimental data, this indicates that the model is wrong or too much simplified. In both cases, the model has to be refined.

summarize established knowledge about a system in a coherent mathematical formulation. In experimental biology, the term “model” is also used to denote a species that is especially suitable for experiments; for example, a genetically modified mouse may serve as a model for human genetic disorders.

1.2.2 Purpose and Adequateness of Models

Modeling is a subjective and selective procedure. A model represents only specific aspects of reality but, if done properly, this is sufficient since the intention of modeling is to answer particular questions. If the only aim is to predict system outputs from given input signals, a model should display the correct input–output relation, while its interior can be regarded as a black box. However, if instead a detailed biological mechanism has to be elucidated, then the system’s structure and the relations between its parts must be described realistically. Some models are meant to be generally applicable to many similar objects (e.g., Michaelis–Menten kinetics holds for many enzymes, the promoter–operator concept is applicable to many genes, and gene regulatory motifs are common), while others are specifically tailored to one

particular object (e.g., the 3D structure of a protein, the sequence of a gene, or a model of deteriorating mitochondria during aging). The mathematical part can be kept as simple as possible to allow for easy implementation and comprehensible results. Or it can be modeled very realistically and be much more complicated. None of the characteristics mentioned above makes a model wrong or right, but they determine whether a model is appropriate to the problem to be solved. The phrase “essentially, all models are wrong, but some are useful” coined by the statistician George Box is indeed an appropriate guideline for model building.

1.2.3 Advantages of Computational Modeling

Models gain their reference to reality from comparison with experiments, and their benefits therefore depend on the quality of the experiments used. Nevertheless, modeling combined with experimentation has a lot of advantages compared with purely experimental studies:

- Modeling drives conceptual clarification. It requires verbal hypotheses to be made specific and conceptually rigorous.

- Modeling highlights gaps in knowledge or understanding. During the process of model formulation, unspecified components or interactions have to be determined.
- Modeling provides independence of the modeled object.
- Time and space may be stretched or compressed *ad libitum*.
- Solution algorithms and computer programs can be used independently of the concrete system.
- Modeling is cheap compared with experiments.
- Models exert by themselves no harm on animals or plants and help to reduce ethical problems in experiments. They do not pollute the environment.
- Modeling can assist experimentation. With an adequate model, one may test different scenarios that are not accessible by experiment. One may follow time courses of compounds that cannot be measured in an experiment. One may impose perturbations that are not feasible in the real system. One may cause precise perturbations without directly changing other system components, which is usually impossible in real systems. Model simulations can be repeated often and for many different conditions.
- Model results can often be presented in precise mathematical terms that allow for generalization. Graphical representation and visualization make it easier to understand the system.
- Finally, modeling allows for making well-founded and testable predictions.

The attempt to formulate current knowledge and open problems in mathematical terms often uncovers a lack of knowledge and requirements for clarification. Furthermore, computational models can be used to test whether proposed explanations of biological phenomena are feasible. Computational models serve as repositories of current knowledge, both established and hypothetical, about how systems might operate. At the same time, they provide researchers with quantitative descriptions of this knowledge and allow them to simulate the biological process, which serves as a rigorous consistency test.

1.3 Basic Notions for Computational Models

1.3.1 Model Scope

Systems biology models consist of mathematical elements that describe properties of a biological system, for instance, mathematical variables describing the concentrations of

metabolites. As a model can only describe certain aspects of the system, all other properties of the system (e.g., concentrations of other substances or the environment of a cell) are neglected or simplified. It is important – and, to some extent, an art – to construct models in such ways that the disregarded properties do not compromise the basic results of the model.

1.3.2 Model Statements

Alongside the model elements, a model can contain various kinds of statements and equations describing facts about the model elements, most notably, their temporal behavior. In kinetic models, the basic modeling paradigm considered in this book, the dynamics is determined by a set of ordinary differential equations describing the substance balances. Statements in other model types may have the form of equality or inequality constraints (e.g., in flux balance analysis), maximality postulates, stochastic processes, or probabilistic statements about quantities that vary in time or between cells.

1.3.3 System State

In dynamical systems theory, a system is characterized by its *state*, a snapshot of the system at a given time. The state of the system is described by the set of variables that must be kept track of in a model: in deterministic models, it needs to contain enough information to predict the behavior of the system for all future times. Each modeling framework defines what is meant by the state of the system. In kinetic rate equation models, for example, the state is a list of substance concentrations. In the corresponding stochastic model, it is a probability distribution or a list of the current number of molecules of a species. In a Boolean model of gene regulation, the state is a string of bits indicating for each gene whether it is expressed ("1") or not expressed ("0"). Also, the temporal behavior can be described in fundamentally different ways. In a *dynamical system*, the future states are determined by the current state, while in a *stochastic process*, the future states are not precisely predetermined. Instead, each possible future history has a certain probability to occur.

1.3.4 Variables, Parameters, and Constants

The quantities in a model can be classified as variables, parameters, and constants. A *constant* is a quantity with a fixed value, such as the natural number e or Avogadro's number (number of molecules per mole). *Parameters* are

quantities that have a given value, such as the K_m value of an enzyme in a reaction. This value depends on the method used and on the experimental conditions and may change. *Variables* are quantities with a changeable value for which the model establishes relations. A subset of variables, the *state variables*, describes the system behavior completely. They can assume independent values and each of them is necessary to define the system state. Their number is equivalent to the dimension of the system. For example, the diameter d and volume V of a sphere obey the relation $V = \pi d^3/6$, where π and 6 are constants, V and d are variables, but only one of them is a state variable since the relation between them uniquely determines the other one.

Whether a quantity is a variable or a parameter depends on the model. In reaction kinetics, the enzyme concentration appears as a parameter. However, the enzyme concentration itself may change due to gene expression or protein degradation, and in an extended model, it may be described by a variable.

1.3.5 Model Behavior

Two fundamental factors that determine the behavior of a system are (i) influences from the environment (input) and (ii) processes within the system. The system structure, that is, the relation among variables, parameters, and constants, determines how endogenous and exogenous forces are processed. However, different system structures may still produce similar system behavior (output); therefore, measurements of the system output often do not suffice to choose between alternative models and to determine the system's internal organization.

1.3.6 Model Classification

For modeling, processes are classified with respect to a set of criteria.

- A structural or *qualitative* model (e.g., a network graph) specifies the interactions among model elements. A *quantitative* model assigns values to the elements and to their interactions, which may or may not change.
- In a *deterministic* model, the system evolution through all following states can be predicted from the knowledge of the current state. *Stochastic* descriptions give instead a probability distribution for the successive states.
- The nature of values that time, state, or space may assume distinguishes a *discrete* model (where values are taken from a discrete set) from a *continuous* model (where values belong to a continuum).

- *Reversibility* processes can proceed in a forward and backward direction. Irreversibility means that only one direction is possible.
- *Periodicity* indicates that the system assumes a series of states in the time interval $\{t, t + \Delta t\}$ and again in the time interval $\{t + i\Delta t, t + (i + 1)\Delta t\}$ for $i = 1, 2, \dots$.

1.3.7

Steady States

The concept of stationary states is important for the modeling of dynamical systems. *Stationary states* (other terms are *steady states* or *fixed points*) are determined by the fact that the values of all state variables remain constant in time. The asymptotic behavior of dynamic systems, that is, the behavior after a sufficiently long time, is often stationary. Other types of asymptotic behavior are oscillatory or chaotic regimes.

The consideration of steady states is actually an abstraction that is based on a separation of time scales. In nature, everything flows. Fast and slow processes – ranging from formation and breakage of chemical bonds within nanoseconds to growth of individuals within years – are coupled in the biological world. While fast processes often reach a quasi-steady state after a short transition period, the change of the value of slow variables is often negligible in the time window of consideration. Thus, each steady state can be regarded as a quasi-steady state of a system that is embedded in a larger nonstationary environment. Despite this idealization, the concept of stationary states is important in kinetic modeling because it points to typical behavioral modes of the system under study and it often simplifies the mathematical problems.

Other theoretical concepts in systems biology are only rough representations of their biological counterparts. For example, the representation of gene regulatory networks by Boolean networks, the description of complex enzyme kinetics by simple mass action laws, or the representation of multifarious reaction schemes by black boxes proved to be helpful simplifications. Although being a simplification, these models elucidate possible network properties and help to check the reliability of basic assumptions and to discover possible design principles in nature. Simplified models can be used to test mathematically formulated hypotheses about system dynamics, and such models are easier to understand and to apply to different questions.

1.3.8

Model Assignment Is Not Unique

Biological phenomena can be described in mathematical terms. Models developed during the last few decades range from the description of glycolytic oscillations with

ordinary differential equations to population dynamics models with difference equations, stochastic equations for signaling pathways, and Boolean networks for gene expression. However, it is important to realize that a certain process can be described in more than one way: a biological object can be investigated with different experimental methods and each biological process can be described with different (mathematical) models. Sometimes, a modeling framework represents a simplified limiting case (e.g., kinetic models as limiting case of stochastic models). On the other hand, the same mathematical formalism may be applied to various biological instances: statistical network analysis, for example, can be applied to cellular transcription networks, the circuitry of nerve cells, or food webs.

The choice of a mathematical model or an algorithm to describe a biological object depends on the problem, the purpose, and the intention of the investigator. Modeling has to reflect essential properties of the system and different models may highlight different aspects of the same system. This ambiguity has the advantage that different ways of studying a problem also provide different insights into the system. However, the diversity of modeling approaches makes it also very difficult to merge established models (e.g., for individual metabolic pathways) into larger supermodels (e.g., models of complete cell metabolism).

1.4 Networks

The network is a crucial concept in systems biology. We study protein–protein interaction networks, protein–RNA interaction networks, metabolic networks (see Chapters 3 and 4 and Section 12.1), signaling networks (Section 12.2), guilt-by-association networks, and networks connecting gene defects with diseases or diseases with other diseases via common gene defects [1]. Throughout this book, you will find more examples.

Networks are best represented by graphs that consist of nodes and edges, which connect the nodes, as illustrated in Figure 1.3. In protein–protein interaction networks, for example, nodes are proteins and edges are their interactions as can for instance be determined by yeast two-hybrid experiments (see Chapter 14). If appropriate, one can introduce different types of nodes for different types of components. For example, the metabolites and converting enzymes in metabolic networks can be represented with bipartite networks, which possess two types of nodes – one for metabolites and the other for enzymes – that are never directly connected by an edge, but only via the other type of node. Petri net type of modeling

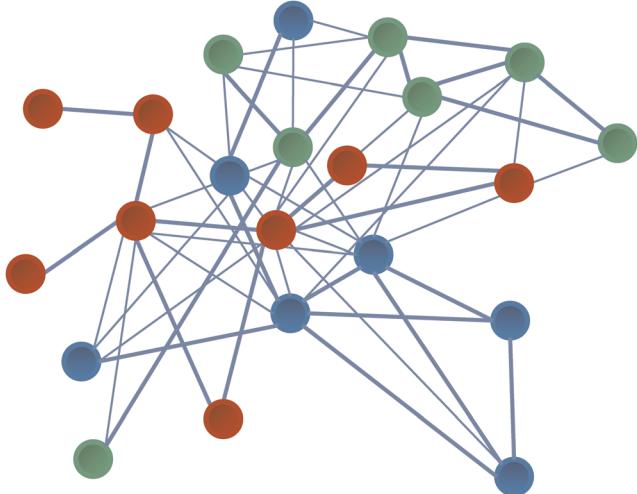


Figure 1.3 Network with nodes (circles) and edges (lines between circles). Different node colors indicate different types of connected components (e.g., proteins, mRNAs, and metabolites).

takes that property into account representing metabolites as places and enzyme-catalyzed reactions as transitions (see Section 7.1). By contrast, classical metabolic modeling considers only one type of node, but different types in different approaches. Systems of ordinary differential equations describing metabolite dynamics take metabolites as nodes and enzymatic reactions as edges (Chapter 4), while flux balance analysis restricts itself to steady states and now focusses on the fluxes through the reactions (now as nodes) that are linked by the stationary metabolites as edges.

1.5 Data Integration

Systems biology has evolved rapidly in the last few years, driven by the new high-throughput technologies. The most important impulse was given by large sequencing projects such as the Human Genome Project, which resulted in the full sequence of the human and other genomes [2,3]. Proteomic technologies have been used to identify the translation status of complete cells (2D gels, mass spectrometry) and to elucidate protein–protein interaction networks involving thousands of components [4]. However, to validate such diverse high-throughput data, one needs to correlate and integrate such information. Thus, an important part of systems biology is data integration.

On the lowest level of complexity, data integration implies common schemes for data storage, data representation, and data transfer. For particular experimental

techniques, this has already been established, for example, in the field of transcriptomics with Minimum Information About a Microarray Experiment [5], Minimum Information for Reporting Next Generation Sequence Genotyping [6], in proteomics with proteomics experiment data repositories [7], and the Human Proteome Organization consortium [8]. On a more complex level, schemes have been defined for biological models and pathways such as Systems Biology Markup Language (SBML) [9], CellML [10], or Systems Biology Graphical Notation (SBGN) [11], which all use an XML-like language style.

Data integration on the next level of complexity consists of data correlation. This is a growing research field as researchers combine information from multiple diverse data sets to learn about and explain natural processes [12,13]. For example, methods have been developed to integrate the results of transcriptome or proteome experiments with genome sequence annotations. In the case of complex disease conditions, it is clear that only integrated approaches can link clinical, genetic, behavioral, and environmental data with diverse types of molecular phenotype information and identify correlative associations. Such correlations, if found, are the key to identifying biomarkers and processes that are either causative or indicative of the disease. Importantly, the identification of biomarkers (e.g., proteins and metabolites) associated with the disease will open up the possibility to generate and test hypotheses on the biological processes and genes involved in this condition. The evaluation of disease-relevant data is a multistep procedure involving a complex pipeline of analysis and data handling tools such as data normalization, quality control, multivariate statistics, correlation analysis, visualization techniques, and intelligent database systems [14]. Several pioneering approaches have indicated the power of integrating data sets from different levels, for example, the correlation of gene membership of expression clusters and promoter sequence motifs [15], the combination of transcriptome and quantitative proteomics data in order to construct models of cellular pathways [13], and the identification of novel metabolite–transcript correlations [16]. Finally, data can be used to build and refine dynamical models, which represent an even higher level of data integration.

1.6 Standards

As experimental techniques generate rapidly growing amounts of data and large models need to be developed and exchanged, standards for both experimental procedures and modeling are a central practical issue in systems biology. Information exchange necessitates a

common language about biological aspects. One seminal example is the Gene Ontology that provides a controlled vocabulary that can be applied to all organisms, even as the knowledge about genes and proteins continues to accumulate. SBML [9] has been established as exchange language for mathematical models of biochemical reaction networks. SBGN [11] defines graphical elements to unambiguously represent biochemical reaction sets and large regulatory networks. A series of “minimum-information-about” statements based on community agreement defines standards for certain types of experiments. Minimum Information Requested in the Annotation of Biochemical Models (MIRIAM) [17] describes standards for this specific type of systems biology models. Minimum Information About a Simulation Experiment (MIASE) [18] helps authors to describe all elements of a computational experiment such that readers can repeat the simulations and create figures as shown in the publication.

1.7 Model Organisms

Model organisms are species that have developed over the years to be extremely popular for scientific investigations. The reasons for such popularity can be manifold. Of great importance is, of course, an easy handling of the organism, that is, culture conditions (temperature, pressure, etc.) that can be set up in the laboratory without much effort and that tolerate some degree of variation, so that results are comparable between groups that use slightly different growth conditions. However, other factors are also important, such as costs (for housing, food, etc.), size (the smaller the size, the more individuals can be studied), or lifespan (short-lived species are more popular for aging studies). Often model organisms are also used to represent important taxonomical properties (prokaryotes, eukaryotes, unicellular organisms, multicellular organisms, vertebrates, and invertebrates), but in all cases the hope is that important biochemical findings made in such model organisms are also of relevance for other species of that taxonomical group or even for humans. Figure 1.4 shows a selection of popular model species, which will be discussed in the next sections. They range from prokaryotic organisms to single and multicellular eukaryotic species up to mammals.

1.7.1 *Escherichia coli*

E. coli is probably the oldest and best studied model organism of all (Figure 1.4a). It is a rod-shaped



Figure 1.4 Popular model organisms for studies of problems in biochemistry and molecular biology. (a) *E. coli* is a rod-like bacterium and the best studied prokaryotic model system. (Public domain image from Wikimedia, http://commons.wikimedia.org/wiki/File:EscherichiaColi_NIAID.jpg.) (b) The yeast *S. cerevisiae* is a simple unicellular eukaryote and is of considerable scientific and industrial interest. (Public domain image from Wikimedia, http://commons.wikimedia.org/wiki/File:S_cerevisiae_under_DIC_microscopy.jpg.) (c) The nematode *C. elegans* is approximately 1 mm and is a popular representative for simple and short-lived multicellular organisms. ("Adult *Caenorhabditis elegans*" by Kbradnam (<http://en.wikipedia.org/wiki/User:Kbradnam>) is licensed under CC BY-SA-2.5, <http://creativecommons.org/licenses/by-sa/2.5/>.) (d) The fruit fly *D. melanogaster* is like *C. elegans* a model for simple multicellular organisms and has extensively been studied in developmental biology. ("*Drosophila melanogaster*" by A. Karwath (<http://commons.wikimedia.org/wiki/User:Aka>) is licensed under CC BY-SA-2.5, <http://creativecommons.org/licenses/by-sa/2.5/>.) (e) Finally, the mouse *M. musculus* is a popular model species for mammals and is thus also of great relevance for humans. (Public domain image from Wikimedia, http://commons.wikimedia.org/wiki/File:House_mouse.jpg.)

bacterium that is found in the intestines of many organisms, including humans. It is a facultative anaerobic organism, which means that it can grow under aerobic as well as anaerobic conditions. *E. coli* is roughly 2 µm long with a diameter of 0.5 µm. Under laboratory conditions, it can easily be cultivated and doubles its number in less than 30 min. It has been studied for more than 50 years and is the most popular prokaryotic model organism. The genome of the *E. coli* strain K-12 has completely been sequenced in 1997 [19] and contains around 4200 genes dispersed along 4.6 million base pairs (Mbp). It is a very streamlined genome containing very few intergenic sequences. The *E. coli* family consists of a large number of strains, and a comparison of the sequence of more than 60 strains has shown that they contain in total more than 15 500 genes, while

only 6% of this pan-genome is present in each strain [20]. *E. coli* was of pivotal importance for developing many of the experimental techniques described in Chapter 14. Today, a large number of scientific resources regarding this model species are available on the Internet. A good starting point is EcoCyc (ecocyc.org), which provides information about the genome and biochemical machinery of the *E. coli* strain K-12 MG1655. Other websites provide information about protein–protein interactions (bacteriome.org/) and systematic single-gene knockout mutants (<http://ecoli.aist-nara.ac.jp/gb6/Resources/deletion/deletion.html>), or a database of available strains (cgsc.biology.yale.edu). For modelers, the CyberCell Database (ccdb.wishartlab.com/CCDB) is also of interest since it aims at providing enzymatic, genetic, and biological information suitable

for developing mathematical models of all parts of a cell of *E. coli* strain K-12.

1.7.2

Saccharomyces cerevisiae

The yeast *S. cerevisiae* is a unicellular fungus, belonging to the ascomycetes (Figure 1.4b). It is not only a useful organism needed for the production of wine, beer, and bread, but also the best studied eukaryotic model system. The cells are easy to grow and double under optimal conditions every 90–100 min. Like *E. coli*, also *S. cerevisiae* can live under aerobic as well as anaerobic conditions. If oxygen is present, the majority of energy is generated via oxidative phosphorylation at the inner mitochondrial membrane and without oxygen energy is produced via glycolysis and fermentation. The yeast normally propagates as a diploid organism via mitosis. Under stress, however, the diploid cells can undergo sporulation, producing four haploid cells in the process. These haploid cells belong to one of two mating classes (sexes), called “a” and “o”. Haploids can either propagate via normal mitosis or mate with other haploids of the different mating class, resulting again in diploid cells. This life cycle makes *S. cerevisiae* interesting for genetic studies; it has also been extensively used by experimental and modeling studies of the cell cycle, glycolysis, osmotic shock, and mating process [21–28]. Cell division occurs in *S. cerevisiae* in an asymmetric fashion called budding and single-cell studies have shown that yeast cells exhibit replicative senescence with a maximum of 30–40 divisions [29]. Since this process is very reminiscent of the replicative senescence known from human fibroblasts [30], *S. cerevisiae* is also employed as a model system for investigations of the aging process. Furthermore, *S. cerevisiae* was also the first eukaryotic organism to be sequenced and its genome consists of about 12 Mbp containing roughly 6000 genes distributed over 16 chromosomes [31]. Homologous recombination (the exchange of sequences between similar strands of DNA) is very efficient in *S. cerevisiae*, which makes the organism also a convenient model for studies of synthetic biology. Using this mechanism, it was possible to replace the complete chromosome 16 with a new, synthetic one through 11 successive rounds of transformation (see Chapter 14) [32]. The synthetic chromosome was streamlined by removing all introns and superfluous tRNA genes and using only two of the three possible stop codons. This opens the possibility to extend the genetic code by a further amino acid once all chromosomes are modified in this way. A good online resource for further information about this model organism is the *Saccharomyces* Genome Database ([www.yeastgenome.org](http://yeastgenome.org)).

1.7.3

Caenorhabditis elegans

Of course, model systems for multicellular organisms are also needed and the nematode *C. elegans* (Figure 1.4c) has become such a model since Sidney Brenner introduced it to the research community [33]. Like the other model organisms, it is easy to cultivate (feeding on bacteria or synthetic medium) and thousands of the about 1 mm long animals can live on a large Petri dish. Wild populations of *C. elegans* consist mainly of hermaphrodites together with a few males. Hermaphrodites not only are capable of self-fertilization (leading to natural inbred lines), but can also mate with males. The hermaphrodite then lays eggs that develop into larvae after hatching and after a total of four larval stages (L1–L4) the adult animal emerges. The complete life cycle from egg to egg takes between 2.5 and 5.5 days, depending on the temperature. The total lifespan of *C. elegans* is rather short with 2–3 weeks. This made *C. elegans* another popular model system for the investigation of the aging process [34]. However, the nematode is also an important model for other fields of research such as molecular biology or neurology. RNA interference (RNAi), for instance, is an important experimental technique (Chapter 14) that was developed based on experiments in *C. elegans* [35]. Furthermore, adult nematodes have a fixed number of somatic cells that is identical for all individuals (1031 in the male and 959 in the hermaphrodite), which makes it possible to generate very detailed anatomical models of the worm. The “slidable worm” (www.wormatlas.org/slidableworm.htm), which is a resource available on the webpage of the WormAtlas database, presents the results of such anatomical studies using an easy-to-use interface. *C. elegans* is also the only animal for which the complete wiring diagram (connectome) of the nervous system has been determined (using electron microscopy serial sections) [36,37]. Finally, *C. elegans* has also been the first multicellular organism for which the complete genome sequence has been determined [38,39]. The 97 Mbp contain approximately 19 000 genes dispersed over six chromosomes. Good online starting points for more information are WormBase (www.wormbase.org), WormBook (www.wormbook.org/), or WormAtlas (www.wormatlas.org/).

1.7.4

Drosophila melanogaster

The fruit fly *D. melanogaster* (Figure 1.4d) is another, immensely popular, model organism that shares many of the properties of *C. elegans*. The animals are easy to breed in captivity and because of their small size (around 1 mm) it is possible to perform studies involving thousands of

individuals (e.g., for selection or population studies). The generation time (about 7 days at 29 °C) and lifespan (about 30 days at 29 °C) are very short and depend strongly on the ambient temperature. This facilitates, for example, artificial selection studies, which take several generations [40]. *D. melanogaster* has four chromosomes ($2n=8$), which can even be studied under the light microscope because of a phenomenon called polyteny. As in many insect larvae, the cells of the salivary glands of *D. melanogaster* undergo multiple rounds of replication without cell division, leading to hundreds of sister chromatids aligned to each other. Polyteny chromosomes are found in cells that need to express a large amount of a specific gene product and transcriptionally active areas appear under the microscope as swollen regions, so-called puffs. Although this technique is now outdated regarding the analysis of transcriptional activity, polyteny chromosomes are still valuable for taxonomic problems. After staining, the puffs form a specific banding pattern that can be used to identify chromosomal deletions and duplications. This can be used in taxonomy to differentiate and classify closely related subspecies. *D. melanogaster* was arguably the most important model species for investigating developmental processes in multicellular organisms [41], which has led to the discovery of Hox genes [42]. These genes code for a set of transcription factors that contain a common 180 bp motif (the homeo-domain) and control the development of the anterior-posterior axis of the animal. A unique feature of these genes is that they are arranged on the chromosomes in the same linear order as the body region that they affect (called collinearity). Thus, Hox genes at one end of the cluster control the development of the anterior region (head), while the genes at the other end of the cluster influence the development of the posterior region (tail). Although originally found in *Drosophila*, Hox genes have been found in many metazoans, including vertebrates [43]. The complete genome was sequenced in 2000 [44] and somewhat surprisingly the number of genes is with approximately 14 000 clearly smaller than the number of genes in *C. elegans*. Further information, tools, and resources are available at FlyBase (flybase.org) and Ensembl Genome Browser (www.ensembl.org/Drosophila_melanogaster).

1.7.5

Mus musculus

The last model system that we want to introduce here is the house mouse *M. musculus domesticus* (Figure 1.4e). It is clearly the model organism with the largest similarity to humans and is therefore also of great relevance for

human research. Humans and mice are both mammals and thus share a common ancestor roughly 80 million years ago, a rather short time span compared with the other model organisms. Consequently, the genome structure and organization is also very similar. The mouse genome, sequenced in 2002 [45], contains 2.5 Gbp and is thus somewhat smaller than the human genome with 2.9 Gbp [2,3], although both genomes contain approximately 20 000–25 000 genes. The similarity at the gene level is quite amazing insofar that for more than 99% of mouse genes a homolog can also be found in the human genome [3], and vice versa. The mouse is also a popular model system because it is very amenable to genetic manipulations. The first mice were cloned in 1998 [46] and today it is common routine to create transgenic mice by introducing DNA constructs into fertilized egg cells and to study the function of existing genes by knocking them out or down (see Chapter 14). The Knockout Mouse Project (KOMP), for instance, aims at generating and providing mouse embryonic stem cells (and eventually whole mice) with single-gene knockout for every gene in the mouse genome (www.komp.org). Because mice have been used for such a long time as model species, many different inbred strains have been developed, which differ in various aspects of their phenotype (e.g., size, lifespan, and disease susceptibility). Of special interest are the various strains of nude mice that have a deletion of the FOXN1 gene, which prevents the formation of a functioning thymus. Without a thymus, these mice cannot produce mature T lymphocytes and therefore lack most forms of immune response (the lack of fur is a side effect of this mutation). As a consequence, they are valuable tools to study tumor development and are also used for transplantation studies, since they do not reject allo- or xenografts. Useful starting points for further information are, for instance, the Mouse Genome Informatics (www.informatics.jax.org/), the Mouse Atlas Project (www.emouseatlas.org), or the Ensembl Genome Browser (www.ensembl.org/Mus_musculus).

References

- 1 Goh, K.I., Cusick, M.E., Valle, D., Childs, B., Vidal, M., and Barabási, A.L. (2007) The human disease network. *Proc. Natl. Acad. Sci. USA*, 104 (21), 8685–8690.
- 2 Lander, E.S. *et al.* (2001) Initial sequencing and analysis of the human genome. *Nature*, 409, 860–921.
- 3 Venter, J.C. *et al.* (2001) The sequence of the human genome. *Science*, 291, 1304–1351.
- 4 Mering, C. *et al.* (2002) Comparative assessment of large-scale data sets of protein–protein interactions. *Nature*, 417, 399–403.
- 5 Brazma, A. *et al.* (2001) Minimum Information About a Microarray Experiment (MIAME): toward standards for microarray data. *Nat. Genet.*, 29, 365–371.

- 6** Mack, S.J., Milius, R.P., Gifford, B.D., Sauter, J., Hofmann, J., Osoegawa, K., Robinson, J., Groeneweg, M., Turenchalk, G.S., Adai, A., Holcomb, C., Rozemuller, E.H., Penning, M.T., Heuer, M.L., Wang, C., Salit, M.L., Schmidt, A.H., Parham, P.R., Müller, C., Hague, T., Fischer, G., Fernandez-Viña, M., Hollenbach, J.A., Norman, P.J., and Maiers, M. (2015) Minimum Information for Reporting Next Generation Sequence Genotyping (MIRNG): guidelines for reporting HLA and KIR genotyping via next generation sequencing. *Hum. Immunol.* doi: 10.1016/j.humimm.2015.09.011.
- 7** Taylor, C.F. et al. (2003) A systematic approach to modeling, capturing, and disseminating proteomics experimental data. *Nat. Biotechnol.*, 21, 247–254.
- 8** Hermjakob, H. et al. (2004) The HUPO PSI's molecular interaction format: a community standard for the representation of protein interaction data. *Nat. Biotechnol.*, 22, 177–183.
- 9** Hucka, M. et al. (2003) The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19, 524–531.
- 10** Lloyd, C.M. et al. (2004) CellML: its future, present and past. *Prog. Biophys. Mol. Biol.*, 85, 433–450.
- 11** Le Novère, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A. et al. (2009) The Systems Biology Graphical Notation. *Nat. Biotechnol.*, 27 (8), 735–741.
- 12** Gitton, Y. et al. (2002) A gene expression map of human chromosome 21 orthologues in the mouse. *Nature*, 420, 586–590.
- 13** Ideker, T. et al. (2001) Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. *Science*, 292, 929–934.
- 14** Kanehisa, M. and Bork, P. (2003) Bioinformatics in the post-sequence era. *Nat. Genet.*, 33, 305–310.
- 15** Tavazoie, S. et al. (1999) Systematic determination of genetic network architecture. *Nat. Genet.*, 22, 281–285.
- 16** Urbanczyk-Wochniak, E. et al. (2003) Parallel analysis of transcript and metabolic profiles: a new approach in systems biology. *EMBO Rep.*, 4, 989–993.
- 17** Le Novère, N. et al. (2005) Minimum Information Requested in the Annotation of Biochemical Models (MIRIAM). *Nat. Biotechnol.*, 23, 1509–1515.
- 18** Waltemath, D., Adams, R., Beard, D.A., Bergmann, F.T., Bhalla, U.S., Britten, R. et al. (2011) Minimum Information About a Simulation Experiment (MIASE). *PLoS Comput. Biol.*, 7 (4), e1001122.
- 19** Blattner, F.R., Plunkett, G., 3rd, Bloch, C.A., Perna, N.T., Burland, V., Riley, M. et al. (1997) The complete genome sequence of *Escherichia coli* K-12. *Science*, 277 (5331), 1453–1462.
- 20** Lukjancenko, O., Wassenaar, T.M., and Ussery, D.W. (2010) Comparison of 61 sequenced *Escherichia coli* genomes. *Microb. Ecol.*, 60 (4), 708–720.
- 21** Hynne, F., Dano, S., and Sorensen, P.G. (2001) Full-scale model of glycolysis in *Saccharomyces cerevisiae*. *Biophys. Chem.*, 94 (1–2), 121–163.
- 22** Klipp, E., Nordlander, B., Kruger, R., Gennemark, P., and Hohmann, S. (2005) Integrative model of the response of yeast to osmotic shock. *Nat. Biotechnol.*, 23 (8), 975–982.
- 23** Diener, C., Schreiber, G., Giese, W., del Rio, G., Schroder, A., and Klipp, E. (2014) Yeast mating and image-based quantification of spatial pattern formation. *PLoS Comput. Biol.*, 10 (6), e1003690.
- 24** Kofahl, B. and Klipp, E. (2004) Modelling the dynamics of the yeast pheromone pathway. *Yeast*, 21 (10), 831–850.
- 25** Adrover, M.À., Zi, Z., Duch, A., Schaber, J., González-Novo, A., Jimenez, J., Nadal-Ribelles, M., Clotet, J., Klipp, E., and Posas, F. (2011) Time-dependent quantitative multicomponent control of the G₁-S network by the stress-activated protein kinase Hog1 upon osmostress. *Sci. Signal.*, 4 (192), ra63. Erratum: *Sci. Signal.*, 4 (197), er5 (2011).
- 26** Chen, K.C., Calzone, L., Csikasz-Nagy, A., Cross, F.R., Novak, B., and Tyson, J.J. (2004) Integrative analysis of cell cycle control in budding yeast. *Mol. Biol. Cell*, 15 (8), 3841–3862.
- 27** Rizzi, M., Baltes, M., Theobald, U., and Reuss, M. (1997) *In vivo* analysis of metabolic dynamics in *Saccharomyces cerevisiae*. II. Mathematical model. *Biotechnol. Bioeng.*, 55 (4), 592–608.
- 28** Kaizu, K., Ghosh, S., Matsuoka, Y., Moriya, H., Shimizu-Yoshida, Y., and Kitano, H. (2010) A comprehensive molecular interaction map of the budding yeast cell cycle. *Mol. Syst. Biol.*, 6, 415.
- 29** Jazwinski, S.M. (1990) Aging and senescence of the budding yeast *Saccharomyces cerevisiae*. *Mol. Microbiol.*, 4 (3), 337–343.
- 30** Hayflick, L. (1965) The limited *in vitro* lifetime of human diploid cell strains. *Exp. Cell Res.*, 37, 614–636.
- 31** Mewes, H.W., Albermann, K., Bahr, M., Frishman, D., Gleissner, A., Hani, J. et al. (1997) Overview of the yeast genome. *Nature*, 387 (6632 Suppl.), 7–65.
- 32** Annaluru, N., Muller, H., Mitchell, L.A., Ramalingam, S., Stracquadanio, G., Richardson, S.M. et al. (2014) Total synthesis of a functional designer eukaryotic chromosome. *Science*, 344 (6179), 55–58.
- 33** Brenner, S. (1973) The genetics of *Caenorhabditis elegans*. *Genetics*, 77, 71–94.
- 34** Johnson, T.E. (2013) 25 years after age-1: genes, interventions and the revolution in aging research. *Exp. Gerontol.*, 48 (7), 640–643.
- 35** Fire, A., Xu, S., Montgomery, M.K., Kostas, S.A., Driver, S.E., and Mello, C.C. (1998) Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegans*. *Nature*, 391 (6669), 806–811.
- 36** White, J.G., Southgate, E., Thomson, J.N., and Brenner, S. (1986) The structure of the nervous system of the nematode *Caenorhabditis elegans*. *Philos. Trans. R. Soc. B*, 314 (1165), 1–340.
- 37** Jarrell, T.A., Wang, Y., Bloniarz, A.E., Brittin, C.A., Xu, M., Thomson, J.N. et al. (2012) The connectome of a decision-making neural network. *Science*, 337 (6093), 437–444.
- 38** C. elegans Sequencing Consortium (1998) Genome sequence of the nematode *C. elegans*: a platform for investigating biology. *Science*, 282, 2012–2018.
- 39** Hillier, L.W., Coulson, A., Murray, J.I., Bao, Z., Sulston, J.E., and Waterston, R.H. (2005) Genomics in *C. elegans*: so many genes, such a little worm. *Genome Res.*, 15 (12), 1651–1660.
- 40** Rose, M. and Charlesworth, B. (1980) A test of evolutionary theories of senescence. *Nature*, 287 (5778), 141–142.
- 41** Nusslein-Volhard, C. and Wieschaus, E. (1980) Mutations affecting segment number and polarity in *Drosophila*. *Nature*, 287 (5785), 795–801.
- 42** Scott, M.P. and Weiner, A.J. (1984) Structural relationships among genes that control development: sequence homology between the Antennapedia, Ultrabithorax, and fushi tarazu loci of *Drosophila*. *Proc. Natl. Acad. Sci. USA*, 81 (13), 4115–4119.
- 43** Gehring, W.J. (1992) The homeobox in perspective. *Trends Biochem. Sci.*, 17 (8), 277–280.
- 44** Adams, M.D., Celniker, S.E., Holt, R.A., Evans, C.A., Gocayne, J.D., Amanatides, P.G. et al. (2000) The genome sequence of *Drosophila melanogaster*. *Science*, 287 (5461), 2185–2195.
- 45** Mouse Genome Sequencing Consortium, Waterston, R.H., Lindblad-Toh, K., Birney, E., Rogers, J., Abril, J.F. et al. (2002) Initial sequencing and comparative analysis of the mouse genome. *Nature*, 420 (6915), 520–562.
- 46** Wakayama, T., Perry, A.C., Zuccotti, M., Johnson, K.R., and Yanagimachi, R. (1998) Full-term development of mice from enucleated oocytes injected with cumulus cell nuclei. *Nature*, 394 (6691), 369–374.

Further Reading

The early days of systems biology: Kitano, H. (2001) *Foundations of Systems Biology*, MIT Press, Cambridge, MA.

The early days of systems biology: Kitano, H. (2002) Systems biology: a brief overview. *Science*, 295 (5560), 1662–1664.

Numbers in cell biology: Flamholz, A., Philips, R., and Milo, R. (2014) The quantified cell. *Mol. Biol. Cell*, 25 (22), 3497–3500.

Numbers in cell biology: Milo, R. and Phillips, R. (2014) *Cell Biology by the Numbers*, Garland Science.

Systemic thinking in cell biology: Lazebnik, Y. (2002) Can a biologist fix a radio? Or, what I learned while studying apoptosis. *Cancer Cell*, 2, 179–182.

Physical constraints on cell function: Dill, K.A., Ghosh, K., and Schmit, J.D. (2011) Physical limits of cells and proteomes. *Proc. Natl. Acad. Sci. USA*, 108 (44), 17876–17882.

Modeling of Biochemical Systems

2

Over the last two decades, formulation of formal (often mathematical) concepts and computational modeling have become more and more familiar in biology, although they have been applied much earlier. Leonardo of Pisa used in his book from 1202 the Fibonacci numbers to describe the growth of a rabbit population. Lotka and Volterra characterized population dynamics of lynx and hares (predators and preys) with their famous equation in the 1920s. Michaels and Menten developed a model to determine the rate for kinetic reactions in 1913. In the mid of the last century, Ludwig von Bertalanffy became famous for his work on biophysics of open systems, thermodynamics of living systems, and the introduction of the notion of “Fließgleichgewicht,” roughly translated with dynamic equilibrium.

Since that time, many different approaches have been introduced and applied for in-depth understanding of biology, to relate independent biological observations to each other (what has the protein content measured in one experiment to do with the mRNA amounts measured independent of another technique?). Concepts from physics and engineering have fruitfully invaded biological research, one example being concepts of control systems for gene regulation. Biology is on its way to a quantitative science like physics, chemistry, or geography. It requires numbers and mathematical predictions to complement biological measurements and to derive useful predictions from them. We can compare it, in some sense, with the weather forecast: Data are combined with models and evaluated by computers to predict tomorrow's weather. The forecast is often not precise, but getting better. However, biology is much more complicated with different species, different cell types making a higher organism, and processes of development and evolution. Nevertheless, many promising results indicate that it is worth trying to provide a mathematical description of biological systems.

2.1 Overview of Common Modeling Approaches for Biochemical Systems

2.2 ODE Systems for Biochemical Networks

- Basic Components of ODE Models
- Illustrative Examples of ODE Models

References

Further Reading

Here, we will give a short and cursory introduction to model concepts and how to formulate a first model for the process of interest. More detailed explanations of construction and analysis procedures for models follow in the later chapters.

2.1 Overview of Common Modeling Approaches for Biochemical Systems

Summary

We give an overview of frequently used modeling approaches in systems biology such as network-based models, rule-based models, or statistical models and discuss their fields of application, strengths, and basic underlying principles. Links to the individual chapters with in-depth explanation, examples, and questions are provided.

Understanding a biological system is not a straight, unidirectional process. It requires understanding of its topology or the structure of the system. It involves analyses of its dynamical behavior and the control mechanisms at play. Also, it requires interpreting how its design and function relate one another in the overall context. The

major guiding principles for model building are as follows: What question is the model supposed to answer? Is it built to explain a surprising observation? Is it built to relate separate observations with each other and with previous knowledge? Is it built to make predictions, for example, about the effect of specific perturbations? Often, a major function of models is to make assumptions about the underlying process explicit and, hence, testable. Only if we have a formal description of the system at hand, we can prove formal relations or falsify certain assumptions.

A mathematical model of a biological system can describe very different aspects and, hence very different approaches are employed. Let us consider the following three major types of computational models, which are further explained below:

- 1) Network-based models
- 2) Rule-based models
- 3) Statistic models

Network-based models describe and analyze properties, states, or dynamics of networks, that is, components and their interactions. Typical and frequently used network-centered modeling frameworks are as follows:

- Systems of ordinary differential equations for biochemical reaction networks
- Stochastic description of biochemical reaction networks and other state change processes, for example, for birth–death processes or reaction networks with small compound numbers
- Boolean models, for example, for gene regulatory networks
- Petri net models, for example, for metabolic networks or for transitions in complex systems

In classical cases, the model has instances for a set of compounds, for example, genes, RNAs, proteins, or small molecules contributing to a process. These compounds can be represented simply by a node or by a set of variables describing their amount or activity or states. The mathematical description suitable for the variables depends on the chosen modeling framework. The topology of the network, that is, the ensemble of all compounds (or nodes of the network) and interactions between them, can be derived from different experiments to identify interactions. These experiments range from classical biochemical methods to more recent high-throughput techniques such as protein–protein interactions by yeast two-hybrid analysis, DNA–protein interactions by chip-on-chip, or gene coexpression profiling (a survey on experimental techniques for systems biology is given in Chapter 14). Since a considerable amount of data is available in publications, an alternative or addition to new experiments is exhaustive literature research,

including text mining as well as systematic screening of databases.

A sensible description of the network topology already allows a set of topological analyses: Are compounds densely or loosely connected? Do we have single important hubs or are all nodes equally well connected? What is the shortest path to get from one node to another node? These questions are answered in Chapter 8. They can provide an understanding of how information is transmitted in a network, how robust it is against changes of its topology (e.g., by knockout of a node or cutting of a connection), or how it may have emerged and changed during evolution.

Based on the completed network topology, it is possible to add more detailed information about the nature of the connections or the nodes. This can be kinetic laws for individual reactions or instructions for combining input information arriving at a node from different edges as in Boolean networks. Again, the type of additional information to add depends on the choice of the model framework, which in turn is largely determined by the question that the model is supposed to answer.

Concerning the choice of modeling framework, we have a number of alternatives. The most important ones are the following:

- 1) For the variables describing the states of the compounds, we can consider either *discrete* or *continuous* values. An example of discrete values is the pair of 0 and 1 used in Boolean networks (Section 7.1) to characterize on/off states of genes due to the presence or absence of transcription factors, respectively. Another example is the set of all natural numbers indicating the number of molecules. Continuous variable values are used to describe compound concentrations or physical properties such as temperature, pressure, or chemical potential.
- 2) For the behavior of the system in time, we may on one side assume that it essentially does not depend on time but is *static* (as, for example, metabolic fluxes in flux balance analysis) (Section 3.3). In *dynamic* networks, on the other hand, states of variables can be updated either on a *continuous* time axis or in *discrete* time steps. Ordinary differential equation (ODE) systems make use of continuous time, while model approaches with discrete variable values typically also employ discrete time schemes.
- 3) The update of states over time can proceed *deterministically*, that is, according to fixed rules that always lead to the same outcome when start conditions are identical. Examples are again ODE systems or classical Boolean networks. Alternatively, state updates may be realized in a *stochastic* fashion, where events occur with a certain probability, leading to different

outcomes for different simulation runs, even under identical initial conditions. Stochastic models are explained in detail in Section 7.2.

Rule-based models or agent-based models represent a different approach of modeling biological phenomena (and, of course, also phenomena in other areas). In rule-based models, every compound of the system can update its state according to a set of rules. For example, a rule could express that protein X is phosphorylated if its kinase Y is activated and its inhibitor Z is not present. Rule-based modeling lists all potential state changes of the individual compounds, but not all potentially occurring states. That is why it can be computationally less demanding than an ODE system, for example, for signaling systems, where compounds may have many degrees of phosphorylation or involve in various complexes. Cellular automata are regular grids of cells (here cells in the sense of abstract location or unit, not of a biological entity), which can assume a finite number of states. States are updated based on rules that depend on the states of neighboring cells. Cellular automata can create complex phenomena, including the classical Game of Life. More complex are the agent-based models, in which all individual compounds such as proteins or cells are considered as autonomous agents with their own rules. The agents move freely in the containing space and update their states according to their rules and the environmental conditions. They can be used to describe many processes such as (i) signaling pathways in models accounting for spatial and structural properties of the cell or (ii) cell states in models describing differentiation or (iii) the interplay of different cells of the immune system or (iv) the interaction of parasites with the occupied host tissue.

In the light of massive data production by the different omics technologies, *statistical models* are very important in systems biology. Statistical models establish relations between measured data and provide a guide to extracting underlying structures of the biological system that gave rise to the data. Examples are the detection of linear or multilinear relationships such as linear regression or ANOVA. One may also perform a comparison of the measured data distribution with well-understood distributions such as binomial distribution, Gaussian distribution, Poisson distribution, and others. Clustering of measurement data is used to find groups of data that behave similarly in some aspects (e.g., in their dynamics). It is frequently applied for the output of networks such as gene expression data and may reveal common regulation patterns. Other methods have been developed to categorize data. An example is support vector machines, which are learning algorithms that are supposed to divide a number of objects into classes with maximal distance.

These classes may or may not reflect biological mechanisms or structures, but often provide a good indication of underlying principles.

If you want to build a theoretical model for a biological process, you would like to use the following short recipe:

- 1) Define the question that the model shall help to answer.
- 2) Seek available information:
 - Read the literature.
 - Look at the available experimental data.
 - Talk to experts in the field.
- 3) Formulate a mental model.
- 4) Decide on the modeling concept (network-based or rule-based, deterministic or stochastic, etc.)
- 5) Formulate the first (simple) mathematical model.
- 6) Test the model performance in comparison to the available data.
- 7) Refine the model, estimate parameters.
- 8) Analyze the system (parameter sensitivity, static and temporal behaviors, etc.)
- 9) Make predictions for scenarios not used to construct the model such as
 - gene knockout or overexpression,
 - application of different stimuli or perturbations.
- 10) Compare predictions and experimental results.

If model predictions and the new experimental data are in agreement, it indicates that the model may have covered correctly important aspects of the described system and can be used to make further predictions. Models are never “right,” but can be appropriate and helpful. If predictions and experimental tests differ, it may be even more interesting. It tells you that important aspects of the biological process have not been understood correctly, not been presented appropriately, or have been completely ignored. The model is falsified (at least in its current state), leading to an interesting journey of finding missing links, alternative explanations, or better parameter sets to explain the observations.

2.2

ODE Systems for Biochemical Networks

Summary

Systems of ordinary differential equations (ODE) are probably the most frequently used approach to model the static and dynamic behaviors of biochemical networks [1]. They employ continuous variable values (mostly concentrations) and continuous time. Since they are important for many aspects presented in this book, they are introduced here briefly. Extensive explanation of analytic and computational evaluation will follow in the later chapters.

2.2.1

Basic Components of ODE Models

To formulate an ODE model for a dynamic biochemical reaction network, we need the following information:

- 1) The basic building blocks are all compounds and all reactions converting these compounds into each other. A list of the reactions and their substrates and products gives us the stoichiometry of the network. This approach is further detailed in Chapter 3.
- 2) The modeler must set the boundary of the system. Which components should the model follow? They become internal components. Which components are not considered relevant for the model? They are completely left out. Which components determine the model behavior, but are not changed by its dynamics? They are called external components. For a metabolic pathway, for example, all metabolites within the pathway may be internal metabolites, while the concentration of a nutrient provided in the medium at fixed experimental conditions may be an external metabolite. All other cellular components may be ignored for that specific study.
- 3) For all reactions that are part of the model, assign kinetic laws (see Chapter 4).
- 4) Determine the values of the kinetic parameters used in the kinetic laws. They can be taken either from databases or literature or they can be fitted to experimental data (Chapter 6).

On the basis of the well-formulated model, we can perform different analysis steps:

- 5) Find out whether the system has a steady state or not. Is the steady state uniquely defined or do we obtain several steady states depending on the parameter values? Calculate steady state concentrations and reaction rates.
- 6) Simulate the time course for a given set of parameter values and initial conditions (tools and techniques are introduced in Chapter 5).
- 7) Analyze the effect of perturbations. The impact of small changes of parameter values is studied by sensitivity analysis and – for biochemical networks – by metabolic control analysis (see Section 4.2). One may also test the effect of complete knockouts of enzymes catalyzing the individual reactions or of knockdowns or overexpression.

2.2.2

Illustrative Examples of ODE Models

To illustrate the steps introduced above, we consider two little models: one for a metabolic network and the other for a small regulatory network.

2.2.2.1 Metabolic Example

Metabolism serves the uptake of nutrients to convert them into energy, mostly in the form of ATP, and into building blocks such as amino acids and lipids. All metabolic reactions are catalyzed by enzymes. The metabolic network in Figure 2.1 resembles the first steps in glycolysis, a major catabolic pathway for the uptake and initial phosphorylation of glucose, which is afterward distributed to other catabolic and anabolic pathways to provide building blocks and energy. Let us call extracellular glucose S_0 and its concentration S_0 . Intracellular glucose is S_1 , singly phosphorylated glucose (glucose-6-phosphate, G6P) is S_2 , and doubly phosphorylated fructose (fructose-1,6-bisphosphate, F16P₂) is S_4 . G6P is provided for other synthetic pathways producing S_3 . Phosphorylation is carried out by consuming ATP (A_3) and converting it into ADP (A_2).

We can describe the dynamics of the network in Figure 2.1a by a set of ordinary differential equations as follows:

$$\begin{aligned}\frac{dS_1}{dt} &= v_1 - v_2, \\ \frac{dS_2}{dt} &= v_2 - v_3 - v_4, \\ \frac{dA_2}{dt} &= -\frac{dA_3}{dt} = v_2 + v_4.\end{aligned}\quad (2.1)$$

When we assign kinetics to the reactions, we can simulate the system. A simple choice of kinetics is mass action kinetics, where the reaction rate is proportional to the concentration of its substrates:

$$v_1 = k_1, \quad v_2 = k_2 \cdot S_1 \cdot A_3, \quad v_3 = k_3 \cdot S_3, \quad v_4 = k_4 \cdot S_2 \cdot A_3. \quad (2.2)$$

You can find more information on kinetic laws in Chapter 4. We will now use this set of equations to simulate the dynamic behavior of the network. Starting with an ATP concentration of 1, an ADP concentration of 0, and zero concentrations of the internal sugars (S_1, S_2), we find that ATP is consumed and ADP is produced. S_1 is produced in an unlimited fashion through the uptake reaction v_1 , but S_2 is produced only as long as ATP is available, afterward it declines. Since there is an unlimited supply of S_0 , the system has no steady state, that is, no state with $dS_i/dt = 0$ ($i = 1, \dots, n$).

A steady state with $dS_1/dt = dS_2/dt = 0$ can be obtained if we consider that ATP and ADP are kept constant by other cellular processes, that is, $dA_3/dt = dA_2/dt = 0$. Then we can consider them as external variables, as shown in Figure 2.1c with the dynamics represented in Figure 2.1d.

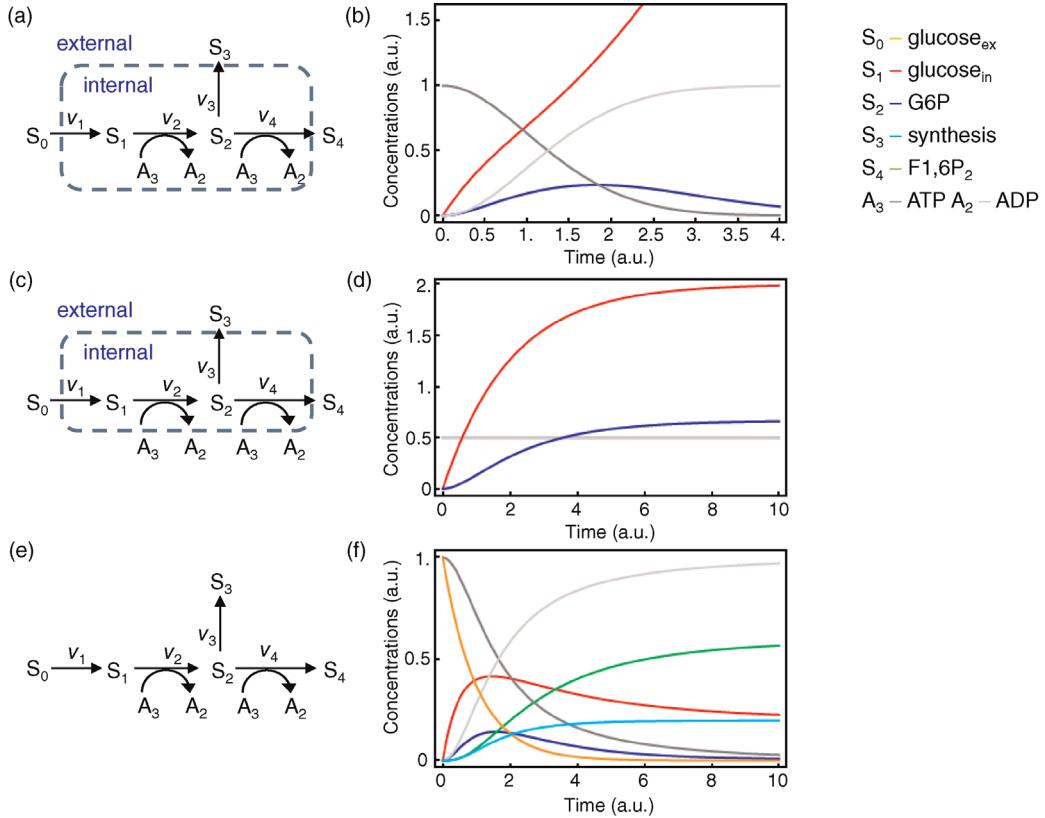


Figure 2.1 Example of a metabolic model. (a) Network representation with S_0 and S_3 considered external and S_1 , S_2 as well as A_2 and A_3 treated as internal variables. (b) Time course resulting from dynamic simulation of the network shown in part (a). (c) The same network as in part (a), but here A_2 and A_3 are also treated as external. (d) Time course resulting from dynamic simulation of the network shown in part (c). (e) Network representation with all components considered as internal and, therefore, dynamic. (f) Dynamics of the network shown in part (e). The dynamics and kinetics are listed in Eqs. (2.1) and (2.2). Parameter values: $k_i = 1$ ($i = 1, \dots, 4$).

If S_0 , S_3 , and S_4 were internal variables that can change ($dS_0/dt = -v_1$, $dS_3/dt = v_3$, $dS_4/dt = v_4$, respectively), then the mass provided by S_0 remains within the system and it approaches a state where S_0 , A_3 , and S_2 decline to 0, while A_2 reaches 1, S_1 and S_4 approach about 0.2, and S_3 about 0.6. However, the model will never reach a true steady state (Figure 2.1e and f).

2.2.2.2 Regulatory Network Example

Stem cell research is of increasing importance in biological research and of great interest in health care. Besides many other promises, it provides the hope that in the future many diseases can be cured by administration of healthy cells of a specific tissue to a diseased person that have been grown out of reprogrammed induced pluripotent stem cells (iPS cells) originating from the same person. The three genes (and gene products) considered as the main regulators of stemness of cells are Oct4, Sox2, and Nanog. They activate each other, but they are controlled by epigenetic marking and by growth factors.

Cellular differentiation is accompanied by hypermethylation of their promoters and by downregulation of their gene expression. In order to create iPS cells, many experimental procedures have been tested. The addition of viral plasmid containing four factors – Sox2, Oct4, c-Myc, and the microRNA Klf4 – that was introduced by Takahashi and Yamanaka in 2006 [2] was most successful.

Here, we will use a simple model to study some basic properties of that system. Let us first assume that Oct4, Sox2, and Nanog stabilize each other. Their expression is suppressed by the epigenetic marking (the DNA methylation), but the proteins also prevent DNA methylation (see Figure 2.2). This mutual inhibition can be described on different levels of detail (e.g., including the joint stabilization of the proteins or not). We will use the following differential equation system that focuses on the collective effect of stemness markers on epigenetic marking and vice versa, that is, the double negative feedback (resulting in a positive feedback) that

each component has on itself:

$$\begin{aligned} \frac{dOSN}{dt} &= \nu_1 - \nu_2 - \nu_3 = k_1 - k_2 \cdot \frac{EM^{n_1}}{(K_1^{n_1} + EM^{n_1})} - k_3 \cdot OSN, \\ \frac{dEM}{dt} &= \nu_4 - \nu_5 - \nu_6 = k_4 - k_5 \cdot \frac{OSN^{n_2}}{(K_2^{n_2} + OSN^{n_2})} - k_6 \cdot EM. \end{aligned} \quad (2.3)$$

OSN denotes the common activity of the stemness markers Oct4, Sox2, and Nanog. EM represents the level of epigenetic marking. The activity of OSN increases

linearly, but EM inhibits it in a fashion described with a Hill function (see Chapter 4). Its basal degradation is proportional to its current concentration. The respective rules hold for EM with OSN as inhibitor. The behavior is illustrated in Figure 2.2. It shows that the system has three steady states. One steady state is unstable, the other two steady states feature either low levels of epigenetic marking and high expression of the stemness factors, indicating that the cell is a stem cell, or high levels of epigenetic marking and low levels of Oct4, Sox2, and Nanog, indicating differentiation. In isolation, the system will always reach one of these states, depending on the initial conditions, and then remain there. It can only be

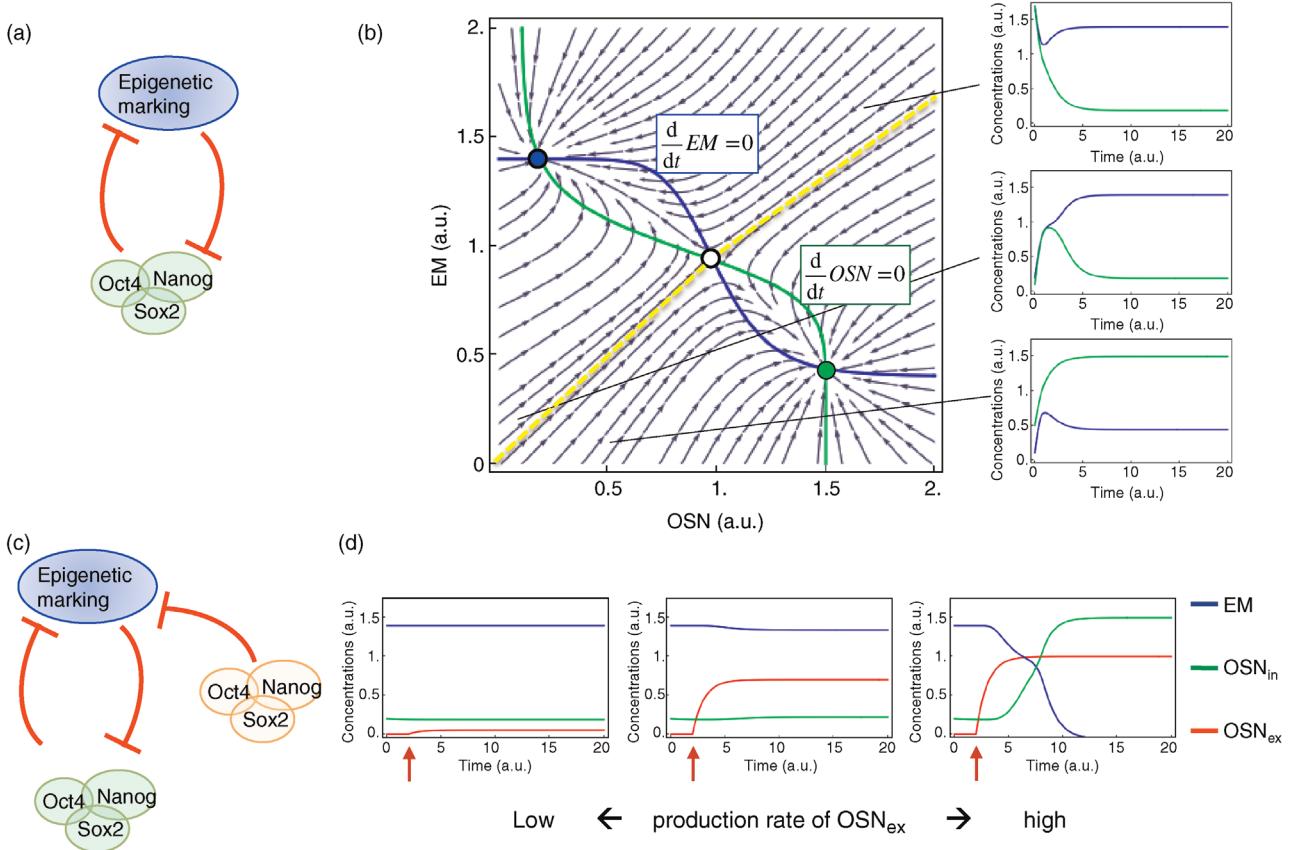


Figure 2.2 Network and dynamics of a model for epigenetic regulation of pluripotency. (a) Epigenetic regulation (EM) and the three factors Oct4, Sox2, and Nanog (OSN) responsible for pluripotency. The dynamics is described with the ordinary differential equations system (2.3). (b) The phase plan showing a plot of EM against OSN represents the three steady states of this system, one stable state for high values of EM (indicated in blue), one stable state for high values of OSN (indicated in green), and one in-between. The state in-between is unstable. The small light blue arrows indicate the actual flow of the system at each point in the phase plan. The green line is the line with no change (nullcline) of OSN ; hence, the flow arrows cross it always vertically. The blue line for steady values of EM is always crossed horizontally. The dashed yellow line is called separatrix since it separates the basin of attraction for the steady state featuring high EM and low OSN (blue dot) from the basin of attraction for the steady state with high OSN and low EM (green dot). The small time plots to the right exemplify that any starting condition within these basins of attraction leads to the respective stable state. (c) The system can be pushed out of its current stable state by supply of another component, here by OSN transcribed from an exogenous vector, which inhibits EM but is not regulated by it. (d) The effect depends on the expression strength of external OSN ; only if it is expressed strongly enough and for sufficient time, it can reverse the cellular decision from high EM to high OSN (and low EM).

moved out of this state by external cues. Under natural conditions, stem cells are forced into differentiation by external signaling compounds such as Wnt or growth factors. When trying to reprogram cells away from the differentiated state toward induced pluripotency, the strategy introduced by Takahashi and Yamanaka favors OSN through the expression from a viral vector. This has the effect that these four compounds are expressed and active, for example, in regulating epigenetic marking, but they are not under epigenetic regulation themselves. If their expression is strong and long enough, they push the cells back into conditions featuring pluripotency with high expression also of endogenous stem cell factors.

Cellular reprogramming with viral vectors has provided many opportunities to study the reprogramming process in detail and determine the contribution of individual regulatory mechanisms, such as cell cycle progression. It is less suited for long-term application in human patients, which is an interesting medical aim, since it implies using viral material and since the uncontrolled expression of pluripotency factors can also lead to unintended side effects such as cancerogenesis. Hence, the search for alternative ways for reprogramming is ongoing, for example, by using small molecules.

References

- 1 Klipp, E., Liebermeister, W., Helbig, A., Kowald, A., and Schaber, J. (2007) Systems biology standards: the community speaks. *Nat. Biotechnol.*, 25, 390–391.
- 2 Takahashi, K. and Yamanaka, S. (2006) Induction of pluripotent stem cells from mouse embryonic and adult fibroblast cultures by defined factors. *Cell*, 126, 663–676.

Further Reading

- Agent-Based Modeling:** An, G., Mi, Q., Dutta-Moscato, J., and Vodovotz, Y. (2009) Agent-based models in translational systems biology. *Wiley Interdiscip. Rev. Syst. Biol. Med.*, 1 (2), 159–171.
- Introduction to Mathematical Concepts in Biology:** Edelstein-Keshet, L. (1988) *Mathematical Models in Biology*, SIAM.
- Introduction to Mathematical Concepts in Biology:** Herbert, S. (2014) *Systems Biology: Introduction to Pathway Modeling*, Ambrosius Publishing.
- Boolean Modeling:** Kauffman, S.A. (1987) Developmental logic and its evolution. *Bioessays*, 6 (2), 82–87.
- Boolean Modeling:** Kauffman, S.A., Peterson, C., Samuelsson, B., and Troein, C. (2003) Random Boolean network models and the yeast transcriptional network. *Proc. Natl. Acad. Sci. USA*, 100, 14796–14799.
- Boolean Modeling:** Keener, J. and Sneyd, J. (1998) *Mathematical Physiology*, Springer, New York.
- Network Motifs:** Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002) Network motifs: simple building blocks of complex networks. *Science*, 298 (5594), 824–827.

Structural Modeling and Analysis of Biochemical Networks

3

In Chapter 2, we outlined general approaches to model biochemical networks. We learned that both the qualitative and quantitative behavior of biological systems depend on the structure of the network and on the kinetics of its individual reactions or processes.

This chapter is devoted to the description and analysis of the structure of biochemical reaction networks. The *basic elements* of a metabolic or regulatory network model are

- 1) the compounds with their concentrations or activities and
- 2) the reactions or transport processes changing the concentrations or activities of the compounds.

In biological environments, reactions are usually catalyzed by enzymes and transport steps are carried out by transport proteins or by pores. Thus, they can be assigned to identifiable biochemical compounds. In the following, we will mainly refer to metabolic networks, that is, networks that are devoted to mass transfer, meaning the conversion of matter from one chemical form into another. This conversion is accompanied by information transfer, but compared with signaling and regulation networks this is not their major function.

The distinction between compounds and converting reactions results in a description of metabolic network with bipartite graphs – compounds being one type of node and reactions or enzymes being the other type of node. A compact representation of such a network is the stoichiometric matrix.

In the following, we explain how to formulate and how to analyze stoichiometric matrices. We will introduce conservation relations for compounds. If the network is considered in steady state, then one can also derive relations

3.1 Structural Analysis of Biochemical Systems

- System Equations
- Information Encoded in the Stoichiometric Matrix
- The Flux Cone
- Elementary Flux Modes and Extreme Pathways
- Conservation Relations – Null Space of N^T

3.2 Constraint-Based Flux Optimization

- Flux Balance Analysis
- Geometric Interpretation of Flux Balance Analysis
- Thermodynamic Constraints
- Applications and Tests of the Flux Optimization Paradigm
- Extensions of Flux Balance Analysis

Exercises

References

Further Reading

between fluxes that are in agreement with the steady-state assumption.

Flux balance analysis (FBA) is a major application of structural analysis of metabolic networks. Based on (i) the steady-state assumption and (ii) further constraints derived from physical, chemical, or other plausible considerations as well as (iii) an objective function such as fast growth of microbial cells, one may derive a sensible distribution of fluxes throughout the whole network. The major approaches will be demonstrated here for small networks to explain the principles. But they have been applied successfully also to larger and even genome-scale networks yielding many scientific findings such as the effect of knockout mutations on metabolic capacity, optimal growth conditions, optimal media compositions, evolutionary adaptation to medium changes, effect of drugs, or identification of suitable drug targets.

3.1 Structural Analysis of Biochemical Systems

Summary

We discuss basic structural and dynamic properties of biochemical reaction networks. We introduce a stoichiometric description of networks and use it to formulate the system (or balance) equations. This will be demonstrated for a number of typical examples. The analysis of the mathematical properties of the stoichiometric matrix can reveal important properties of the reaction system: we can learn how moieties are conserved, even over dynamic changes of the whole system, and how steady-state fluxes are balanced within networks. The search for identifiable pathways in a complex network is also based on the stoichiometric matrix and leads to the concepts of flux cone, elementary flux modes, and extreme pathways.

3.1.1 System Equations

Stoichiometric coefficients denote the proportion of substrate and product molecules involved in a reaction. For example, for the reaction



the stoichiometric coefficients of S_1 , S_2 , and P are -1 , -1 , and 2 , respectively. The assignment of stoichiometric coefficients is not unique. We could also argue that for the production of one mole P , half a mole of each S_1 and S_2 have to be used and, therefore, choose $-1/2$, $-1/2$, and 1 . Or, if we change the direction of the reaction, then we may choose 1 , 1 , and -2 .

The change of concentrations in time can be described using ordinary differential equations (ODEs). For the reaction depicted in Eq. (3.1) and the first choice of stoichiometric coefficients, we obtain

$$\frac{dS_1}{dt} = -v, \quad \frac{dS_2}{dt} = -v, \quad \text{and} \quad \frac{dP}{dt} = 2v. \quad (3.2)$$

This means that the decay of S_1 with rate v is accompanied by the decay of S_2 with the same rate and by the production of P with the double rate.

For a metabolic network consisting of m substances and r reactions, the system dynamics is described by the *system equations* (or *balance equations*, since the balance of substrate production and degradation is considered) [1,2]:

$$\frac{dS_i}{dt} = \sum_{j=1}^r n_{ij} v_j \quad \text{for } i = 1, \dots, m. \quad (3.3)$$

The quantities n_{ij} are the stoichiometric coefficients of the i th metabolite in the j th reaction. Here, we assume that the reactions are the only cause for concentration changes and that no mass flow occurs due to convection or diffusion. External metabolites are not included in the balance equations. These metabolites are not described in the model because they are not considered relevant or their concentrations are kept constant by processes out of the scope of the model. The balance equations (Eq. (3.3)) can also be applied if the system consists of several compartments. In this case, every compound in different compartments has to be considered as an individual compound and transport steps are formally considered as reactions transferring the compound belonging to one compartment into the same compound belonging to the other compartment. Volume differences must be taken into account, since a certain number of molecules moving from one compartment to another change the concentrations in these compartments differently in case of different volumes (see Section 7.3).

The stoichiometric coefficients n_{ij} assigned to the compounds S_i and the reactions v_j can be combined into the *stoichiometric matrix* \mathbf{N} with

$$\mathbf{N} = \{n_{ij}\} \quad \text{for } i = 1, \dots, m \quad \text{and} \quad j = 1, \dots, r,$$

or

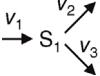
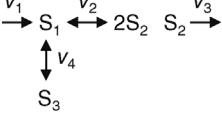
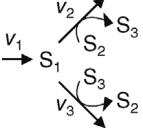
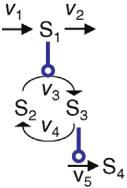
$$\mathbf{N} = \begin{pmatrix} v_1 & v_2 & \cdots & v_r \\ n_{11} & n_{12} & \cdots & n_{1r} \\ n_{21} & n_{22} & \cdots & n_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ n_{m1} & n_{m2} & \cdots & n_{mr} \end{pmatrix} \begin{matrix} S_1 \\ S_2 \\ \vdots \\ S_m \end{matrix}, \quad (3.4)$$

where each column belongs to a reaction and each row to a compound. Table 3.1 provides some examples for reaction networks and their respective stoichiometric matrices.

Note that the stoichiometric matrix \mathbf{N} does not contain information about whether reactions are reversible or irreversible. In order to determine the signs in \mathbf{N} , the direction of the arrows must be assigned, for example, as positive “from left to right” and “from top to bottom.” If the net flow of a reaction proceeds in the opposite direction as the arrow indicates, the value of rate v is negative.

Altogether, the mathematical description of the metabolic system consists of a vector $\mathbf{S} = (S_1, S_2, \dots, S_n)^T$ of concentration values, a vector $\mathbf{v} = (v_1, v_2, \dots, v_r)^T$ of reaction rates, a parameter vector $\mathbf{p} = (p_1, p_2, \dots, p_m)^T$, and the stoichiometric matrix \mathbf{N} . If the system is in steady

Table 3.1 Different reaction networks, their stoichiometric matrices, and the respective system of ODEs.

	<i>Network</i>	<i>Stoichiometric matrix</i>	<i>ODE system</i>
N1	$S_1 + S_2 + S_3 \xrightleftharpoons{v_1} S_4 + 2S_5$	$N = \begin{pmatrix} -1 \\ -1 \\ -1 \\ 1 \\ 2 \end{pmatrix}$	$\frac{dS_1}{dt} = \frac{dS_2}{dt} = \frac{dS_3}{dt} = -v_1$ $\frac{dS_4}{dt} = v_1$ $\frac{dS_5}{dt} = 2v_1$
N2	$v_1 \rightarrow S_1 \xrightarrow{v_2} S_2 \xrightarrow{v_3} S_3 \xrightarrow{v_4} S_4 \xrightarrow{v_5}$	$N = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}$	$\frac{dS_1}{dt} = v_1 - v_2$ $\frac{dS_2}{dt} = v_2 - v_3$ $\frac{dS_3}{dt} = v_3 - v_4$ $\frac{dS_4}{dt} = v_4 - v_5$
N3		$N = (1 \ -1 \ -1)$	$\frac{dS_1}{dt} = v_1 - v_2 - v_3$
N4		$N = \begin{pmatrix} 1 & -1 & 0 & -1 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	$\frac{dS_1}{dt} = v_1 - v_2 - v_4$ $\frac{dS_2}{dt} = 2v_2 - v_3$ $\frac{dS_3}{dt} = v_4$
N5		$N = \begin{pmatrix} 1 & -1 & -1 \\ 0 & -1 & 1 \\ 0 & 1 & -1 \end{pmatrix}$	$\frac{dS_1}{dt} = v_1 - v_2 - v_4$ $\frac{dS_2}{dt} = -v_2 + v_3$ $\frac{dS_3}{dt} = v_2 - v_3$
N6		$N = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\frac{dS_1}{dt} = v_1 - v_2$ $\frac{dS_2}{dt} = v_4 - v_3$ $\frac{dS_3}{dt} = -v_4 + v_5$ $\frac{dS_4}{dt} = v_5$

Note that external metabolites are neither drawn in the network nor included in the stoichiometric matrix. Thin arrows denote reactions and bold arrows denote activation.

state, we can also consider the vector $\mathbf{J} = (J_1, J_2, \dots, J_r)^T$ containing the steady-state fluxes. With these notions, the balance equation reads

$$\frac{d\mathbf{S}}{dt} = \mathbf{N}\mathbf{v}, \quad (3.5)$$

a compact form that is suited for various types of analyses.

3.1.2 Information Encoded in the Stoichiometric Matrix \mathbf{N}

The stoichiometric matrix contains important information about the structure of the metabolic network. Using the stoichiometric matrix, we may calculate which combinations of individual fluxes are possible in steady state (i.e., calculate the admissible steady-state flux space).

We may easily find out dead ends and unbranched reaction pathways. In addition, we may discover the conservation relations for the included reactants.

In steady state, it holds that

$$\frac{dS}{dt} = Nv = 0. \quad (3.6)$$

Note that $\mathbf{0}$ is a vector with length n , that is, $\mathbf{0} = (0, 0, \dots, 0)^T$. The right equality sign in Eq. (3.6) denotes a linear equation system for determination of the rates \mathbf{v} . From linear algebra, it is known that this equation has nontrivial solutions only for $\text{Rank}(N) < r$ (see Section 15.1 for an introduction to linear algebra). A kernel matrix \mathbf{K} fulfilling

$$NK = \mathbf{0} \quad (3.7)$$

expresses the respective linear dependencies between the columns of the stoichiometric matrix [3]. \mathbf{K} consists of $r - \text{Rank}(N)$ basis vectors as columns and can be determined using the Gauss algorithm (see mathematical textbooks). The kernel is not uniquely defined. Multiplication of \mathbf{K} with a regular matrix \mathbf{Q} of appropriate size ($\mathbf{K}' = \mathbf{K} \cdot \mathbf{Q}$, equivalently to linear combination of the columns of \mathbf{K}) yields another valid kernel \mathbf{K}' of N .

Every possible set \mathbf{J} of steady-state fluxes can be expressed as linear combination of the columns \mathbf{k}_i of \mathbf{K} :

$$\mathbf{J} = \sum_{i=1}^{r-\text{Rank}(N)} \alpha_i \cdot \mathbf{k}_i. \quad (3.8)$$

The coefficients must have units corresponding to the units of reaction rates (e.g., mM s^{-1}).

If the entries in a certain row are zero in all basis vectors, we have found an equilibrium reaction. In any steady state, the net rate of this reaction must be zero. For the reaction system N4 in Table 3.1, it holds that $r = 4$ and $\text{Rank}(N) = 3$. Its kernel consists of only one column $\mathbf{K} = (1 \ 1 \ 1 \ 0)^T$. Hence, $v_4 = \sum_{i=1}^4 \alpha_i \cdot 0 = 0$. In any steady state, the rates of production and degradation of S_3 must be equal, thereby leading to zero net change.

If all basis vectors contain the same entries for a set of rows, this indicates an unbranched reaction path. In each steady state, the net rate of all respective reactions is equal.

Up to now, we have not been concerned about (ir)reversibility of reactions in the network. The irreversibility of a reaction does not affect the stoichiometric matrix. However, it has consequences for the choice of basis vectors \mathbf{k}_i for the kernel \mathbf{K} . A set of basis vectors must be chosen to satisfy the signs of fluxes when calculated by Eq. (3.8).

Example 3.1

For the network N2 in Table 3.1, we have $r = 5$ reactions and $\text{Rank}(N) = 4$. The kernel matrix contains just $1 = 5 - 4$ basis vectors, which are multiples of $\mathbf{k} = (1 \ 1 \ 1 \ 1 \ 1)^T$. This means that in steady state the flux through all reactions must be equal.

Network N3 comprises $r = 3$ reactions and has $\text{Rank}(N) = 1$. Each representation of the kernel matrix contains $3 - 1 = 2$ basis vectors, for example,

$$\mathbf{K} = (\mathbf{k}_1 \ \mathbf{k}_2) \quad \text{with} \quad \mathbf{k}_1 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}, \quad \mathbf{k}_2 = \begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \quad (3.9)$$

and for the steady-state flux holds

$$\mathbf{J} = \alpha_1 \cdot \mathbf{k}_1 + \alpha_2 \cdot \mathbf{k}_2. \quad (3.10)$$

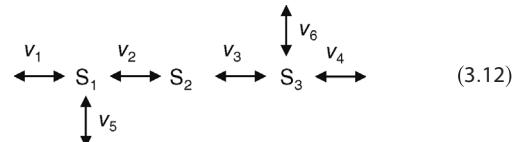
Network N6 can present a small signaling cascade. It has five reactions and $\text{Rank}(N) = 3$. Two basis vectors of the kernel are

$$\begin{aligned} \mathbf{k}_1 &= (1 \ 1 \ 0 \ 0 \ 0)^T, \\ \mathbf{k}_2 &= (0 \ 0 \ 1 \ 1 \ 0)^T. \end{aligned} \quad (3.11)$$

If we calculate the possible steady-state fluxes according to Eq. (3.10), we can easily see that in every steady state it holds that production and degradation of S_1 are balanced ($J_1 = J_2$) and that the fluxes through the cycle are equal ($J_3 = J_4$). In addition, J_5 must be equal to zero, otherwise S_4 would accumulate. One could prevent the last effect by also including the degradation of S_4 into the network.

Example 3.2

Consider the reaction scheme



The system comprises $r = 6$ reactions. The stoichiometric matrix reads

$$\mathbf{N} = \begin{pmatrix} 1 & -1 & 0 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 1 \end{pmatrix}$$

with $\text{Rank}(N) = 3$. Thus, the kernel matrix is spanned by three basis vectors, for example, $\mathbf{k}_1 = (1 \ 1 \ 1 \ 0 \ 0 \ -1)^T$, $\mathbf{k}_2 = (1 \ 0 \ 0 \ 0 \ 1 \ 0)^T$, and $\mathbf{k}_3 = (-1 \ -1 \ -1 \ -1 \ 0 \ 0)^T$. The entries for the second and third reactions are always equal; thus, in any steady state, the fluxes through reactions 2 and 3 must be equal.

3.1.3 The Flux Cone

The stoichiometric analysis of biochemical network analysis can be modified by considering only irreversible reactions (e.g., by splitting reversible reactions into two irreversible ones). Based on such a unidirectional representation, the basis vectors (Eq. (3.8)) form a convex cone in the flux space. This mapping relates stoichiometric analysis to the concepts of convex geometry as follows. The steady-state assumption requires that a flux vector is an element of the null space of the stoichiometric matrix \mathbf{N} spanned by matrix \mathbf{K} . A row of \mathbf{K} can be interpreted as a hyperplane in flux space. The intersection of all these hyperplanes forms the null space. Provided that all reactions are unidirectional or irreversible, the intersection of the null space with the semipositive orthant of the flux space forms a polyhedral cone, the flux cone. The intersection procedure results in a set of rays or edges starting at 0, which fully describe the cone. The edges are represented by vectors and any admissible steady state of the system is a positive combination of these vectors. An illustration is presented in Figure 3.1.

3.1.4 Elementary Flux Modes and Extreme Pathways

A stringent definition of the term “pathway” in a metabolic network is not straightforward. A descriptive definition of a pathway is a set of reactions that are linked by common metabolites. Typical examples include glycolysis or different amino acid synthesis pathways. More detailed inspection of metabolic maps such as the *Boehringer chart* [4] shows that metabolism is highly interconnected

and better addressed as a network. Pathways that are known for a long time from biochemical experience are already hard to recognize, and it is even harder to find out new pathways, for example, in metabolic maps that have been reconstructed from sequenced genomes of bacteria.

The problem of clearly identifying functional pathways has been elaborated in the concepts of *elementary flux modes* [3,5–10] and *extreme pathways* [11–14]. In both cases, the stoichiometry of a metabolic network is investigated to discover which direct routes are possible that lead from one external metabolite to another external metabolite. Both approaches use the steady-state assumption and take into account that some reactions are reversible, while others are irreversible. Despite these two constraints, we still obtain too many solutions, especially for larger networks. For elementary flux modes, this problem is solved by the requirement that they cannot be further decomposed, while extreme fluxes are bound to the generating vectors of the flux cone, as explained below.

We start with defining a *flux mode* \mathbf{M} . It is the set of flux vectors that represent direct routes through the network between external metabolites. In mathematical terms, it is defined as the set

$$\mathbf{M} = \{\mathbf{v} \in \mathbb{R}^r | \mathbf{v} = \lambda \mathbf{v}^*, \lambda > 0\}, \quad (3.13)$$

where \mathbf{v}^* is an r -dimensional vector (unequal to the null vector) fulfilling two conditions:

- 1) the steady-state condition $\mathbf{N}\mathbf{v} = \mathbf{0}$, that is, Eq. (3.6), and
- 2) sign restriction, that is, the flux directions in \mathbf{v}^* fulfill the prescribed irreversibility relations. \mathbf{v}^{irr} denotes the subvector of \mathbf{v}^* that contains only nonnegative fluxes.

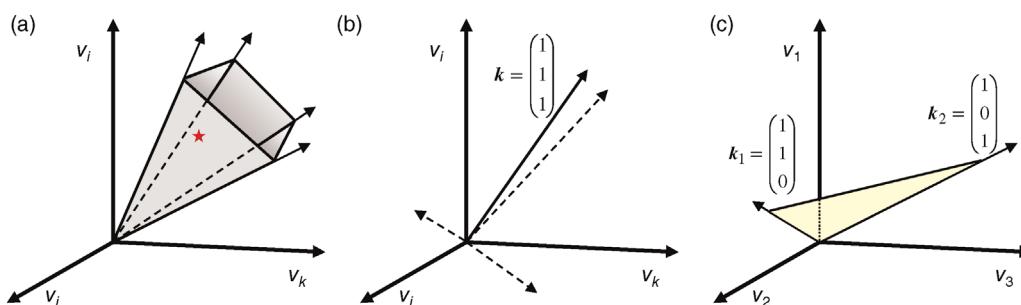


Figure 3.1 Flux cone: schematic representation of the subspace of feasible steady states within the space spanned by all positive-valued vectors for rates of irreversible reactions, v_i , $i = 1, \dots, r$. Only three dimensions are shown. Feasible solutions are linear combinations of basis vectors of matrix \mathbf{K} (see text). (a) Illustrative representation of the flux cone for a higher dimensional system (with $r - \text{Rank}(\mathbf{N}) = 4$). The basis vectors of \mathbf{K} are rays starting at the origin. The line connecting the four rays indicates possible limits for real flux distributions set by constraints. The little asterisk indicates one special feasible solution for the fluxes. (b) The flux cone for an unbranched reaction chain of arbitrary length, such as the network N2 in Table 3.1, is just a ray since \mathbf{K} is represented by a single basis vector containing only 1s. (c) The flux cone for network N3 in Table 3.1 is the plane spanned by the basis vectors $\mathbf{k}_1 = (1 \ 1 \ 0)^T$ and $\mathbf{k}_2 = (1 \ 0 \ 1)^T$.

A flux mode \mathbf{M} comprising \mathbf{v} is called reversible if the set \mathbf{M}' comprising $-\mathbf{v}$ is also a flux mode.

A flux mode is an *elementary flux mode* if it uses a minimal set of reactions and cannot be further decomposed, that is, the vector \mathbf{v} cannot be represented as non-negative linear combination of two vectors that fulfill conditions (1) and (2) but contain more zero entries than \mathbf{v} . An elementary flux mode is a minimal set of enzymes that could operate at steady state, with all the irreversible reactions used in the appropriate direction. The number of elementary flux modes is at least as high as the number of basis vectors of the null space. The set of elementary

flux modes is uniquely defined. Pfeiffer *et al.* [6] developed a software ("Metatool") to calculate the elementary flux modes for metabolic networks.

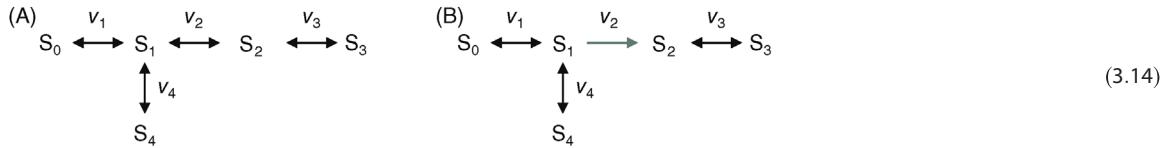
A flux mode is an *extreme pathway* if

- 1) all reactions are nonnegative, that is, $\mathbf{v} = \mathbf{v}^{\text{irr}}$;
- 2) it belongs to the edges of the flux cone, which also means that it represents a basis vector of \mathbf{K} .

To achieve the first, reversible reactions are broken down into their forward and backward components and exchange fluxes have to be defined in the appropriate direction. This way, the set of extreme pathways is a

Example 3.3

The systems (A) and (B) differ by the fact that reaction 2 is either reversible or irreversible.



The elementary flux modes connect the external metabolites S_0 and S_3 , S_0 and S_4 , or S_3 and S_4 . The stoichiometric matrix and the flux modes for case (A) and case (B) are

$$\mathbf{N} = \begin{pmatrix} 1 & -1 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{pmatrix}, \quad \mathbf{v}^A = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ -1 \\ -1 \\ -1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ 1 \\ 1 \end{pmatrix}, \quad \text{and}$$

$$\mathbf{v}^B = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ 0 \\ 0 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \\ -1 \end{pmatrix}. \quad (3.15)$$

The possible routes are illustrated in Figure 3.2.

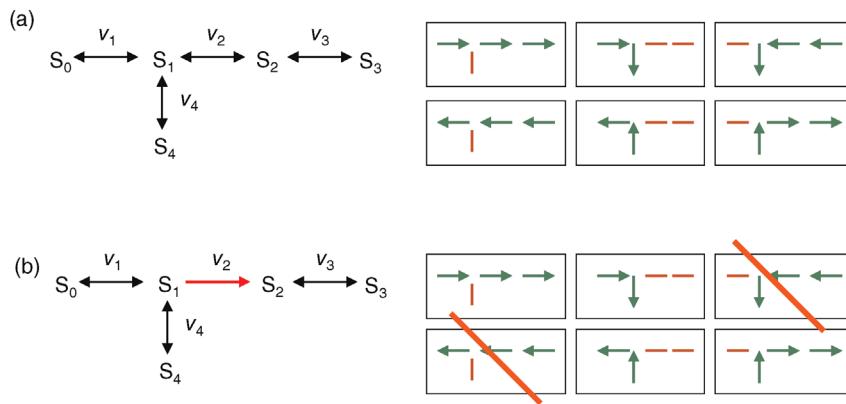


Figure 3.2 Schematic representation of elementary flux modes for the reaction network depicted in Eq. (3.14).

subset of the set of elementary flux modes and the extreme pathways are systemically independent.

Elementary flux modes and extreme pathways can be used to understand the range of metabolic pathways in a network, to test a set of enzymes for production of a desired product and detect nonredundant pathways, to reconstruct metabolism from annotated genome sequences and analyze the effect of enzyme deficiency, to reduce drug effects, and to identify drug targets. A specific application, the flux balance analysis will be explained in Section 3.2.1.

3.1.5

Conservation Relations – Null Space of N^T

If a chemical entity is neither added to nor removed from the reaction system (neither produced nor degraded), its total concentration remains constant. This also holds if the substance interacts with other compounds by forming complexes.

For the mathematical derivation of the conservation relations [3], we consider a matrix G fulfilling

$$GN = 0. \quad (3.16)$$

Due to Eq. (3.5), it follows

$$G\dot{S} = GNv = 0. \quad (3.17)$$

Integrating this equation leads directly to the conservation relations

$$GS = \text{constant}. \quad (3.18)$$

The number of independent rows of G is equal to $n - \text{Rank}(N)$, where n is the number of metabolites in the system. G^T is the kernel matrix of N^T ; hence, it has similar properties to K . Matrix G can also be found using the Gauss algorithm. It is not unique, but every linear combination of its rows is again a valid solution (equivalent to a premultiplication of G with a regular matrix of appropriate size, i.e., $PG = G'$). There exists a simplest representation $G = (G_0 \ I_{n-\text{Rank}(N)})$. Finding this representation may be helpful for a simple statement of conservation relations, but this may necessitate renumbering and reordering of metabolite concentrations (see below).

Importantly, conservation relations can be used to simplify the system of differential equations $\dot{S} = Nv$ describing the dynamics of our reaction system. The idea is to eliminate linear dependent differential equations and to replace them by appropriate algebraic equations. Below the procedure is explained systematically [2].

First we have to reorder the rows in the stoichiometric matrix N as well as in the concentration vector S such that a set of independent rows is on top and the dependent rows are at the bottom. Then the matrix N

Example 3.4

Consider a set of two reactions comprising a kinase and a phosphatase reaction



The metabolite concentration vector reads $S = (ATP \ ADP)^T$, and the stoichiometric matrix is $N = \begin{pmatrix} -1 & 1 \\ 1 & -1 \end{pmatrix}$ yielding $G = (1 \ 1)$. From the condition $GS = \text{constant}$, it follows that $ATP + ADP = \text{constant}$. Thus, we have a conservation of adenine nucleotides in this system. The actual values of $ATP + ADP$ must be determined from the initial conditions.

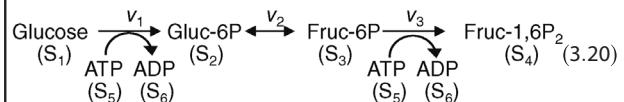
is split into the independent part N_{indep} and the dependent part N' and a *link matrix* L is introduced in the following way:

$$N = \begin{pmatrix} N_{\text{indep}} \\ N' \end{pmatrix} = LN_{\text{indep}} = \begin{pmatrix} I_{\text{Rank}(N)} \\ L' \end{pmatrix} N_{\text{indep}}. \quad (3.22)$$

$I_{\text{Rank}(N)}$ is the identity matrix of size $\text{Rank}(N)$. The

Example 3.5

For the following model of the upper part of glycolysis



the stoichiometric matrix N (note the transpose!) and a possible representation of the conservation matrix G are given by

$$N^T = \begin{pmatrix} -1 & 1 & 0 & 0 & -1 & 1 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & -1 & 1 \end{pmatrix} \quad \text{and}$$

$$G = \begin{pmatrix} 2 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix}. \quad (3.21)$$

The interpretation of the second and third rows of G is straightforward, showing the conservation of adenine nucleotides (g_2 , $ADP + ATP = \text{constant}$) and the conservation of sugars (g_3), respectively. The interpretation of the first row is less intuitive. If we construct the linear combination $g_4 = -g_1 + 3 \cdot g_2 + 2 \cdot g_3 = (0 \ 1 \ 1 \ 2 \ 3 \ 2)$, we find the conservation of phosphate groups.

differential equation system may be rewritten accordingly:

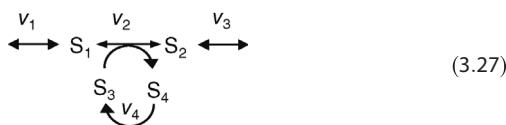
$$\dot{\mathbf{S}} = \begin{pmatrix} \dot{S}_{\text{indep}} \\ \dot{S}_{\text{dep}} \end{pmatrix} = \begin{pmatrix} I_{\text{Rank}(N)} \\ L' \end{pmatrix} N_{\text{indep}} v \quad (3.23)$$

and the dependent concentrations fulfill

$$\dot{S}_{\text{dep}} = L' \cdot \dot{S}_{\text{indep}}. \quad (3.24)$$

Example 3.6

For the reaction system



the stoichiometric matrix, the reduced stoichiometric matrix, and the link matrix read

$$N = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{pmatrix},$$

$$N_{\text{indep}} = \begin{pmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & -1 & 0 & 1 \end{pmatrix},$$

$$L = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -1 \end{pmatrix}, \quad L' = (0 \ 0 \ -1).$$

The conservation relation $S_3 + S_4 = \text{constant}$ is expressed by $G = (0 \ 0 \ 1 \ 1)$. The ODE system

$$\dot{S}_1 = v_1 - v_2,$$

$$\dot{S}_2 = v_2 - v_3,$$

$$\dot{S}_3 = v_4 - v_2,$$

$$\dot{S}_4 = v_2 - v_4$$

can be replaced by the algebro-differential equation system

$$\dot{S}_1 = v_1 - v_2,$$

$$\dot{S}_2 = v_2 - v_3,$$

$$\dot{S}_3 = v_4 - v_2,$$

$$S_3 + S_4 = \text{constant},$$

which has one differential equation less.

Integration leads to

$$S_{\text{dep}} = L' \cdot S_{\text{indep}} + \text{constant}. \quad (3.25)$$

This relation is fulfilled during the entire time course. Thus, we may replace the original system by a reduced differential equation system

$$\dot{S}_{\text{indep}} = N_{\text{indep}} v \quad (3.26)$$

supplemented with the set of algebraic equations (3.25).

Eukaryotic cells contain a variety of organelles such as the nucleus, mitochondria, or vacuoles, which are separated by membranes. Reaction pathways may cross these compartment boundaries. If a substance S occurs in two different compartments, for example, in the cytosol and in mitochondria, the respective concentrations can be assigned to two different variables, S^{C1} and S^{C2} . Formally, the transport across the membrane can be considered as a reaction with rate v . It is important to note that both compartments have different volumes V^{C1} and V^{C2} . Thus, transport of a certain amount of S with rate v from compartment C1 into the compartment C2 changes the concentrations differently:

$$V^{C1} \cdot \frac{d}{dt} S^{C1} = -v \quad \text{and} \quad V^{C2} \cdot \frac{d}{dt} S^{C2} = v, \quad (3.28)$$

where $V \cdot S$ denotes substance amount in moles. Compartment models are discussed in more detail in Section 7.3.

3.2 Constraint-Based Flux Optimization

Summary

Flux balance analysis, an optimality-based method for flux prediction, is one of the most popular modeling approaches for metabolic systems. Flux optimization methods do not describe *how* a certain flux distribution is realized (by kinetics or enzyme regulation), but *which* flux distribution is optimal for the cell – for example, providing the highest rate of biomass production at a limited inflow of external nutrients. This allows us to predict flux distributions without the need for a kinetic description. From a species' genome sequence, the metabolic network can be roughly predicted [15–17]. Even if we do not know anything about the enzyme kinetics, we can infer which metabolites the network can produce and which precursors are needed to produce biomass. Given a number of nutrients and a hypothesized optimality requirement, for example, for fast biomass production, we can try to predict an optimal flux distribution in the network.

3.2.1

Flux Balance Analysis

Flux balance analysis [18–23] investigates the theoretical capabilities and modes of metabolism by imposing a number of constraints on the metabolic flux distributions. A first constraint on the flux vector \mathbf{v} is set by the assumption of a steady state $d\mathbf{S}/dt = \mathbf{N}\mathbf{v}$, where \mathbf{N} again denotes the stoichiometric matrix (see Section 3.1). A second constraint stems from consideration of thermodynamics, assuming the irreversibility of certain reactions under physiological conditions. A third constraint acknowledges the limited capacity of enzymes and imposes upper bounds on certain reaction fluxes. For example, in the case of a Michaelis–Menten-type enzyme, the reaction rate is limited by the maximal rate set by the enzyme's concentration and turnover number. In general, the latter constraints on individual metabolic fluxes read

$$v_i^{\min} \leq v_i \leq v_i^{\max}. \quad (3.29)$$

Partial inhibition of enzymes can be modeled by tighter maximality constraints leading to reduced maximal rates [24]. Together, these constraints confine the steady-state fluxes to a feasible set, but usually do not yield a unique solution. Thus, as a fourth requirement, an optimality assumption is added: the flux distribution has to maximize an objective function $f(\mathbf{v})$

$$\max_{\mathbf{v}} f(\mathbf{v}) = \sum_{i=1}^r c_i v_i, \quad (3.30)$$

where the coefficients c_i represent weights for the individual rates v_i . Examples of such objective functions are maximization of ATP production, minimization of nutrient uptake, maximal yield of a desired product, maximal biomass yield, or a combination thereof. The above assumptions lead to an optimization problem with constraints

$$\begin{aligned} \mathbf{c}^T \mathbf{v} &= \text{max}, \\ \mathbf{N}\mathbf{v} &= \mathbf{0}, \\ \begin{pmatrix} \mathbf{I} \\ -\mathbf{I} \end{pmatrix} &\geq \begin{pmatrix} \mathbf{v}^{\min} \\ -\mathbf{v}^{\max} \end{pmatrix}. \end{aligned} \quad (3.31)$$

The latter inequality represents the constraints (3.29). This is a standard problem of linear programming that can be solved by the simplex algorithm.

3.2.2

Geometric Interpretation of Flux Balance Analysis

We can imagine possible flux distributions \mathbf{v} as points in a multidimensional flux space. Each dimension in this

space corresponds to a reaction in the network and represents its reaction velocity (see Figure 3.1).

The stationary fluxes, which are constrained by the linear equations $\mathbf{N}\mathbf{v} = \mathbf{0}$, form a hyperplane. If all individual fluxes are constrained by lower and upper bounds, the resulting region of allowed flux distributions is a convex polyhedron (Figure 3.1): any combination $\lambda\mathbf{v}_\alpha + (1-\lambda)\mathbf{v}_\beta$ of two allowed flux distributions \mathbf{v}_α and \mathbf{v}_β with $0 \leq \lambda \leq 1$ is again an allowed flux distribution. Flux balance analysis maximizes a linear function within this polyhedron. The optimum has to lie somewhere on the surface: depending on the direction of the fitness gradient, either there is a unique optimum in a corner of the polyhedron or the fitness function is maximized on an entire surface. Figure 3.3 illustrates a set of cases for the network N3 introduced in Table 3.1.

3.2.3

Thermodynamic Constraints

Flux balance analysis requires that flux patterns are stationary, but it does not check whether they are thermodynamically feasible. According to the second law of thermodynamics, chemical reactions at constant pressure p and temperature T need to be driven by a consumption of Gibbs free energy $G(p, T)$. The Gibbs free energy of a biochemical system is associated with the amount and types of molecules:

$$G(p, T) = \sum_i m_i \mu_i, \quad (3.32)$$

where μ_i and m_i denote the chemical potential and the amount (in moles) of substance i , respectively. (Note that we use symbol m for amounts instead of the typical n to avoid confusion with the stoichiometric coefficients considered below.) A chemical reaction will change the substance amounts according to the stoichiometric coefficients n_{ij} (Eq. (3.4)), and the resulting Gibbs free energy change (in kJ per mole reaction events) can be expressed by the chemical potential difference

$$\Delta_r G_j(p, T) = \sum_i \mu_i n_{ij}. \quad (3.33)$$

The symbol Δ_r denotes changes associated with chemical reactions, and the negative value $A_j = -\Delta_r G_j(p, T)$ is also called *reaction affinity*. The vector of chemical potential differences satisfies the Wegscheider condition

$$(\Delta_r \mathbf{G})^T \mathbf{K} = 0, \quad (3.34)$$

where \mathbf{K} is again the kernel matrix of the stoichiometric matrix \mathbf{N} , satisfying $\mathbf{NK} = \mathbf{0}$. This tells us that the drop

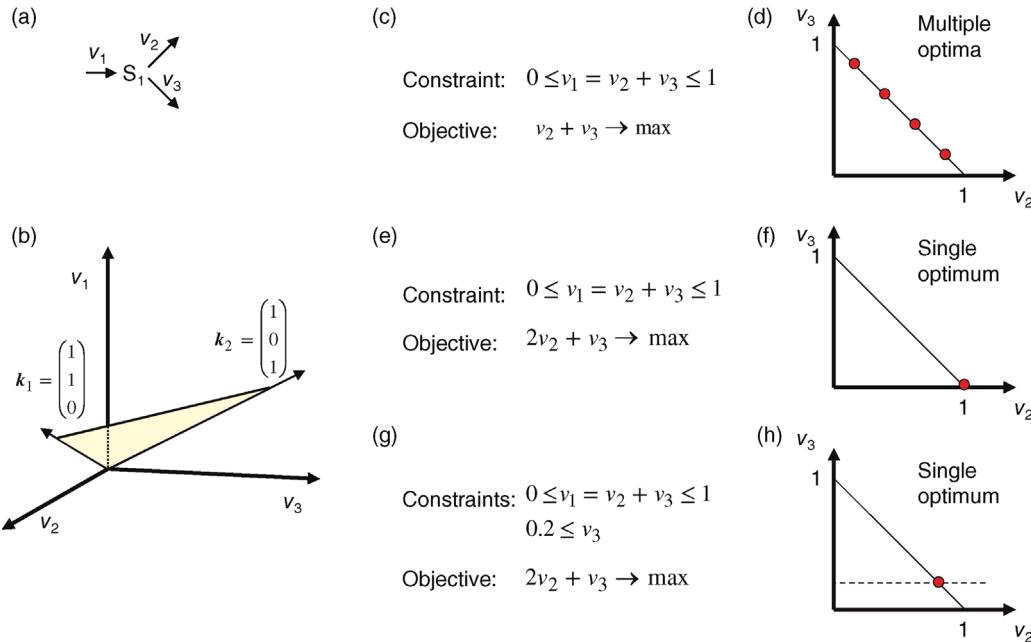


Figure 3.3 Constraint-based flux balance analysis. (a) Simple branched network. (b) The kernel vectors k_1 and k_2 span a plane of admissible steady-state fluxes in the flux space, that is, a two-dimensional flux cone. All solutions must lie on this plane. (c–h) Different examples for constraints, objectives, and resulting optimal fluxes. (c and d) The constraint of an upper and a lower bound for v_1 and the objective of maximizing v_1 (here equivalently to maximizing $v_2 + v_3$) yield infinitely many optimal solutions lying on the line $v_2 + v_3 = 1$. (e and f) If the objective is instead $2v_2 + v_3 \rightarrow \max$, we obtain a single optimal solution at $v_2 = 1$, $v_3 = 0$. (g and h) The stronger constraint $0.2 \leq v_3$ shifts the solution to $v_2 = 0.8$, $v_3 = 0.2$.

(or increase) in free energy between two compounds in a network is independent of the reaction path that is taken to get from one compound to the other one. According to the second law of thermodynamics, the Gibbs free energy must decrease in any occurring reaction. For a forward reaction, the difference of chemical potentials (Equation 3.33) must be negative and the reaction affinity must be positive. In general, for a reaction j , we obtain the condition

$$\sum_i \mu_i n_{ij} v_j \leq 0. \quad (3.35)$$

Therefore, a given flux pattern $\mathbf{v} = (v_1, \dots, v_r)^T$ is only feasible if condition (3.35) can be satisfied by some vector $(\mu_1, \dots, \mu_n)^T$ of chemical potentials. This condition can be tested using the stoichiometric matrix [24].

Flux balance analysis does not require that condition (3.35) is fulfilled and can therefore lead to incorrect flux signs. This problem can be avoided by predefining some of the flux directions, which will restrict the solution space in advance. *Energy balance analysis* [25,26], in contrast, ensures thermodynamically feasible fluxes by a joint optimization of the fluxes v_j and the chemical potential differences $\Delta_r G_j$. Besides the conditions (3.31), it imposes the additional requirements (3.34) and (3.35), which leads to an optimization problem with nonlinear constraints.

The chemical potentials are related not only to the flux directions, but also to the substance concentrations: for an ideal mixture (with vanishing mixing enthalpy), the chemical potential of substance i at pressure p and temperature T reads

$$\mu_i(p, T) = \mu_i^0(p, T) + RT \ln S_i, \quad (3.36)$$

where S_i denotes the concentration of metabolite i in mM. If the standard chemical potentials μ_i^0 are known (e.g., calculated by the group contribution method [27,28]), Eq. (3.36) translates to constraints between flux directions and substance concentrations. These constraints can be used to determine ranges of possible substance concentrations or to check whether measured concentrations are in agreement with the assumed fluxes [29,30].

3.2.4

Applications and Tests of the Flux Optimization Paradigm

Constraint-based methods such as flux balance analysis allow us to predict the metabolic fluxes and the biomass production (corresponding to the maximal growth rate) under different external conditions, for example,

availability of nutrients. The predictions can be used to simulate dynamically the growth of cell populations and the consumption of nutrients [19]. By comparing the predictions of FBA (biomass production or metabolic fluxes) with experimental data, one can check the assumed network structure for errors (e.g., missing reactions) and test Boolean models of gene regulation [31]. For instance, a low predicted growth rate would indicate that the organism is not viable. By testing the networks for deletion mutants, essential genes can be predicted [32]. The accuracy of such predictions (92% for a *Escherichia coli* model [16]) can be used as a quality score to check the consistency of the model structure and to point to missing reactions.

This approach implies that flux patterns in wild-type and mutant cells, under different external conditions, are optimized for the same general objective function. However, a study by Schuetz *et al.* [33] indicates that cells may optimize different objectives depending on the experimental conditions: metabolic fluxes in the central metabolism of *E. coli* cells were compared with predictions based on 11 alternative (linear and nonlinear) objective functions. Under glucose limitation in continuous cultures, cells seemed to maximize their yield of ATP or biomass per glucose consumed. Unlimited growth on glucose in respiring batch cultures, on the other hand, was best described by assuming a maximization of ATP production divided by the sum of squared reaction fluxes. This modified objective can be interpreted as a compromise between large ATP production and small enzymatic costs.

Such considerations of minimal effort had been formulated before in the *principle of minimal fluxes* [34]. Large reaction velocities require large amounts of enzymes, which put a burden on the cell. If the cost of enzyme production plays a role, cells will benefit from flux patterns that require less enzyme production, so pathways that do not contribute to biomass production (or whatever quantity is maximized) should be shut off to save energy and material. The principle of minimal fluxes assumes that the flux pattern has to meet some functional requirement – for example, to yield a prescribed rate of biomass production – while the magnitudes of individual reaction fluxes are minimized.

Even if we accept the assumption of optimality in general, constraint-based methods (i) do not explain by which biological mechanisms changes in flux distributions are actually achieved (e.g., inherent dynamics of the metabolic network, transcriptional regulation), (ii) do not cover the trade-off between cost and benefit of enzyme production, (iii) rely, instead, on ad-hoc assumptions, for example, about maximal fluxes, and (iv) assume a steady state and do not account for dynamic objectives such as fast adaptation to changes of supply and demand.

We will further address the issue of optimality of biological systems and the application to detect organization principles in Chapter 11.

3.2.5 Extensions of Flux Balance Analysis

Metabolic networks are embedded in a highly regulated cellular environment. A number of studies and approaches extended flux balance analysis to address specific biological observations, to integrate molecular and cellular information, and to accommodate further general principles. They are briefly summarized here.

3.2.5.1 Minimization of Metabolic Adjustments

Minimization of metabolic adjustments (MoMA) is a flux-based analysis technique similar to FBA [35]. It is based on the same stoichiometric constraints, but the demand of optimal growth flux for mutants is relaxed. Instead, MoMA provides an approximate solution for a suboptimal growth flux state, which is nearest in flux distribution to the unperturbed state. It is based on the assumption that in case of a knockout of an enzyme-coding gene metabolic fluxes undergo a minimal redistribution with respect to the flux configuration of the wild type.

The mathematical formulation of these requirements leads to a quadratic programming problem:

$$\begin{aligned} \text{Constraint : } & \mathbf{Nv} = \mathbf{0} \\ \text{Objective : } & \| \mathbf{v}_w - \mathbf{v}_d \|^2 \rightarrow \min \end{aligned} \quad (3.37)$$

with \mathbf{v}_w presenting the fluxes of the wild type and \mathbf{v}_d the fluxes for the gene deletion mutant. Figure 3.4 shows a comparison of FBA and MoMA for an illustrative example.

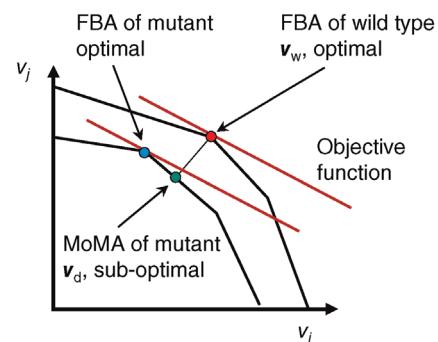


Figure 3.4 The principle of MoMA in comparison with FBA. The coordinates represent two selected fluxes, the outer black line the feasible state space for the wild type and the inner black line the feasible state space for the knockout mutant. The red line shows the objective function. FBA for the wild type yields the solution indicated by the red dot, while FBA for the mutant would result in the solution at the blue dot. MoMA requests that the distance between wild-type FBA solution and the newly attained state is minimal. This is given by the flux distribution marked by the green dot.

3.2.5.2 Flux Variability Analysis

Flux balance analysis often provides not a unique solution but the objective may be achieved by a whole range of alternate optimal solutions. Flux variability analysis (FVA) provides the ranges of possible flux through each individual reaction that is compatible with the steady-state and optimality conditions for the system as a whole [36]. A potential interpretation of the result is that a reaction with a small range of possible fluxes is more important for the functioning of the metabolism than reactions with a wide range of allowed flux solutions. FastFVA is an implementation based on an enhanced algorithm providing solutions significantly faster [37]. Figure 3.5 provides an illustration.

3.2.5.3 Dynamic FBA

Dynamic FBA is an extension of FBA for situations where changes in the metabolic network are relevant and an attempt to adapt the model to these changes over time [38]. This is achieved by a relaxation of the strict assumptions underlying steady-state analysis. The problem can be formulated either as a dynamic optimization problem starting at a given set of initial conditions for the metabolite concentrations or as a static optimization for a set of time intervals. In the second case, the time course is obtained by re-running FBA repeatedly with changing conditions. Since initial metabolite concentrations are given, the update rules also allow to calculate metabolite time profiles in a linearized fashion, that is, $S_i(t + \Delta T) = S_i(t) + Nv\Delta T$ for time steps of duration ΔT .

3.2.5.4 Regulatory FBA

Metabolic networks are subject to external and internal changes and their dynamics are influenced both by the available amount of nutrients and by the expression of genes coding for the metabolic enzymes. To accommodate the transcriptional regulation of enzymes in the FBA framework, regulatory events may impose temporary, adjustable constraints on the solution space, such as

$$v_k(t) = 0, \quad \text{when } t_1 \leq t \leq t_2, \quad (3.38)$$

instead of the constant constraints formulated in Eq. (3.29) [39]. The regulatory constraints change the shape of the accessible solution space. Consider, for example, the network N3 given in Table 3.1 and illustrated in Figure 3.3. If we assume a knockout of the gene coding for the enzyme of the second reaction and, hence, set $v_2(t) = 0$ for a certain period, only solution $k_2 = (1 \ 0 \ 1)^T$ would remain. This is equivalent to say that $v_1 = v_3$ or that the entire flux goes through only one of the branches.

For more complex networks with mutual influence of genes on each other, the transcriptional regulatory structure can be described with Boolean logic (Section 7.1), assigning a value of 1 to an expressed gene and a value of 0 to a nonexpressed gene. The relation between genes and their expression dynamics is then described with Boolean rules, that is, combinations of operators such as AND, OR, or NOT. A schematic of the two interconnected networks is represented in Figure 3.6. This approach has been used to study

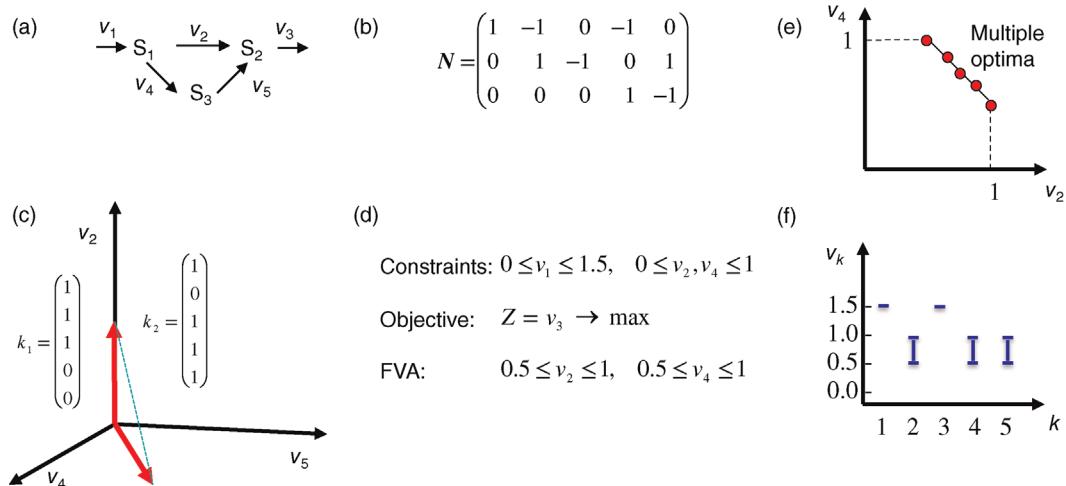


Figure 3.5 Flux variability analysis. For the example network in (a) with the given stoichiometry in (b), we find two representations of the kernel vector \mathbf{K} represented in (c). The optimization problem formulated in (d) results in multiple solutions where the objective function Z is maximal; fluxes v_2 and v_4 can still vary between 0.5 and 1; however, their sum must equal 1.5, that is, the maximal input flux v_1 . (e) shows the multiple optima in a phase plane spanned by the fluxes v_2 and v_4 . (f) shows the variability of fluxes in conditions that maximize the objective function under the given constraints.

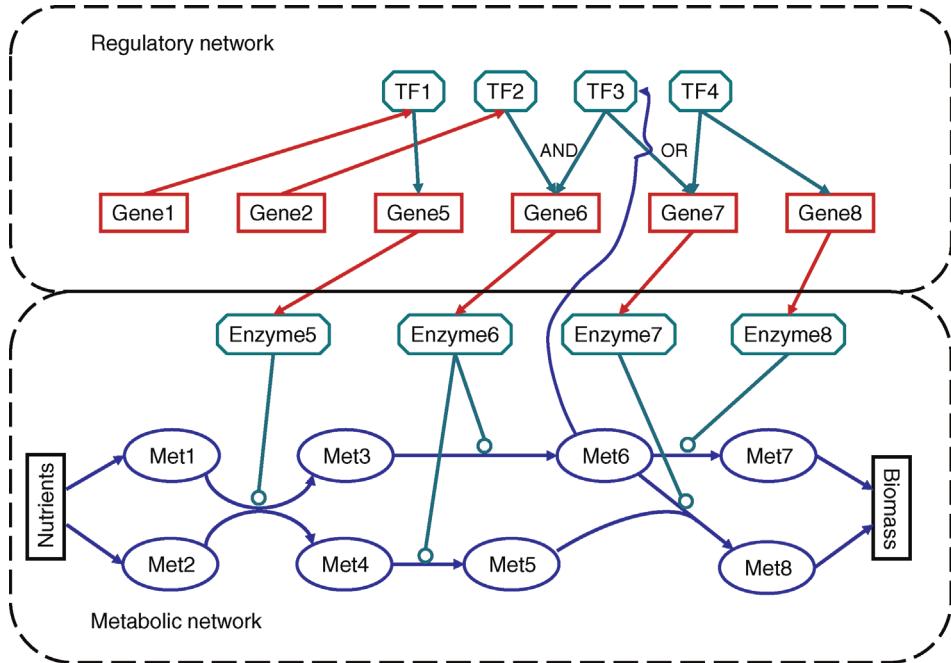


Figure 3.6 Schematic representation of an integrated metabolic and regulatory network. TF: transcription factors; Met: metabolites. Genes code for transcription factors and metabolic enzymes, which in turn catalyze the metabolic reactions. The purpose of the metabolic network is to produce biomass from the nutrients.

whether microorganisms such as *E. coli* can live (i.e., produce sufficient biomass) when living on exhaustible carbon sources and in which order the carbon sources are consumed. Extended analyses integrated large-scale

metabolic models with regulatory models for gene expression [40] and for signaling [41] in *E. coli* to study gene expression variability and response to stimuli within the FBA framework.

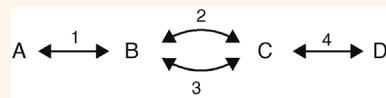
Exercises

- 1) A canonical view of the upper part of glycolysis starts with glucose and comprises the following reactions (in brackets: possible abbreviations): The enzyme hexokinase (HK, E₁) phosphorylates glucose (Gluc, S₁) to glucose-6-phosphate (G6P, S₂) under consumption of ATP (S₅) and production of ADP (S₆). The enzyme phosphoglucoisomerase (PGI, E₂) converts glucose-6-phosphate to fructose-6-phosphate (F6P, S₃). The enzyme phosphofructokinase (PFK, E₃) phosphorylates F6P a second time to yield fructose-1,6-bisphosphate (F1,6BP, S₄). The enzyme fructose bisphosphatase catalyzes the reverse reaction (E₄).
 - a) Sketch the reaction network and formulate a set of differential equations (without specifying the kinetics of the individual reactions).
- 2) a) Write down the sets of differential equations for the networks N1–N6 given in Table 3.1 without specifying their kinetics.
 - b) Formulate the stoichiometric matrix N. What is the rank of N?
 - c) Calculate steady-state fluxes (matrix K) and conservation relations (matrix G).
 - d) Compare your results with Example 3.5.
- 3) a) Determine the rank of the stoichiometric matrices, independent steady-state fluxes, and conservation relations.

Do all systems have a (nontrivial) steady state?
- 3) Inspect networks N3 and N4 in Table 3.1. Can you find elementary flux modes? Use an available tool (e.g., Metatool) to check.

- 4) Consider the branch point in Figure 3.2a, with reactions $A \rightarrow X$, $B \rightarrow X$, and $X \rightarrow C$. The concentrations of A , B , and C are fixed and the reactions are irreversible. (a) Assume upper bounds $v_1 \leq 1$, $v_2 \leq 2$ and the fitness function $f(v) = v_3$. Write down the corresponding linear programming problem and compute the resulting flux distribution. (b) Assume, in addition, an upper bound $v_3 \leq 1$. Draw the allowed region in flux space and determine the optimal flux distribution.

- 5) Show that a circular conversion flux $A \rightarrow B \rightarrow C \rightarrow A$ is thermodynamically unfeasible. Consider the following reaction scheme



with fixed concentrations of A and B and balanced metabolites C and D . Determine all flux distributions that are both stationary and thermodynamically feasible.

References

- 1 Glansdorff, P. and Prigogine, I. (1971) *Thermodynamic Theory of Structure, Stability and Fluctuations*, Wiley-Interscience, London.
- 2 Reder, C. (1988) Metabolic control theory: a structural approach. *J. Theor. Biol.*, 135, 175–201.
- 3 Heinrich, R. and Schuster, S. (1996) *The Regulation of Cellular Systems*, Chapman & Hall, New York.
- 4 Michal, G. (1999) *Biochemical Pathways*, Spektrum Akademischer Verlag, Heidelberg.
- 5 Schilling, C.H., Schuster, S., Palsson, B.O., and Heinrich, R. (1999) Metabolic pathway analysis: basic concepts and scientific applications in the post-genomic era. *Biotechnol. Prog.*, 15, 296–303.
- 6 Pfeiffer, T., Sanchez-Valdenebro, I., Nuno, J.C., Montero, F., and Schuster, S. (1999) METATOOL: for studying metabolic networks. *Bioinformatics*, 15, 251–257.
- 7 Schuster, S., Dandekar, T., and Fell, D.A. (1999) Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends Biotechnol.*, 17, 53–60.
- 8 Schuster, S., Fell, D.A., and Dandekar, T. (2000) A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nat. Biotechnol.*, 18, 326–332.
- 9 Schuster, S., Hilgetag, C., Woods, J.H., and Fell, D.A. (2002) Reaction routes in biochemical reaction systems: algebraic properties, validated calculation procedure and example from nucleotide metabolism. *J. Math. Biol.*, 45, 153–181.
- 10 Schuster, S., Hilgetag, C., and Fell, D.A. (1994) Detecting elementary modes of functioning in metabolic networks, in *What Is Controlling Life?* (eds E. Gnaiger, F.N. Gellerich, and M. Wyss), Innsbruck University Press, Innsbruck, pp. 103–105.
- 11 Schilling, C.H. and Palsson, B.O. (2000) Assessment of the metabolic capabilities of *Haemophilus influenzae* Rd through a genome-scale pathway analysis. *J. Theor. Biol.*, 203, 249–283.
- 12 Schilling, C.H., Letscher, D., and Palsson, B.O. (2000) Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. *J. Theor. Biol.*, 203, 229–248.
- 13 Wiback, S.J. and Palsson, B.O. (2002) Extreme pathway analysis of human red blood cell metabolism. *Biophys. J.*, 83, 808–818.
- 14 Schilling, C.H. and Palsson, B.O. (1998) The underlying pathway structure of biochemical reaction networks. *Proc. Natl. Acad. Sci. USA*, 95, 4193–4198.
- 15 Reed, J.L., Famili, I., Thiele, I., and Palsson, B.O. (2006) Towards multidimensional genome annotation. *Nat. Rev. Genet.*, 7, 130–141.
- 16 Feist, A.M., Henry, C.S., Reed, J.L., Krummenacker, M., Joyce, A.R. et al. (2007) A genome-scale metabolic reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Mol. Syst. Biol.*, 3, 121.
- 17 Thiele, I. and Palsson, B.O. (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nat. Protoc.*, 5, 93–121.
- 18 Fell, D.A. and Small, J.R. (1986) Fat synthesis in adipose tissue. An examination of stoichiometric constraints. *Biochem. J.*, 238, 781–786.
- 19 Varma, A. and Palsson, B.O. (1994) Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type *Escherichia coli* W3110. *Appl. Environ. Microbiol.*, 60, 3724–3731.
- 20 Varma, A. and Palsson, B.O. (1994) Predictions for oxygen supply control to enhance population stability of engineered production strains. *Biotechnol. Bioeng.*, 43, 275–285.
- 21 Edwards, J.S. and Palsson, B.O. (2000) Metabolic flux balance analysis and the *in silico* analysis of *Escherichia coli* K-12 gene deletions. *BMC Bioinform.*, 1, 1.
- 22 Edwards, J.S. and Palsson, B.O. (2000) The *Escherichia coli* MG1655 *in silico* metabolic genotype: its definition, characteristics, and capabilities. *Proc. Natl. Acad. Sci. USA*, 97, 5528–5533.
- 23 Ramakrishna, R., Edwards, J.S., McCulloch, A., and Palsson, B.O. (2001) Flux-balance analysis of mitochondrial energy metabolism: consequences of systemic stoichiometric constraints. *Am. J. Physiol. Regul. Integr. Comp. Physiol.*, 280, R695–R704.
- 24 Price, N.D., Schellenberger, J., and Palsson, B.O. (2004) Uniform sampling of steady-state flux spaces: means to design experiments and to interpret enzymopathies. *Biophys. J.*, 87, 2172–2186.
- 25 Beard, D.A., Babson, E., Curtis, E., and Qian, H. (2004) Thermodynamic constraints for biochemical networks. *J. Theor. Biol.*, 228, 327–333.
- 26 Beard, D.A., Liang, S.D., and Qian, H. (2002) Energy balance for analysis of complex metabolic networks. *Biophys. J.*, 83, 79–86.
- 27 Mavrovouniotis, M.L. (1990) Group contributions for estimating standard Gibbs energies of formation of biochemical compounds in aqueous solution. *Biotechnol. Bioeng.*, 36, 1070–1082.
- 28 Jankowski, M.D., Henry, C.S., Broadbelt, L.J., and Hatzimanikatis, V. (2008) Group contribution method for thermodynamic analysis of complex metabolic networks. *Biophys. J.*, 95, 1487–1499.
- 29 Henry, C.S., Jankowski, M.D., Broadbelt, L.J., and Hatzimanikatis, V. (2006) Genome-scale thermodynamic analysis of *Escherichia coli* metabolism. *Biophys. J.*, 90, 1453–1461.
- 30 Kummel, A., Panke, S., and Heinemann, M. (2006) Putative regulatory sites unraveled by network-embedded thermodynamic analysis of metabolome data. *Mol. Syst. Biol.*, 2, 2006.0034.

- 31** Covert, M.W. and Palsson, B.O. (2002) Transcriptional regulation in constraints-based metabolic models of *Escherichia coli*. *J. Biol. Chem.*, 277, 28058–28064.
- 32** Wodke, J.A., Puchalka, J., Lluch-Senar, M., Marcos, J., Yus, E. et al. (2013) Dissecting the energy metabolism in *Mycoplasma pneumoniae* through genome-scale metabolic modeling. *Mol. Syst. Biol.*, 9, 653.
- 33** Schuetz, R., Kuepfer, L., and Sauer, U. (2007) Systematic evaluation of objective functions for predicting intracellular fluxes in *Escherichia coli*. *Mol. Syst. Biol.*, 3, 119.
- 34** Holzhutter, H.G. (2004) The principle of flux minimization and its application to estimate stationary fluxes in metabolic networks. *Eur. J. Biochem.*, 271, 2905–2922.
- 35** Segre, D., Vitkup, D., and Church, G.M. (2002) Analysis of optimality in natural and perturbed metabolic networks. *Proc. Natl. Acad. Sci. USA*, 99, 15112–15117.
- 36** Mahadevan, R. and Schilling, C.H. (2003) The effects of alternate optimal solutions in constraint-based genome-scale metabolic models. *Metab. Eng.*, 5, 264–276.
- 37** Gudmundsson, S. and Thiele, I. (2010) Computationally efficient flux variability analysis. *BMC Bioinform.*, 11, 489.
- 38** Mahadevan, R., Edwards, J.S., and Doyle, F.J., 3rd (2002) Dynamic flux balance analysis of diauxic growth in *Escherichia coli*. *Biophys. J.*, 83, 1331–1340.
- 39** Covert, M.W., Schilling, C.H., and Palsson, B. (2001) Regulation of gene expression in flux balance models of metabolism. *J. Theor. Biol.*, 213, 73–88.
- 40** Shlomi, T., Eisenberg, Y., Sharan, R., and Ruppin, E. (2007) A genome-scale computational study of the interplay between transcriptional regulation and metabolism. *Mol. Syst. Biol.*, 3, 101.
- 41** Covert, M.W., Xiao, N., Chen, T.J., and Karr, J.R. (2008) Integrating metabolic, transcriptional regulatory and signal transduction models in *Escherichia coli*. *Bioinformatics*, 24, 2044–2050.

Further Reading

Definition of metabolic pathways: Schuster, S., Dandekar, T., and Fell, D.A. (1999) Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends Biotechnol.*, 17 (2), 53–60 (Review).

Definition of metabolic pathways: Schuster, S., Fell, D.A., and Dandekar, T. (2000) A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nat. Biotechnol.*, 18 (3), 326–332.

Analysis of the stoichiometric matrix: Heinrich, R. and Schuster, S. (1996) *The Regulation of Cellular Systems*, Chapman & Hall, New York.

Constraint-based analysis of metabolic networks: Varma, A. and Palsson, B. (1993) Metabolic flux balancing: basic concepts, scientific and practical use. *Biotechnol. Bioeng.*, 12, 994–998.

Genome-scale network reconstruction and methods: Herrgard, M.J. et al. (2008) A consensus yeast metabolic network reconstruction obtained from a community approach to systems biology. *Nat. Biotechnol.*, 26 (10), 1155–1160.

Genome-scale network reconstruction and methods: Thiele, I. and Palsson, B.O. (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nat. Protoc.*, 5, 93–121.

Applications of flux balance analysis: Ibarra, R.U., Edwards, J.S., and Palsson, B.O. (2002) *Escherichia coli* K-12 undergoes adaptive evolution to achieve *in silico* predicted optimal growth. *Nature*, 420 (6912), 186–189.

Applications of flux balance analysis: Kuepfer, L., Sauer, U., and Blank, L.M. (2005) Metabolic functions of duplicate genes in *Saccharomyces cerevisiae*. *Genome Res.*, 15 (10), 1421–1430.

Kinetic Models of Biochemical Networks: Introduction

4

4.1

Reaction Kinetics and Thermodynamics

Summary

Kinetic modeling of metabolic reactions has a long tradition and forms the basis of many complex models for metabolic and regulatory networks. In this chapter, we will make you familiar with the basic concept of kinetic models for specific reactions. We introduce the mass action rate law, Michaelis–Menten kinetics, and different extended, applied, or contemporary modeling approaches. The role of the major parameters, K_m and V_{max} is explained. K_m and V_{max} are also related to the parameters of single reaction steps. We will show how you can derive and apply more advanced kinetic expressions. The effect of modifiers – activators and inhibitors – is shown for different kinetic mechanisms. Thermodynamic laws determine and limit the dynamic behavior and the steady state of kinetic systems; therefore, thermodynamic foundations and constraints are briefly introduced.

4.1.1

Kinetic Modeling of Enzymatic Reactions

Deterministic kinetic modeling of individual biochemical reactions has a long history. The Michaelis–Menten model for the rate of an irreversible one-substrate reaction is an integral part of biochemistry and has recently celebrated its centenary. The K_m value is a major characteristic of the interaction between enzyme and substrate. Biochemical reactions are catalyzed by enzymes, that is, specific proteins or ribonucleic acids, which often function in complex with cofactors. They have a catalytic center, are usually highly specific, and remain unchanged by the reaction. One enzyme molecule can catalyze

4.1 Reaction Kinetics and Thermodynamics

- Kinetic Modeling of Enzymatic Reactions
- The Law of Mass Action
- Reaction Thermodynamics
- Michaelis–Menten Kinetics
- Regulation of Enzyme Activity by Effectors
- Generalized Mass Action Kinetics
- Approximate Kinetic Formats
- Convenience Kinetics and Modular Rate Laws

4.2 Metabolic Control Analysis

- The Coefficients of Control Analysis
- The Theorems of Metabolic Control Theory
- Matrix Expressions for Control Coefficients
- Upper Glycolysis as Realistic Model Example
- Time-Dependent Response Coefficients

Exercises

References

Further Reading

thousands of reactions per second (this so-called turnover number ranges from 10^2 to 10^7 s^{-1}). Enzyme catalysis leads to a rate acceleration of about 10^6 up to 10^{12} -fold compared to the uncatalyzed, spontaneous reaction.

The basic quantities are the concentration S of a substance S , that is, the number n of molecules (or, alternatively, moles) of this substance per volume V , and the rate v of a reaction, that is, the change of concentration S per time t . This type of modeling is macroscopic and phenomenological, compared to the microscopic approach, where single molecules and their interactions are considered. Chemical and biochemical kinetics rely on the assumption that the reaction rate v at a certain point in time and space can be expressed as a unique function of

the concentrations of all substances at this point in time and space. Classical enzyme kinetics assumes for sake of simplicity a spatial homogeneity (the “well-stirred” test tube) and no direct dependency of the rate on time:

$$\nu(t) = \nu(S(t)). \quad (4.1)$$

In more advanced modeling approaches paving the way for whole cell modeling, spatial inhomogeneities are taken into account. Spatial modeling pays tribute to the fact that many components are membrane bound and that cellular structures hinder the free movement of molecules. But, in most cases one can assume that diffusion is rapid enough to allow for an even distribution of all substances in space.

4.1.2 The Law of Mass Action

Biochemical kinetics is based on the mass action law, introduced by Guldberg and Waage in the nineteenth century [1–3]. It states that the reaction rate is proportional to the probability of a collision of the reactants. This probability is in turn proportional to the concentration of reactants to the power of the molecularity, which is the number in which the molecule species enter the reaction. For a simple reaction such as



the reaction rate reads

$$\nu = \nu_+ - \nu_- = k_+ S_1 \cdot S_2 - k_- P^2, \quad (4.3)$$

where ν is the net rate, ν_+ and ν_- are the rates of the forward and backward reactions, respectively, and k_+ and k_- are the *kinetic or rate constants*, that is, the respective proportionality factors.

The molecularity is 1 for S_1 and S_2 and 2 for P . If we measure the concentration in moles per liter ($\text{mol} \cdot \text{l}^{-1}$ or M) and the time in seconds (s), then the rate has the unit $\text{M} \cdot \text{s}^{-1}$. Accordingly, the rate constants for bimolecular reactions have the unit $\text{M}^{-1} \cdot \text{s}^{-1}$. Rate constants for monomolecular reactions have the dimension s^{-1} .

The general mass action rate law for a reaction transforming m_i substrates with concentrations S_i into m_j products with concentrations P_j reads

$$\nu = \nu_+ - \nu_- = k_+ \prod_{i=1}^{m_i} S_i^{n_i} - k_- \prod_{j=1}^{m_j} P_j^{n_j}, \quad (4.4)$$

where n_i and n_j denote the respective molecularities of S_i and P_j in this reaction.

The equilibrium constant K_{eq} (we will also use the simpler symbol q) characterizes the ratio of substrate and product concentrations in equilibrium (S_{eq} and P_{eq}), that

is, the state where the thermodynamic affinity vanishes and where the forward and backward rates become equal. The rate constants are related to K_{eq} in the following way:

$$K_{\text{eq}} = \frac{k_+}{k_-} = \frac{\prod_{j=1}^{m_j} P_j^{n_j}}{\prod_{i=1}^{m_i} S_i^{n_i}}. \quad (4.5)$$

The relation between the thermodynamic and the kinetic description of biochemical reactions will be outlined in Section 4.1.3.

The equilibrium constant for the reaction given in Eq. (4.2) is $K_{\text{eq}} = P_{\text{eq}}^2 / (S_{1,\text{eq}} \cdot S_{2,\text{eq}})$. The dynamics of the concentrations far from equilibrium is described by the ODEs

$$\frac{d}{dt} S_1 = \frac{d}{dt} S_2 = -\nu \quad \text{and} \quad \frac{d}{dt} P = 2\nu. \quad (4.6)$$

The time course of S_1 , S_2 , and P is obtained by integration of these ODEs (see Section 15.2).

4.1.3 Reaction Thermodynamics

Biochemical reactions in isolation or as part of a larger reaction network are governed by the laws of

Example 4.1

The kinetics of a simple decay like

$$S \rightarrow \dots \quad (4.7)$$

is described by $\nu = kS$ and $dS/dt = -kS$. Integration of this ODE from time $t = 0$ with the initial concentration S_0 to an arbitrary time t with concentration $S(t)$, $\int_{S_0}^S dS/S = - \int_{t=0}^t kdt$, yields the temporal expression $S(t) = S_0 e^{-kt}$ (Figure 4.1).

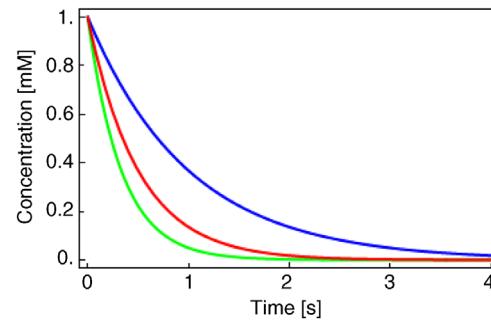


Figure 4.1 Exponential decay of a compound as described in Eq. (4.7). The initial concentration is $S_0 = 1 \text{ mM}$. The dynamics are shown for three different values of k : $k = 1 \cdot \text{s}^{-1}$ (blue), $k = 2 \cdot \text{s}^{-1}$ (red), and $k = 3 \cdot \text{s}^{-1}$ (green).

thermodynamics. This means that they cannot create or destroy energy, they can only convert it or store it in chemical bonds or release it from there. An important purpose of metabolism is to extract energy from nutrients, which is necessary for the synthesis of molecules, growth, and proliferation. We distinguish between energy-supplying reactions, energy-demanding reactions, and energetically neutral reactions. The principles of reversible and irreversible thermodynamics and their application to chemical reactions allow understanding of energy circulation in the cell.

A biochemical process is characterized by the direction of the reaction, by whether it occurs spontaneously or not, and by the position of the equilibrium. The first law of thermodynamics, that is, the law of energy conservation, tells us that the total energy of a closed system remains constant during any process. The second law of thermodynamics states that a process occurs spontaneously only if it increases the total entropy of the system. Unfortunately, entropy is usually not directly measurable. A more suitable measure is the Gibbs free energy G , which is the energy capable of carrying out work under isotherm-isobar conditions, that is, at constant temperature and constant pressure. The change of the Gibbs free energy is given as

$$\Delta G = \Delta H - T\Delta S, \quad (4.8)$$

where ΔH is the change in enthalpy, ΔS is the change in entropy, and T is the absolute temperature in Kelvin. ΔG is a measure for the driving force, the spontaneity of a chemical reaction. The reaction proceeds spontaneous under release of energy, if $\Delta G < 0$ (exergonic process). If $\Delta G > 0$, then the reaction is energetically not favorable and will not occur spontaneously (endergonic process). $\Delta G = 0$ implies that the system has reached its equilibrium. Endergonic reactions may proceed if they obtain energy from a strictly exergonic reaction by energetic coupling. In tables, Gibbs free energy is usually given for standard conditions (ΔG^0), that is, for a concentration of the reaction partners of 1 M, a temperature of $T = 298$ K, and, for gaseous reactions, a pressure of $p = 98.1$ kPa = 1 atm. The unit is kJ mol^{-1} . Gibbs free energy differences satisfy a set of relations as follows. The Gibbs free energy difference for a reaction can be calculated from the balance of free energies of formation of its products and substrates:

$$\Delta G = \sum G_p - \sum G_s. \quad (4.9)$$

The enzyme cannot change the Gibbs free energies of the substrates and products of a reaction, neither their difference, but it changes the way the reaction proceeds microscopically, the so-called reaction path, thereby

lowering the activation energy for the reaction. The *transition state theory* explains this as follows. During the course of a reaction, the metabolites must pass one or more transition states of maximal free energy, in which bonds are solved or newly formed. The transition state is unstable; the respective molecule configuration is called an activated complex. It has a lifetime of around one molecule vibration, $10^{-14}\text{--}10^{-13}$ s, and it can hardly be experimentally verified. The difference ΔG^\ddagger of Gibbs free energy between the reactants and the activated complex determines the dynamics of a reaction: the higher this difference, the lower the probability that the molecules may pass this barrier and the lower the rate of the reaction. The value of ΔG^\ddagger depends on the type of altered bonds, on steric, electronic, or hydrophobic demands, and on temperature.

Figure 4.2 presents a simplified view of the reaction course of the noncatalyzed reaction and with an enzyme. The substrate and the product are situated in local minima of the free energy; the active complex is assigned to the local maximum. The Gibbs free energy difference ΔG is proportional to the logarithm of the equilibrium constant K_{eq} of the respective reaction:

$$\Delta G = -RT \ln K_{\text{eq}}, \quad (4.10)$$

where R is the gas constant, $8.314 \text{ J mol}^{-1} \text{ K}^{-1}$. The value of ΔG^\ddagger corresponds to the kinetic constant k_+ of the forward reaction (Eqs. (4.3)–(4.5)) by $\Delta G^\ddagger = -RT \ln k_+$, while $\Delta G^\ddagger + \Delta G$ is related to the rate constant k_- of the backward reaction.

The interaction of the reactants with an enzyme may alter the reaction path and, thereby, lead to lower values of ΔG^\ddagger as well as higher values of the kinetic constants. However, the enzyme will not change the equilibrium

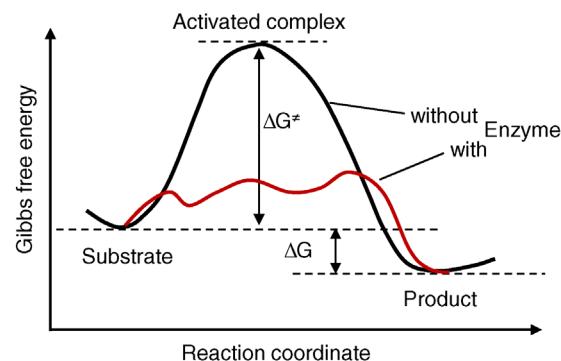


Figure 4.2 Change of Gibbs free energy along the course of a reaction. The substrate and the product are situated in local minima of the free energy; the active complex is assigned to the local maximum. The enzyme may change the reaction path and thereby lower the barrier of Gibbs free energy.

Table 4.1 Values of ΔG^0 and K_{eq} for some important reactions^a.

Reaction	$\Delta G^0/(k\text{J mol}^{-1})$
$2\text{H}_2 + \text{O}_2 \rightarrow 2\text{H}_2\text{O}$	-474
$2\text{H}_2\text{O}_2 \rightarrow 2\text{H}_2\text{O} + \text{O}_2$	-99
$\text{PP}_i + \text{H}_2\text{O} \rightarrow 2\text{P}_i$	-33.49
$\text{ATP} + \text{H}_2\text{O} \rightarrow \text{ADP} + \text{P}_i$	-30.56
Glucose-6-phosphate + $\text{H}_2\text{O} \rightarrow$ Glucose + P_i	-13.82
Glucose + $\text{P}_i \rightarrow$ Glucose-6-phosphate + H_2O	+13.82
Glucose-1-phosphate \rightarrow Glucose-6-phosphate	-7.12
Glucose-6-phosphate \rightarrow Fructose-6-phosphate	+1.67
Glucose + 6 $\text{O}_2 \rightarrow 6\text{CO}_2 + 6\text{H}_2\text{O}$	-2890

^a Source: ZITAT: Lehninger, A.L. Biochemistry, 2nd edition, New York, Worth, 1975, p. 397.

constant of the reaction. The Gibbs free energy may assume several local minima and maxima along the path of reaction. They are related to unstable intermediary complexes. Values for the difference of free energy for some biologically important reactions are given in Table 4.1. Note that the free energy differences always refer to specific standard concentrations.

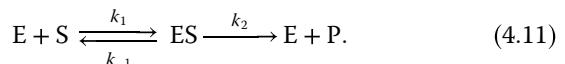
A biochemical reaction is reversible if it may proceed in both directions, leading to a positive or negative sign of the rate v . The actual direction depends on the current reactant concentrations. In theory, every reaction should be reversible. In practice, we can consider many reactions as irreversible, since (i) reactants in cellular environment cannot assume any concentration, (ii) coupling of a chemical conversion to ATP consumption leads to a severe drop in free energy and therefore makes a reaction reversal energetically unfavorable, and (iii) for compound destruction, such as protein degradation, reversal by chance is extremely unlikely.

The detailed consideration of enzyme mechanisms by applying the mass action law for the single events has led

to a number of standard kinetic descriptions, which will be explained in the following. For further information on equilibrium thermodynamics in reaction systems also see Section 15.6.

4.1.4 Michaelis–Menten Kinetics

Brown [4] proposed an enzymatic mechanism for invertase, catalyzing the cleavage of saccharose to glucose and fructose. This mechanism holds in general for all one-substrate reactions without backward reaction and without effectors, such as



It comprises a reversible formation of an enzyme–substrate complex ES from the free enzyme E and the substrate S and an irreversible release of the product P. The ODE system for the dynamics of this reaction reads

$$\frac{dS}{dt} = -k_1 \text{E} \cdot \text{S} + k_{-1} \text{ES}, \quad (4.12)$$

$$\frac{d\text{ES}}{dt} = k_1 \text{E} \cdot \text{S} - (k_{-1} + k_2) \text{ES}, \quad (4.13)$$

$$\frac{d\text{E}}{dt} = -k_1 \text{E} \cdot \text{S} + (k_{-1} + k_2) \text{ES}, \quad (4.14)$$

$$\frac{d\text{P}}{dt} = k_2 \text{ES}. \quad (4.15)$$

The reaction rate is equal to the negative decay rate of the substrate as well as to the rate of product formation:

$$v = -\frac{dS}{dt} = \frac{dP}{dt}. \quad (4.16)$$

This ODE system (Eqs. (4.12)–(4.16)) cannot be solved analytically. Figure 4.3 shows numerical solutions for different parameter sets.

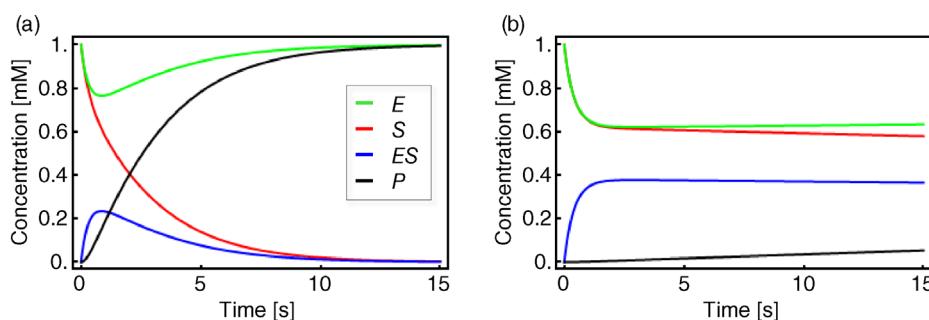


Figure 4.3 Temporal evolution of the equation system (4.12)–(4.15). Shown are S (red), E (green), ES (blue), and P (black). The initial concentrations are in both panels $S(0) = E(0) = 1\text{ mM}$ and $ES(0) = P(0) = 0\text{ mM}$. Parameter values: $k_1 = 1 \cdot \text{M}^{-1}\text{s}^{-1}$, $k_2 = 1 \cdot \text{s}^{-1}$, and either $k_3 = 1 \cdot \text{s}^{-1}$ (a) or $k_3 = 0.01 \cdot \text{s}^{-1}$ (b).

Different assumptions have been used to simplify this system in a satisfactory way. Michaelis and Menten [5] considered a *quasi-equilibrium* between the free enzyme and the enzyme–substrate complex, meaning that the reversible conversion of E and S to ES is much faster than the decomposition of ES into E and P, or in terms of the kinetic constants, that is,

$$k_1, k_{-1} \gg k_2. \quad (4.17)$$

This is the situation as shown in Figure 4.3b.

Briggs and Haldane [6] assumed that during the course of reaction a state is reached where the concentration of the ES complex remains constant, the so-called quasi-steady state. This assumption is justified only if the initial substrate concentration is much larger than the enzyme concentration, $S(t=0) \gg E$, otherwise such a state will never be reached. In mathematical terms, we obtain

$$\frac{dES}{dt} = 0. \quad (4.18)$$

In the following, we derive an expression for the reaction rate from the ODE system (4.12)–(4.15) and the quasi-steady-state assumption for ES. First, adding Eqs. (4.13) and (4.14) results in

$$\frac{dES}{dt} + \frac{dE}{dt} = 0 \quad \text{or} \quad E_{\text{total}} = E + ES = \text{constant}. \quad (4.19)$$

This expression shows that enzyme is neither produced nor consumed in this reaction; it may be free or part of the complex, but its total concentration remains constant. Introducing (4.19) into (4.13) under the steady-state assumption (4.18) yields

$$ES = \frac{k_1 E_{\text{total}} S}{k_1 S + k_{-1} + k_2} = \frac{E_{\text{total}} S}{S + ((k_{-1} + k_2)/k_1)}. \quad (4.20)$$

For the reaction rate, this gives

$$v = \frac{k_2 E_{\text{total}} S}{S + ((k_{-1} + k_2)/k_1)}. \quad (4.21)$$

In enzyme kinetics, it is convention to present Eq. (4.21) in a simpler form, which is important in theory and practice

$$v = \frac{V_{\max} S}{S + K_m}. \quad (4.22)$$

Equation (4.22) is the expression for Michaelis–Menten kinetics. The parameters have the following meaning: the *maximal velocity*,

$$V_{\max} = k_2 E_{\text{total}}, \quad (4.23)$$

is the maximal rate that can be attained, when the enzyme is completely saturated with substrate. The

Michaelis constant,

$$K_m = \frac{k_{-1} + k_2}{k_1}, \quad (4.24)$$

is equal to the substrate concentration that yields the half-maximal reaction rate. For the quasi-equilibrium assumption (Eq. (4.17)), it holds that $K_m \approx k_{-1}/k_1$. The maximum velocity divided by the enzyme concentration (here $k_2 = V_{\max}/E_{\text{total}}$) is often called the turnover number, k_{cat} . The meaning of the parameters is illustrated in the plot of rate versus substrate concentration (Figure 4.4).

4.1.4.1 How to Derive a Rate Equation

Below, we will present some enzyme kinetic standard examples. Individual mechanisms for your specific enzyme of interest may be more complicated or merely differ from these standards. Therefore, we summarize here the general way of deriving a rate equation.

- 1) Draw a wiring diagram of all steps to consider (e.g., Eq. (4.11)). It contains all substrates and products (S and P) and n free or bound enzyme species (E and ES).
- 2) The right sides of the ODEs for the concentrations changes sum up the rates of all steps leading to or away from a certain substance (e.g., Eqs. (4.12)–(4.15)). The rates follow mass action kinetics (Eq. (4.3)).
- 3) The sum of all enzyme-containing species is equal to the total enzyme concentration E_{total} (the right side of all differential equations for enzyme species sums up to zero). This constitutes one equation.
- 4) The assumption of quasi-steady state for $n - 1$ enzyme species (i.e., setting the right sides of the respective ODEs equal to zero) together with (3) result in n

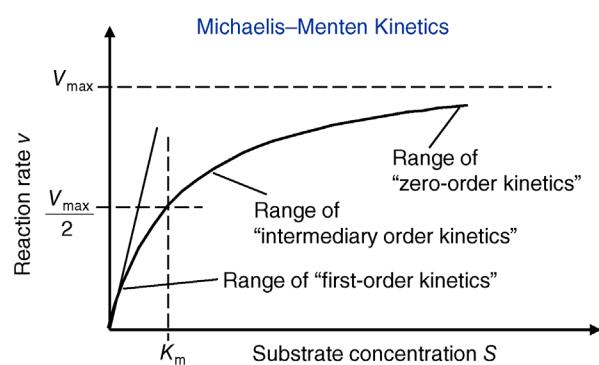


Figure 4.4 Dependence of reaction rate v on substrate concentration S in Michaelis–Menten kinetics. V_{\max} denotes the maximal reaction rate that can be reached for large substrate concentration. K_m is the substrate concentration that results in half-maximal reaction rate. For low substrate concentration, v increases almost linearly with S , while for high substrate concentrations v is almost independent of S .

algebraic equations for the concentrations of the n enzyme species.

- 5) The reaction rate is equal to the rate of product formation (e.g., Eq. (4.16)). Insert the respective concentrations of enzyme species resulting from (4).

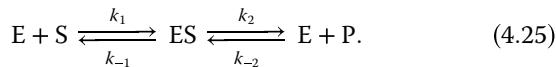
4.1.4.2 Parameter Estimation and Linearization of the Michaelis–Menten Equation

To assess the values of the parameters V_{\max} and K_m for an isolated enzyme, one measures the initial rate for different initial concentrations of the substrate. Since the rate is a nonlinear function of the substrate concentration, one has to determine the parameters by nonlinear regression. Another way is to transform Eq. (4.22) to a linear relation between variables and then apply linear regression.

The advantage of the transformed equations is that one may read the parameter value more or less directly from the graph obtained by linear regression of the measurement data. In the Lineweaver–Burk plot [7] (Table 4.2), the values for V_{\max} and K_m can be obtained from the intersections of the graph with the ordinate and the abscissa, respectively. The Lineweaver–Burk plot is also helpful to easily discriminate different types of inhibition (see below). The drawback of the transformed equations is that they may be sensitive to errors for small or high substrate concentrations or rates. Eadie and Hofstee [8] and Hanes and Woolf [9] have introduced other types of linearization to overcome this limitation.

4.1.4.3 The Michaelis–Menten Equation for Reversible Reactions

In practice, many reactions are reversible. The enzyme may catalyze the reaction in both directions. Consider the following mechanism:



The product formation is given by

$$\frac{dP}{dt} = k_2 ES - k_{-2} E \cdot P = v. \quad (4.26)$$

The respective rate equation reads

$$v = E_{\text{total}} \frac{\frac{Sq - P}{Sk_1/(k_{-1}k_2) + 1/k_{-2} + k_2/(k_{-1}k_2) + P/k_{-1}}}{\frac{(V_{\max}^{\text{for}}/K_{mS})S - (V_{\max}^{\text{back}}/K_{mP})P}{1 + S/K_{mS} + P/K_{mP}}}. \quad (4.27)$$

While the parameters $k_{\pm 1}$ and $k_{\pm 2}$ are the kinetic constants of the individual reaction steps, the phenomenological parameters V_{\max}^{for} and V_{\max}^{back} denote the maximal velocity in forward or backward direction, respectively, under zero product or substrate concentration, and the phenomenological parameters K_{mS} and K_{mP} denote the substrate or product concentration causing half-maximal forward or backward rate. They are related by the so-called Haldane relation in the following way [10]:

$$K_{\text{eq}} = \frac{V_{\max}^{\text{for}} K_{mP}}{V_{\max}^{\text{back}} K_{mS}}. \quad (4.28)$$

4.1.5 Regulation of Enzyme Activity by Effectors

Enzymes may immensely increase the rate of a reaction, but this is not their only function. Enzymes are involved in metabolic regulation in various ways. Their production and degradation is often adapted to the current requirements of the cell. Furthermore, they may be targets of effectors, both inhibitors and activators.

The effectors are small molecules, or proteins, or other compounds that influence the performance of the enzymatic reaction. The interaction of effector and enzyme changes the reaction rate. Such regulatory interactions

Table 4.2 Different approaches for the linearization of Michaelis–Menten enzyme kinetics.

	Lineweaver–Burk	Eadie–Hofstee	Hanes–Woolf
Transformed equation.	$\frac{1}{v} = \frac{K_m}{V_{\max} S} + \frac{1}{V_{\max}}$	$v = V_{\max} - K_m \frac{v}{S}$	$\frac{S}{v} = \frac{S}{V_{\max}} + \frac{K_m}{V_{\max}}$
New variables	$\frac{1}{v}, \frac{1}{S}$	$v, \frac{v}{S}$	$\frac{S}{v}, \frac{S}{S}$
Graphical representation			

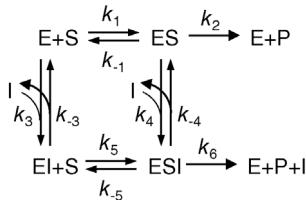


Figure 4.5 General scheme of inhibition in Michaelis–Menten kinetics. Reactions 1 and 2 belong to the standard scheme of Michaelis–Menten kinetics. Competitive inhibition is given, if in addition reaction 3 (and not reactions 4, 5, or 6) occurs. Uncompetitive inhibition involves reactions 1, 2, and 4, and noncompetitive inhibition comprises reactions 1, 2, 3, 4, and 5. Occurrence of reaction 6 indicates partial inhibition.

that are crucial for the fine-tuning of metabolism will be considered here [11].

Basic types of inhibition are distinguished by the state, in which the enzyme may bind the effector (i.e., the free enzyme E, the enzyme–substrate complex ES, or both), and by the ability of different complexes to release the product. The general pattern of inhibition is schematically represented in Figure 4.5. The different types result, if some of the interactions may not occur.

The rate equations are derived according to the following scheme:

- 1) Consider binding equilibriums between compounds and their complexes:

$$\begin{aligned} K_m &\cong \frac{k_{-1}}{k_1} = \frac{E \cdot S}{ES}, \quad K_{1,3} = \frac{k_{-3}}{k_3} = \frac{E \cdot I}{EI}, \\ K_{1,4} &= \frac{k_{-4}}{k_4} = \frac{ES \cdot I}{ESI}, \quad K_{1,5} = \frac{k_{-5}}{k_5} = \frac{EI \cdot S}{ESI}. \end{aligned} \quad (4.29)$$

Note that, if all reactions may occur, the Wegscheider condition [12] holds in the form

$$\frac{k_1 k_4}{k_{-1} k_{-4}} = \frac{k_3 k_5}{k_{-3} k_{-5}}, \quad (4.30)$$

which means that the difference in the free energies between two compounds (e.g., E and ESI) is independent of the choice of the reaction path (here via ES or via EI).

- 2) Take into account the moiety conservation for the total enzyme (include only those complexes that occur in the course of reaction):

$$E_{\text{total}} = E + ES + EI + ESI. \quad (4.31)$$

- 3) The reaction rate is equal to the rate of product formation

$$\nu = \frac{dP}{dt} = k_2 ES + k_6 ESI. \quad (4.32)$$

Equations (4.29)–(4.31) constitute four independent equations for the four unknown concentrations of E, ES, EI, and ESI. Their solution can be inserted into Eq. (4.32). The effect of the inhibitor depends on the concentrations of substrate and inhibitor and on the relative affinities to the enzyme. Table 4.3 lists the different types of inhibition for irreversible and reversible Michaelis–Menten kinetics together with the respective rate equations.

In the case of *competitive* inhibition, the inhibitor competes with the substrate for the binding site (or inhibits substrate binding by binding elsewhere to the enzyme) without being transformed itself. An example for this type is the inhibition of succinate dehydrogenase by malonate. The enzyme converts succinate to fumarate forming a double bond. Malonate has two carboxyl groups, like the proper substrates, and may bind to the enzyme, but the formation of a double bond cannot take place. Since substrates and inhibitor compete for the binding sites, a high concentration of one of them may displace the other one. For very high substrate concentrations, the same maximal velocity as without inhibitor is reached, but the effective K_m value is increased.

In the case of *uncompetitive* inhibition, the inhibitor binds only to the ES complex. The reason may be that the substrate binding caused a conformational change, which opened a new binding site. Since S and I do not compete for binding sites, an increase in the concentration of S cannot displace the inhibitor. In the presence of inhibitor, the original maximal rate cannot be reached (lower V_{\max}). For example, an inhibitor concentration of $I = K_{1,4}$ halves the K_m value as well as V_{\max} . Uncompetitive inhibition occurs rarely for one-substrate reactions, but more frequently in the case of two substrates. One example is inhibition of arylsulphatase by hydrazine.

Noncompetitive inhibition is present, if substrate binding to the enzyme does not alter the binding of the inhibitor. There must be different binding sites for substrate and inhibitor. In the classical case, the inhibitor has the same affinity to the enzyme with or without bound substrate. If the affinity changes, this is called mixed inhibition. A standard example is inhibition of chymotrypsin by H^+ -ions.

If the product may also be formed from the enzyme–substrate–inhibitor complex, the inhibition is only partial. For high rates of product release (high values of k_6), this can even result in an activating instead of an inhibiting effect.

The general types of inhibition, competitive, uncompetitive, and noncompetitive inhibition, also apply for the reversible Michaelis–Menten mechanism. The respective rate equations are also listed in Table 4.3.

Table 4.3 Types of inhibition for irreversible and reversible Michaelis–Menten kinetics^b.

Name	Implementation	Equation – irreversible	Equation – reversible case	Characteristics
Competitive inhibition	I binds only to free E; P-release only from ES-complex $k_{\pm 4} = k_{\pm 5} = k_6 = 0$	$v = \frac{V_{\max}S}{K_m \cdot i_3 + S}$	$v = \frac{V_{\max}^f(S/K_{mS}) - V_{\max}^r(P/K_{mP})}{(S/K_{mS}) + (P/K_{mP}) + i_3}$	K_m changes, V_{\max} remains same. S and I compete for the binding place; high S may out compete I.
Uncompetitive Inhibition	I binds only to the ES-complex; P-release only from ES-complex $k_{\pm 3} = k_{\pm 5} = k_6 = 0$	$v = \frac{V_{\max}S}{K_m + S \cdot i_4}$	$v = \frac{V_{\max}^f(S/K_{mS}) - V_{\max}^r(P/K_{mP})}{1 + ((S/K_{mS}) + (P/K_{mP})) \cdot i_4}$	K_m and V_{\max} change, but their ratio remains same. S may not out compete I
Noncompetitive inhibition	I binds to E and ES; P-release only from ES $K_{I,3} = K_{I,4}, k_6 = 0$	$v = \frac{V_{\max}S}{(K_m + S) \cdot i_3}$	$v = \frac{V_{\max}^f(S/K_{mS}) - V_{\max}^r(P/K_{mP})}{(1 + (S/K_{mS}) + (P/K_{mP})) \cdot i_3}$	K_m remains, V_{\max} changes. S may not out compete I
Mixed inhibition	I binds to E and ES; P-release only from ES $K_{I,3} \neq K_{I,4}, k_6 = 0$	$v = \frac{V_{\max}S}{K_m \cdot i_4 + S \cdot i_3}$		K_m and V_{\max} change. $K_{I,3} > K_{I,4}$: competitive-noncompetitive inhibition $K_{I,3} < K_{I,4}$: noncompetitive-uncompetitive inhibition K_m and V_{\max} change if $k_6 > k_2$: activation instead of inhibition.
Partial Inhibition	I may bind to E and ES; P-release from ES and ESI $K_{I,3} \neq K_{I,4}, k_6 \neq 0$	$v = \frac{V_{\max}S(1 + ((k_6I)/k_2K_{I,3}))}{K_m \cdot i_4 + S \cdot i_3}$		

^b The following abbreviations are used: $K_{I,3} = \frac{k_{-3}}{k_3}$, $K_{I,4} = \frac{k_{-4}}{k_4}$, $i_3 = 1 + \frac{I}{K_{I,3}}$, $i_4 = 1 + \frac{I}{K_{I,4}}$.

4.1.5.1 Substrate Inhibition

A common characteristic of enzymatic reaction is the increase of the reaction rate with increasing substrate concentration S up to the maximal velocity V_{\max} . But in some cases, a decrease of the rate above a certain value of S is recorded. A possible reason is the binding of a further substrate molecule to the enzyme–substrate complex yielding the complex ESS that cannot form a product. This kind of inhibition is reversible if the second substrate can be released. The rate equation can be derived using the scheme of uncompetitive inhibition by replacing the inhibitor by another substrate. It reads

$$v = k_2ES = \frac{V_{\max}S}{K_m + S(1 + (S/K_1))}. \quad (4.33)$$

This expression has an optimum, that is, a maximal value of v , at

$$S_{\text{opt}} = \sqrt{K_m K_1} \quad \text{with} \quad v_{\text{opt}} = \frac{V_{\max}}{1 + 2\sqrt{K_m/K_1}}. \quad (4.34)$$

The dependence of v on S is shown in Figure 4.6. A typical example for substrate inhibition is the binding of two succinate molecules to malonate dehydrogenase, which possesses two binding pockets for the carboxyl group. This is schematically represented in Figure 4.6.

4.1.5.2 Binding of Ligands to Proteins

Every molecule that binds to a protein is a ligand, irrespective of whether it is subject of a reaction or not. Below we consider binding to monomer and oligomer proteins. In oligomers, there may be interactions between the binding sites on the subunits.

Consider binding of one ligand (S) to a protein (E) with only one binding site:



The binding constant K_B is given by

$$K_B = \left(\frac{ES}{E \cdot S} \right)_{\text{eq}}. \quad (4.36)$$

The reciprocal of K_B is the dissociation constant K_D . The fractional saturation Y of the protein is determined by the number of subunits that have bound ligands, divided by the total number of subunits. The fractional saturation for one subunit is

$$Y = \frac{ES}{E_{\text{total}}} = \frac{ES}{ES + E} = \frac{K_B \cdot S}{K_B \cdot S + 1}. \quad (4.37)$$

The plot of Y versus S at constant total enzyme concentration is a hyperbola, like the plot of v versus S in the Michaelis–Menten kinetics (Eq. (4.22)). At a process where the binding of S to E is the first step followed by

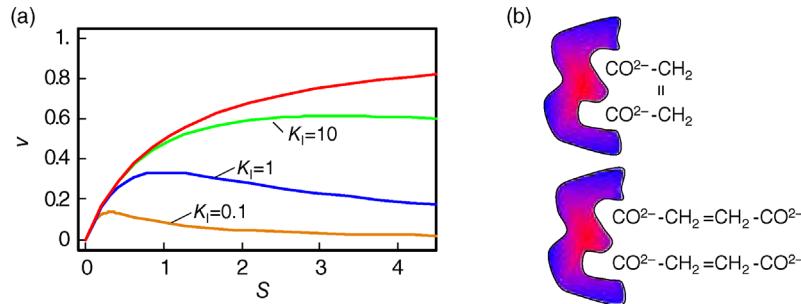


Figure 4.6 Substrate inhibition. (a) Plot of reaction rate v against substrate concentration S for an enzyme with substrate inhibition. The upper curve shows Michaelis-Menten kinetics without inhibition, the lower curves show kinetics for the indicated values of binding constant K_i . Parameter values: $V_{\max} = 1$, $K_m = 1$. (b) Visualization of a possible mechanism for substrate inhibition: The enzyme (gray item) has two binding pockets to bind different parts of a substrate molecule (upper scheme). In case of high substrate concentration, two different molecules may enter the binding pockets, thereby preventing the specific reaction (lower scheme).

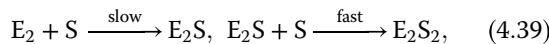
product release and where the initial concentration of S is much higher as the initial concentration of E , the rate is proportional to the concentration of ES and it holds

$$\frac{v}{V_{\max}} = \frac{ES}{E_{\text{total}}} = Y. \quad (4.38)$$

If the protein has several binding sites, then interactions may occur between these sites, that is, the affinity to further ligands may change after binding of one or more ligands. This phenomenon is called *cooperativity*. Positive or negative cooperativity denote increase or decrease in the affinity of the protein to a further ligand, respectively. Homotropic or heterotropic cooperativity denotes that the binding to a certain ligand influences the affinity of the protein to a further ligand of the same or another type, respectively.

4.1.5.3 Positive Homotropic Cooperativity and the Hill Equation

Consider a dimeric protein with two identical binding sites. The binding to the first ligand facilitates the binding to the second ligand.



where E is the monomer and E_2 is the dimer. The fractional saturation is given by

$$Y = \frac{E_2S + 2 \cdot E_2S_2}{2 \cdot E_{2,\text{total}}} = \frac{E_2S + 2 \cdot E_2S_2}{2 \cdot E_2 + 2 \cdot E_2S + 2 \cdot E_2S_2}. \quad (4.40)$$

If the affinity to the second ligand is strongly increased by binding to the first ligand, then E_2S will react with S as soon as it is formed and the concentration of E_2S can be neglected. In the case of complete *cooperativity*, that is,

every protein is either empty or fully bound, Eq. (4.39) reduces to



The binding constant reads

$$K_B = \frac{E_2S_2}{E_2 \cdot S^2}, \quad (4.42)$$

and the fractional saturation is

$$Y = \frac{2 \cdot E_2S_2}{2 \cdot E_{2,\text{total}}} = \frac{E_2S_2}{E_2 + E_2S_2} = \frac{K_B \cdot S^2}{1 + K_B \cdot S^2}. \quad (4.43)$$

Generally, for a protein with n subunits it holds:

$$v = V_{\max} \cdot Y = \frac{V_{\max} \cdot K_B \cdot S^n}{1 + K_B \cdot S^n}. \quad (4.44)$$

This is the general form of the *Hill equation*. To derive it, we assumed complete homotropic cooperativity. The plot of the fractional saturation Y versus substrate concentration S is a sigmoid curve with the inflection point at $1/K_B$. The quantity n (often “ h ” is used instead) is termed the *Hill coefficient*.

The derivation of this expression was based on experimental findings concerning the binding of oxygen to hemoglobin (Hb) [13,14]. In 1904, Bohr *et al.* found that the plot of the fractional saturation of Hb with oxygen against the oxygen partial pressure had a sigmoid shape. Hill (1909) explained this with interactions between the binding sites located at the hem subunits. At this time, it was already known that every subunit hem binds one molecule of oxygen. Hill assumed complete cooperativity and predicted an experimental Hill coefficient of 2.8. Today it is known that hemoglobin has four binding sites, but that the cooperativity is not complete. The sigmoid binding characteristic has the advantage that Hb binds

strongly to oxygen in the lung with a high oxygen partial pressure while it can release O_2 easily in the body with low oxygen partial pressure.

4.1.5.4 The Monod–Wyman–Changeux Model for Sigmoid Kinetics

The Monod model [15] explains sigmoid enzyme kinetics by taking into account the interaction of subunits of an enzyme. We will show here the main characteristics and assumptions of this kinetics. The full derivation is given in the web material. It uses the following assumptions: (i) the enzyme consists of n identical subunits, (ii) each subunit can assume an active (R) or an inactive (T) conformation, (iii) all subunits change their conformations at the same time (concerted change), and (iv) the equilibrium between the R and the T conformation is given by an allosteric constant

$$L = \frac{T_0}{R_0}. \quad (4.45)$$

The binding constants for the active and inactive conformations are given by K_R and K_T , respectively. If substrate molecules can only bind to the active form, that is, if $K_T = 0$, the rate can be expressed as

$$V = \frac{V_{\max} K_R S}{(1 + K_R S)} \frac{1}{[1 + \{L/(1 + K_R S)^n\}]}, \quad (4.46)$$

where the first factor $(V_{\max} K_R S)/(1 + K_R S)$ corresponds to the Michaelis–Menten rate expression, while the second factor $[1 + \{L/(1 + K_R S)^n\}]^{-1}$ is a regulatory factor.

For $L = 0$, the plot v versus S is hyperbola as in Michaelis–Menten kinetics. For $L > 0$, we obtain a sigmoid curve shifted to the right. A typical value for the allosteric constant is $L \cong 10^4$ (Figure 4.7).

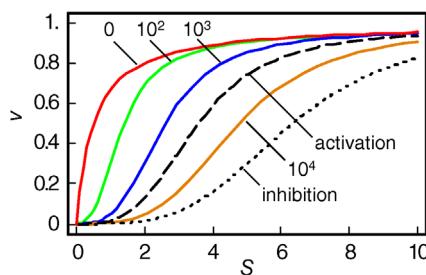


Figure 4.7 Model of Monod, Wyman, and Changeux: Dependence of the reaction rate on substrate concentration for different values of the allosteric constant L , according to Eq. (4.46). Parameters: $V_{\max} = 1$, $n = 4$, $K_R = 2$, $K_T = 0$. The value of L is indicated at the curves. Obviously, increasing value of L causes stronger sigmoidity. The influence of activators or inhibitors (compare Eq. (4.47)) is illustrated with the dotted line for $K_I = 2$ and with the dashed line for $K_{AA} = 2$ ($L = 10^4$ in both cases).

Up to now we considered in the model of Monod, Wyman, and Changeux only homotropic and positive effects. But this model is also well suited to explain the dependence of the reaction rate on activators and inhibitors. Activators A bind only to the active conformation and inhibitors I bind only to the inactive conformation. This shifts the equilibrium to the respective conformation. Effectively, the binding to effectors changes L :

$$L' = L \frac{(1 + K_I)^n}{(1 + K_{AA})^n}, \quad (4.47)$$

where K_I and K_{AA} denote binding constants. The interaction with effectors is a heterotropic effect. An activator weakens the sigmoidity, while an inhibitor strengthens it.

A typical example for an enzyme with sigmoid kinetics that can be described with the Monod model is the enzyme phosphofructokinase, which catalyzes the transformation of fructose-6-phosphate and ATP to fructose-1,6-bisphosphate. AMP, NH_4 , and K^+ are activators, ATP is an inhibitor.

4.1.6

Generalized Mass Action Kinetics

Mass action kinetics (see Section 4.1.1) has experienced refinements in different ways. The fact that experimental results frequently do not show the linear dependence of rate on concentrations as assumed in mass action laws is acknowledged in power law kinetics used in the S-systems approach. Here, the rate reads

$$\frac{v_j}{v_j^0} = k_j \prod_{i=1}^n \left(\frac{S_i}{S_i^0} \right)^{g_{j,i}}, \quad (4.48)$$

where the concentrations S_i and rates v_j are normalized to some standard value denoted by superscript 0, and $g_{j,i}$ is a real number instead of an integer as in Eq. (4.4). The normalization yields dimensionless quantities. The power law kinetics can be considered as a generalization of the mass action rate law. The exponent $g_{j,i}$ is equal to the concentration elasticities, that is, the scaled derivatives of rates with respect to substrate concentrations (see Section 4.3, Eq. (4.107)). Substrates and effectors (their concentrations both denoted by S_i) enter expression (4.48) in the same formal way, but the respective exponents $g_{j,i}$ will be different. The exponents $g_{j,i}$ will be positive for substrates and activators, but should assume a negative value for inhibitors.

4.1.7

Approximate Kinetic Formats

In metabolic modeling studies, approximate kinetic formats are used (for a recent review see Ref. [16]). They

preassume that each reaction rate v_j is proportional to the enzyme concentration E_j . The rates, enzyme concentrations, and substrate concentrations are normalized with respect to a references state, which is usually a steady state. This leads to the general expression

$$\frac{v_j}{v_j^0} = \frac{E_j}{E_j^0} \cdot f\left(\frac{\mathbf{S}}{\mathbf{S}^0}, \boldsymbol{\epsilon}_c^0\right), \quad (4.49)$$

where $\boldsymbol{\epsilon}_c$ is the matrix of concentration elasticities as explained in Section 4.3. One example is the so-called lin-log kinetics

$$\frac{\mathbf{v}}{\mathbf{v}^0} = \frac{\mathbf{E}}{\mathbf{E}^0} \left(\mathbf{I} + \boldsymbol{\epsilon}_c^0 \cdot \ln \frac{\mathbf{S}}{\mathbf{S}^0} \right), \quad (4.50)$$

where \mathbf{I} is the $r \times r$ identity matrix. Another example is an approximation of the power-law kinetics

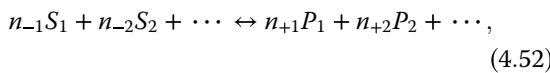
$$\ln \frac{\mathbf{v}}{\mathbf{v}^0} = \ln \frac{\mathbf{E}}{\mathbf{E}^0} + \boldsymbol{\epsilon}_c^0 \ln \frac{\mathbf{S}}{\mathbf{S}^0}. \quad (4.51)$$

Approximative kinetics simplify the determination of model parameters and, especially, of concentration elasticities, since Eq. (4.51) as set of linear equations in the elasticity coefficients.

4.1.8

Convenience Kinetics and Modular Rate Laws

The convenience kinetics [17] has been introduced to ease parameter estimation and to have a kinetic mechanism, where all parameters are independent on each other and not related via the Haldane relation (Eq. (4.28)). It is a generalized form of Michaelis–Menten kinetics that covers all possible stoichiometries, and describes enzyme regulation by activators and inhibitors. For a reaction with stoichiometry



it reads

$$\mathbf{v} = E_{\text{total}} \cdot f_{\text{reg}} \cdot \frac{k_{\text{cat}}^{\text{for}} \prod_i (S_i/K_{m,S_i})^{n_{-i}} - k_{\text{cat}}^{\text{back}} \prod_j (P_j/K_{m,P_j})^{n_{+j}}}{\prod_i (1 + (S_i/K_{m,S_i}) + \cdots + (S_i/K_{m,S_i})^{n_{-i}}) + \prod_j (1 + (P_j/K_{m,P_j}) + \cdots + (P_j/K_{m,P_j})^{n_{+j}}) - 1}, \quad (4.53)$$

with enzyme concentration E_{total} and turnover rates $k_{\text{cat}}^{\text{for}}$ and $k_{\text{cat}}^{\text{back}}$. The regulatory prefactor f_{reg} is either 1 (in case of no regulation) or a product of terms $M/(K_A + M)$ or $1 + M/K_A$ for activators and $K_I/(K_I + M)$ for inhibitors. Activation constants K_A and inhibition constants K_I are measured in concentration units. M is the concentration of the modifier.

In analogy to Michaelis–Menten kinetics, K_m values denote substrate concentrations, at which the reaction rate is half-maximal if the reaction products are absent; K_I and K_A values denote concentrations, at which the inhibitor or activator has its half-maximal effect. In this respect, many parameters in convenience kinetics are comparable to the kinetic constants measured in enzyme assays. This is important for parameter estimation (see Section 4.2).

To facilitate thermodynamic independence of the parameters, we introduce new system parameters that can be varied independently, without violating any thermodynamic constraints (see Section 4.1.1). For each reaction, we define the velocity constant $K_V = (k_{\text{cat}}^{\text{for}} \cdot k_{\text{cat}}^{\text{back}})^{1/2}$ (geometric mean of the turnover rates in both directions). Given the equilibrium and velocity constants, the turnover rates can be written as $k_{\text{cat}}^{\text{for}} = K_V (K_{\text{eq}})^{-1/2}$, $k_{\text{cat}}^{\text{back}} = K_V (K_{\text{eq}})^{1/2}$. The equilibrium constants K_{eq} can be expressed by independent parameters such as the Gibbs free energies of formation: for each substance i , we define the dimensionless energy constant $K_i^G = \exp(G_i(0)/(RT))$ with Boltzmann's gas constant $R = 8.314 \text{ J mol}^{-1} \text{ K}^{-1}$ and absolute temperature T . The equilibrium constants then satisfy $\ln K_{\text{eq}} = -N^T \ln K^G$.

In more general terms, modular rate laws are a family of reversible rate laws for reactions with arbitrary stoichiometries and various types of regulation, including mass-action, Michaelis–Menten, and uni–uni reversible Hill kinetics as special cases 20 385 728. Their general form reads

$$v = E_{\text{total}} \cdot f_{\text{reg}} \cdot \frac{T}{D + D_{\text{reg}}}, \quad (4.54)$$

where f_{reg} describes complete or partial regulation (e.g., by an inhibitor), T is the numerator (equivalently to the one as used in equation (4.53)), while the components of the denominator, D and D_{reg} , depend on reaction stoichiometry, selected rate law, allosteric regulation, and on the preferred model parameterization. Five versions of denominator have been introduced:

-
- 1) Power-law modular rate law: $D = 1$ (such as mass action kinetics)
 - 2) Common modular rate law: as in Eq. (4.53)
 - 3) Simultaneous binding modular rate law:

$$D = \prod_i \left(1 + \frac{S_i}{K_{m,S_i}}\right)^{n_{-i}} \prod_j \left(1 + \frac{P_j}{K_{m,P_j}}\right)^{n_{+j}}$$

4) Direct binding modular rate law:

$$D = 1 + \prod_i \left(\frac{S_i}{K_{m,S_i}} \right)^{n-i} + \prod_j \left(\frac{P_j}{K_{m,P_j}} \right)^{n+j}$$

5) Force-dependent modular rate law:

$$D = \sqrt{\prod_i \left(\frac{S_i}{K_{m,S_i}} \right)^{n-i} \prod_j \left(\frac{P_j}{K_{m,P_j}} \right)^{n+j}}$$

With a thermodynamically safe parameterization of these rate laws, parameter sets obtained by model fitting, sampling, or optimization are guaranteed to lead to consistent chemical equilibrium states, as demonstrated above for convenience kinetics.

4.2 Metabolic Control Analysis

Summary

Metabolic control analysis (MCA) is a powerful quantitative and qualitative framework for studying the relationship between steady-state properties of a network of biochemical reaction and the properties of the individual reactions. It investigates the sensitivity of steady-state properties of the network to small parameter changes. MCA is a useful tool for theoretical and experimental analysis of control and regulation in cellular systems.

MCA was independently founded by two different groups in the 1970s [18,19] and was further developed by many different groups upon the application to different metabolic systems. A milestone in its formalization was provided in Ref. [20]. Originally intended for metabolic networks, MCA has nowadays found applications also for signaling pathways, gene expression models, and hierarchical networks [21–25].

Metabolic networks are very complex systems that are highly regulated and exhibit a lot of interactions like feedback inhibition or common substrates such as ATP for different reactions. Many mechanisms and regulatory properties of isolated enzymatic reactions are known. The development of MCA was motivated by a series of questions like the following: Can one predict properties or behavior of metabolic networks from the knowledge about their parts, the isolated reactions? Which individual steps control a flux or a steady-state concentration? Is there a rate-limiting step? Which effectors or modifications have the most prominent effect on the reaction rate? In biotechnological production processes, it is of interest which enzyme(s) should be activated in order to increase the rate of synthesis of a desired metabolite. There are also related problems in health care. Concerning metabolic disorders involving overproduction of a metabolite, which reactions should be modified in order to downregulate this metabolite while perturbing the rest of the metabolism as weakly as possible?

In metabolic networks, the steady-state variables, that is, the fluxes and the metabolite concentrations, depend on the value of parameters such as enzyme concentrations, kinetic constants (like Michaelis constants and maximal activities), and other model specific parameters. The effect of perturbations, moreover, depends on the place of the perturbation. As an illustration, in Example 4.2, we discuss a linear metabolic pathway whose enzymes are successively inhibited. We see in Figure 4.8 that an inhibition of the first enzyme has a different temporal effect than inhibition of the later enzymes. Also the steady states (here the values reached at time point 15) are different if different enzymes are hit.

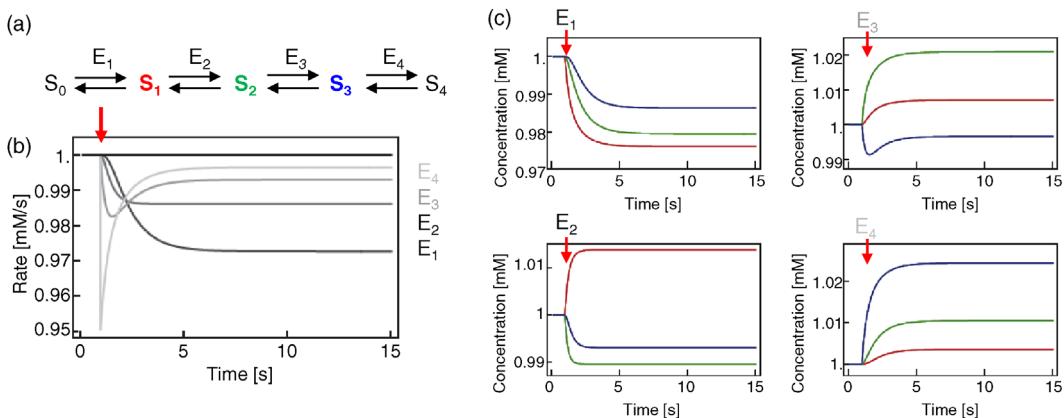


Figure 4.8 The effect of inhibiting an enzyme in an unbranched metabolic pathway depends on the position of that enzyme in the pathway. (a) Scheme of a linear metabolic pathway. Parameter values: see text. (b) Before perturbation, the system is at steady state. At time $t = 1$, one of the enzymes as indicated by gray scale is mildly inhibited by reducing its value by 5% (from 1 to 0.95). The rate of reaction 4 is presented. (c) Dynamics of metabolite concentrations upon different perturbations. Colors of time courses correspond to the colors of metabolites in (a).

Example 4.2

The effect of enzyme inhibition depends on the position of the inhibited enzyme in the network. Figure 4.8 presents a linear metabolic pathway. The reaction rates for all reactions are linear reversible mass action laws, $v_i = E_i(k_i S_{i-1} - k_{-i} S_i)$, $i = 1, \dots, 4$. For simplicity, we have chosen equal parameter values for all reactions, that is, $S_0 = S_4 = \text{constant}$, $k_i = 2$, $k_{-i} = 1$, $i = 1, \dots, 4$, and $E_i = 1$, $i = 1, \dots, 4$. Accordingly, all metabolites have a concentration of 1 in the steady state before perturbation. At time point 1 (red arrow), we perturb in each case one of the reactions by decreasing the concentration of the respective enzyme by 5%. Depending on which reaction is inhibited, the effect occurs faster or slower. In steady state, the effect of inhibition is successively decreasing from inhibiting E_1 down to E_4 (i.e., the deviation from the original steady state). Substrate concentrations also change upon enzyme inhibition. The effect on metabolite dynamics and steady-state levels is dependent on the choice of which enzyme is inhibited. If E_1 is inhibited, all metabolite concentrations decline, while they increase if E_4 is inhibited. In between it holds that inhibition of producing reactions has a diminishing effect on metabolites, while inhibition of consuming reactions leads to an accumulation. The different effects of (quantitatively comparable) perturbations of different enzymes on the steady-state flux and on the metabolite concentrations have been interpreted as control that those enzymes exert over the respective system variables.

The relations between steady-state variables and kinetic parameters are usually nonlinear. Up to now, there is no general theory that predicts the effect of large parameter changes in a network. The approach presented in the following is, basically, restricted to small parameter changes. Mathematically, the system is linearized at steady state, which yields exact results, if the parameter changes are infinitesimally small.

In this section, we will first define a set of mathematical expressions that are useful to quantify control in biochemical reaction networks. Later we will show the relations between these functions and their application for prediction of reaction network behavior.

4.2.1

The Coefficients of Control Analysis

Biochemical reaction systems are networks of metabolites connected by chemical reactions. Their behavior is determined by the properties of their components – the individual reactions and their kinetics – as well as by the network structure – the involvement of compounds in

different reaction or in brief: the stoichiometry. Hence, the effect of a perturbation exerted on a reaction in this network will depend on both – the local properties of this reaction and the embedding of this reaction in the global network.

Let $y(x)$ denote a quantity that depends on another quantity x . The effect of the change Δx on y is expressed in terms of sensitivity coefficients:

$$c_x^y = \left(\frac{x}{y} \frac{\partial y}{\partial x} \right)_{\Delta x \rightarrow 0}.$$

In practical applications, Δx might be, for example, identified with 1% change of x and Δy with the percentage change of y . The factor x/y is a normalization factor that makes the coefficient independent of units and of the magnitude of x and y . In the limiting case $\Delta x \rightarrow 0$, the sensitivity coefficients can be written as

$$c_x^y = \frac{x}{y} \frac{\partial y}{\partial x} = \frac{\partial \ln y}{\partial \ln x}. \quad (4.55)$$

Both right-hand expressions are mathematically equivalent.

Two distinct types of coefficients, local and global coefficients, reflect the relations among local and global effects of changes. *Elasticity coefficients* are local coefficients pertaining to individual reactions. They can be calculated in any given state. *Control coefficients* and *response coefficients* are global quantities. They refer to a given steady state of the entire system. After a perturbation of x , the relaxation of y to new steady state is considered.

The general form of the coefficients in control analysis as defined in Eq. (4.55) contains the normalization x/y . The normalization has the advantage that we get rid of units and can compare, for example, fluxes belonging to different branches of a network. The drawback of the normalization is that x/y is not defined as soon as $y = 0$, which may happen for certain parameter combinations. In those cases, it is favorable to work with nonnormalized coefficients. Throughout this chapter, we will consider usually normalized quantities. If we use nonnormalized coefficients, they are flagged as \tilde{c} with $\tilde{c} = \partial y / \partial x$. In general, the use of one or the other type of coefficient is also a matter of personal choice of the modeler.

Changes reflected by the different coefficients are illustrated in Figure 4.9.

4.2.1.1 The Elasticity Coefficients

An elasticity coefficient quantifies the sensitivity of a reaction rate to the change of a concentration or a parameter while all other arguments of the kinetic law are kept fixed. It measures the direct effect on the reaction velocity, while the rest of the network is not taken into

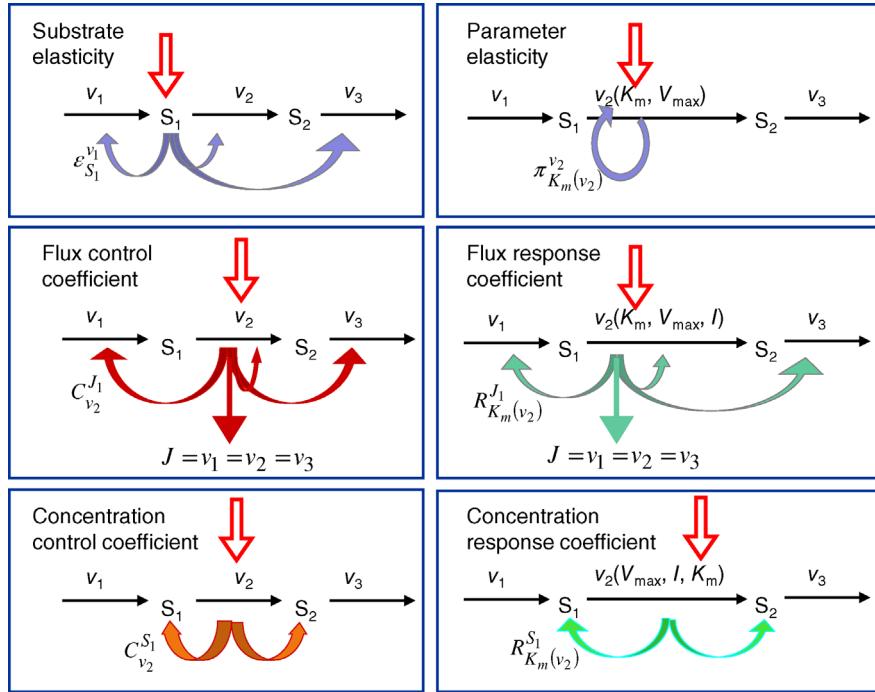


Figure 4.9 Schematic representation of perturbation and effects quantified by different coefficients of metabolic control analysis.

consideration. The sensitivity of the rate v_k of a reaction to the change of the concentration S_i of a metabolite is calculated by the ϵ -elasticity:

$$\epsilon_i^k = \frac{S_i}{v_k} \frac{\partial v_k}{\partial S_i}. \quad (4.56)$$

The nonnormalized elasticity is $\tilde{\epsilon}_i^k = \partial v_k / \partial S_i$. The π -elasticity is defined with respect to parameters p_m like kinetic constants, concentrations of enzymes, or concentrations of external metabolites as follows:

$$\pi_m^k = \frac{p_m}{v_k} \frac{\partial v_k}{\partial p_m}. \quad (4.57)$$

4.2.1.2 Control Coefficients

When defining control coefficients, we refer to a stable steady state of the metabolic system characterized by steady-state concentrations $S^{ss} = S^{ss}(p)$ and steady-state fluxes $J = v(S^{ss}(p), p)$. Any sufficiently small perturbation of an individual reaction rate, $v_k \rightarrow v_k + \Delta v_k$, by a parameter change $p_k \rightarrow p_k + \Delta p_k$ drives the system to a new steady state in close proximity with $J \rightarrow J + \Delta J$ and $S^{ss} \rightarrow S^{ss} + \Delta S$. A measure for the change of fluxes and concentrations are the control coefficients.

The *flux control coefficient* for the control of rate v_k over flux J_j is defined as

$$C_k^j = \frac{v_k}{J_j} \frac{\partial J_j / \partial p_k}{\partial v_k / \partial p_k}. \quad (4.63)$$

Example 4.3

In Michaelis–Menten kinetics, the rate v of a reaction depends on the substrate concentration S in the form $v = V_{max}S / (K_m + S)$ (Eq. (4.22)). The sensitivity is given by the elasticity $\epsilon_S^v = \partial \ln v / \partial \ln S$. Since the Michaelis–Menten equation defines a mathematical dependency of v on S , it is easy to calculate that

$$\epsilon_S^v = \frac{S}{v} \frac{\partial}{\partial S} \left(\frac{V_{max}S}{K_m + S} \right) = \frac{S}{v} \frac{V_{max}(K_m + S) - V_{max}S}{(K_m + S)^2} = \frac{S}{K_m + S}. \quad (4.58)$$

The normalized ϵ -elasticity in the case of mass action kinetics can be calculated similarly and is always 1. Whenever the rate does not depend directly on a concentration (e.g., for a metabolite of a reaction system that is not involved in the considered reaction), the elasticity is zero.

The control coefficients quantify the control that a certain reaction v_k exerts on the steady-state flux J . It should be noted that the rate change, Δv_k , is caused by the change of a parameter p_k that has a direct effect solely on v_k . Thus, it holds

$$\frac{\partial v_k}{\partial p_k} \neq 0 \quad \text{and} \quad \frac{\partial v_l}{\partial p_k} = 0 \quad (l \neq k). \quad (4.64)$$

Example 4.4

Typical values of elasticity coefficients will be explained for an isolated reaction transforming substrate S into product P. The reaction is catalyzed by enzyme E with the inhibitor I, and the activator A as depicted below



Usually, the elasticity coefficients for metabolite concentrations are in the following range:

$$\varepsilon_S^v = \frac{S \partial v}{v \partial S} > 0 \quad \text{and} \quad \varepsilon_P^v = \frac{P \partial v}{v \partial P} \leq 0. \quad (4.60)$$

In most cases, the rate increases with the concentration of the substrate (compare, e.g., Eq. (4.58)) and decreases with the concentration of the product. An exception from $\varepsilon_S^v > 0$ occurs in the case of substrate inhibition (Eq. (4.33)), where the elasticity will become negative for $S > S_{\text{opt}}$. The relation $\varepsilon_P^v = 0$ holds, if the reaction is irreversible or if the product concentration is kept zero by external mechanisms. The elasticity coefficients with respect to effectors I or A should obey

$$\varepsilon_A^v = \frac{A \partial v}{v \partial A} > 0 \quad \text{and} \quad \varepsilon_I^v = \frac{I \partial v}{v \partial I} < 0, \quad (4.61)$$

since this is essentially what the notions activator and inhibitor mean.

For the most kinetic laws, the reaction rate v is proportional to the enzyme concentration E . For example, E is a multiplicative factor in the mass action rate law as well as in the maximal rate of the Michelis-Menten rate law. Therefore, it holds that

$$\varepsilon_E^v = \frac{\partial \ln v}{\partial \ln E} = 1. \quad (4.62)$$

More complicated interactions between enzymes and substrates like metabolic channeling (direct transfer of the metabolite from one enzyme to the next without release to the medium) may lead to exceptions from this rule.

Such a parameter might be the enzyme concentration, a kinetic constant, or the concentration of a specific inhibitor or effector.

In a more compact form the flux control coefficient reads

$$C_k^j = \frac{v_k \partial J_j}{J_j \partial v_k}. \quad (4.65)$$

The respective nonnormalized flux control coefficient is $\tilde{C}_k^j = \partial J_j / \partial v_k$. Equivalently, the *concentration control*

coefficient of concentrations S_i^{ss} with respect to v_k reads

$$C_k^i = \frac{v_k \partial S_i^{\text{ss}}}{S_i^{\text{ss}} \partial v_k}. \quad (4.66)$$

4.2.1.3 Response Coefficients

The steady state is determined by the values of the parameters. A third type of coefficients expresses the direct dependence of steady-state variables on parameters. The response coefficients are defined as

$$R_m^j = \frac{p_m}{J_j} \frac{\partial J_j}{\partial p_m} \quad \text{and} \quad R_m^i = \frac{p_m}{S_i^{\text{ss}}} \frac{\partial S_i^{\text{ss}}}{\partial p_m}, \quad (4.67)$$

where the first coefficient expresses the response of the flux to a parameter perturbation, while the latter describes the response of a steady-state concentration.

4.2.1.4 Matrix Representation of the Coefficients

Control, response, and elasticity coefficients are defined with respect to all rates, steady-state concentrations, fluxes, or parameters in the metabolic system and in the respective model. They can be arranged in matrices:

$$\begin{aligned} \mathbf{C}^J &= \{C_k^j\}, \mathbf{C}^S = \{C_k^i\}, \mathbf{R}^J = \{R_m^j\}, \mathbf{R}^S = \{R_m^i\}, \\ \boldsymbol{\varepsilon} &= \{\varepsilon_i^k\}, \boldsymbol{\pi} = \{\pi_m^k\}. \end{aligned} \quad (4.68)$$

Matrix representation can also be chosen for all types of nonnormalized coefficients. The arrangement in matrices allows to applying matrix algebra in control analysis. In particular, the matrices of normalized control coefficients can be calculated from the matrices of nonnormalized control coefficient as follows:

$$\begin{aligned} \mathbf{C}^J &= (\text{dg}\mathbf{J})^{-1} \cdot \tilde{\mathbf{C}}^J \cdot \text{dg}\mathbf{J} & \mathbf{C}^S &= (\text{dg}\mathbf{S}^{\text{ss}})^{-1} \cdot \tilde{\mathbf{C}}^J \cdot \text{dg}\mathbf{J} \\ \mathbf{R}^J &= (\text{dg}\mathbf{J})^{-1} \cdot \tilde{\mathbf{R}}^J \cdot \text{dg}\mathbf{p} & \mathbf{R}^S &= (\text{dg}\mathbf{S}^{\text{ss}})^{-1} \cdot \tilde{\mathbf{R}}^S \cdot \text{dg}\mathbf{p} \\ \boldsymbol{\varepsilon} &= (\text{dg}\mathbf{v})^{-1} \cdot \tilde{\boldsymbol{\varepsilon}} \cdot \text{dg}\mathbf{S}^{\text{ss}} & \boldsymbol{\pi} &= (\text{dg}\mathbf{v})^{-1} \cdot \tilde{\boldsymbol{\pi}} \cdot \text{dg}\mathbf{p} \end{aligned} \quad (4.69)$$

The symbol “dg” stands for the diagonal matrix, that is, for a system with three reactions it holds

$$\text{dg}\mathbf{J} = \begin{pmatrix} J_1 & 0 & 0 \\ 0 & J_2 & 0 \\ 0 & 0 & J_3 \end{pmatrix}.$$

4.2.2

The Theorems of Metabolic Control Theory

Let us assume that we are interested in calculating the control coefficients for a system under investigation. Usually, the steady-state fluxes or concentrations cannot be expressed explicitly as function of the reaction rates. Therefore, flux and concentration control coefficients

cannot simply be determined by taking the respective derivatives, as we did for the elasticity coefficients in Example 4.3.

Fortunately, the work with control coefficients is eased by a set of theorems. The first type of theorems, the *summation theorems*, makes a statement about the total control over a flux or a steady-state concentration. The second type of theorems, the *connectivity theorems*, relates the control coefficients to the elasticity coefficients. Both types of theorems together with network information encoded in the stoichiometric matrix contain enough information to calculate all control coefficients.

Here, we will first introduce the theorems. Then, we will present a hypothetical perturbation experiment (as introduced by Kacser & Burns) to illustrate the summation theorem. Finally, the theorems will be derived mathematically.

4.2.2.1 The Summation Theorems

The summation theorems make a statement about the total control over a certain steady-state flux or concentration. The flux control coefficients and concentration control coefficients fulfill, respectively,

$$\sum_{k=1}^r C_{v_k}^{J_j} = 1 \quad \text{and} \quad \sum_{k=1}^r C_{v_k}^{S_i} = 0, \quad (4.70)$$

for any flux J_i and any steady-state concentration S_i^{st} . The quantity r is the number of reactions. The flux control coefficients of a metabolic network for one steady-state flux sum up to one. This means that all enzymatic reactions can share the control over this flux. The control coefficients of a metabolic network for one steady-state concentration are balanced. This means again that the enzymatic reactions can share the control over this concentration, but some of them exert a negative control while others exert a positive control. Both relations can also be expressed in matrix formulation. We get

$$\mathbf{C}^J \cdot \mathbf{1} = \mathbf{1} \quad \text{and} \quad \mathbf{C}^S \cdot \mathbf{0} = \mathbf{0}. \quad (4.71)$$

The symbols $\mathbf{1}$ and $\mathbf{0}$ denote column vectors with r rows containing as entries only ones or zeros, respectively. The summation theorems for the nonnormalized control coefficients read

$$\tilde{\mathbf{C}}^J \cdot \mathbf{K} = \mathbf{K} \quad \text{and} \quad \tilde{\mathbf{C}}^S \cdot \mathbf{K} = \mathbf{0}, \quad (4.72)$$

where \mathbf{K} is the matrix satisfying $\mathbf{N} \cdot \mathbf{K} = \mathbf{0}$ (see Section 4.2). A more intuitive derivation of the summation theorems is given in the following example according to Kacser and Burns [18].

Example 4.5

The summation theorem for flux control coefficients can be derived using a thought experiment.

Consider the following unbranched pathway with fixed concentrations of the external metabolites, S_0 and S_3 :



What happens to steady-state fluxes and metabolite concentrations, if we perform an experimental manipulation of all three reactions leading to the same fractional change α of all three rates?

$$\frac{\delta v_1}{v_1} = \frac{\delta v_2}{v_2} = \frac{\delta v_3}{v_3} = \alpha. \quad (4.74)$$

The flux must increase to the same extent, $\delta J/J = \alpha$, but, since rates of producing and degrading reactions increase to the same amount, the concentrations of the metabolites remain constant $\delta S_1/S_1 = \delta S_2/S_2 = 0$.

The combined effect of all changes in local rates on the system variables $S_1^{\text{ss}}, S_2^{\text{ss}}$, and J can be written as the sum of all individual effects caused by the local rate changes. For the flux holds

$$\frac{\delta J}{J} = C_1^J \frac{\delta v_1}{v_1} + C_2^J \frac{\delta v_2}{v_2} + C_3^J \frac{\delta v_3}{v_3}. \quad (4.75)$$

It follows

$$\alpha = \alpha(C_1^J + C_2^J + C_3^J) \quad \text{or} \quad 1 = C_1^J + C_2^J + C_3^J. \quad (4.76)$$

This is just a special case of Eq. (4.70). In the same way, for the change of concentration S_1^{ss} , we obtain

$$\frac{\delta S_1^{\text{ss}}}{S_1^{\text{ss}}} = C_1^{S_1} \frac{\delta v_1}{v_1} + C_2^{S_1} \frac{\delta v_2}{v_2} + C_3^{S_1} \frac{\delta v_3}{v_3}. \quad (4.77)$$

Finally, we get

$$0 = C_1^{S_1} + C_2^{S_1} + C_3^{S_1} \quad \text{as well as} \\ 0 = C_1^{S_2} + C_2^{S_2} + C_3^{S_2}. \quad (4.78)$$

Although shown here only for a special case, these properties hold in general for systems without conservation relations. The general derivation is given in Section 4.2.3.

4.2.2.2 The Connectivity Theorems

Flux control coefficients and elasticity coefficients are related by the expression

$$\sum_{k=1}^r C_{v_k}^{J_j} \epsilon_{S_i}^{v_k} = 0. \quad (4.79)$$

Note that the sum runs over all rates v_k for any flux J_j . Considering the concentration S_i of a specific metabolite

and a certain flux J_j , each term contains the elasticity $\varepsilon_{S_i}^{\nu_k}$ describing the direct influence of a change of S_i on the rates ν_k and the control coefficient expressing the control of ν_k over J_j .

The connectivity theorem between concentration control coefficients and elasticity coefficients reads

$$\sum_{k=1}^r C_{\nu_k}^{S_h} \varepsilon_{S_i}^{\nu_k} = -\delta_{hi}. \quad (4.80)$$

Again, the sum runs over all rates ν_k , while S_h and S_i are the concentrations of two fixed metabolites. The symbol $\delta_{hi} = \begin{cases} 0, & \text{if } h \neq i \\ 1, & \text{if } h = i \end{cases}$ is the so-called Kronecker symbol.

In matrix formulation, the connectivity theorems read

$$C^J \cdot \varepsilon = \mathbf{0} \quad \text{and} \quad C^S \cdot \varepsilon = -I, \quad (4.81)$$

where I denotes the identity matrix of size $n \times n$. For nonnormalized coefficients, it holds

$$\tilde{C}^J \cdot \tilde{\varepsilon} \cdot L = \mathbf{0} \quad \text{and} \quad \tilde{C}^S \cdot \tilde{\varepsilon} \cdot L = -L, \quad (4.82)$$

Example 4.6

To calculate the control coefficients, we study the following reaction system:



The flux control coefficients obey the theorems

$$C_1^J + C_2^J = 1 \quad \text{and} \quad C_1^J \varepsilon_S^1 + C_2^J \varepsilon_S^2 = 0, \quad (4.85)$$

which can be solved for the control coefficients to yield

$$C_1^J = \frac{\varepsilon_S^2}{\varepsilon_S^2 - \varepsilon_S^1} \quad \text{and} \quad C_2^J = \frac{-\varepsilon_S^1}{\varepsilon_S^2 - \varepsilon_S^1}. \quad (4.86)$$

Since usually $\varepsilon_S^1 < 0$ and $\varepsilon_S^2 > 0$ (see Example 4.4), both control coefficients assume positive values $C_1^J > 0$ and $C_2^J > 0$. This means, that both reactions exert a positive control over the steady-state flux, and acceleration of any of them leads to increase of J , which is in accordance with common intuition.

The concentration control coefficients fulfill

$$C_1^S + C_2^S = 0 \quad \text{and} \quad C_1^S \varepsilon_S^1 + C_2^S \varepsilon_S^2 = -1, \quad (4.87)$$

which yields

$$C_1^S = \frac{1}{\varepsilon_S^2 - \varepsilon_S^1} \quad \text{and} \quad C_2^S = \frac{-1}{\varepsilon_S^2 - \varepsilon_S^1}. \quad (4.88)$$

With $\varepsilon_S^1 < 0$ and $\varepsilon_S^2 > 0$, we get $C_1^S > 0$ and $C_2^S < 0$, that is, increase of the first reaction causes a raise in the steady-state concentration of S while acceleration of the second reaction leads to the opposite effect.

where L is the link matrix that expresses the relation between independent and dependent rows in the stoichiometric matrix (Section 3.15, Eq. (3.22)). A comprehensive representation of both summation and connectivity theorems for nonnormalized coefficients is given by the following equation:

$$\begin{pmatrix} \tilde{C}^J \\ \tilde{C}^S \end{pmatrix} \cdot (\mathbf{K} \cdot \tilde{\varepsilon} \cdot L) = \begin{pmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & -L \end{pmatrix}. \quad (4.83)$$

The summation and connectivity theorem together with the structural information of the stoichiometric matrix are sufficient to calculate the control coefficients for a metabolic network. This shall be illustrated for a small network in the next example.

4.2.3

Matrix Expressions for Control Coefficients

After having introduced the theorems of MCA, we will derive expressions for the control coefficients in matrix form. These expressions are suited for calculating the coefficients even for large-scale models. We start from the steady-state condition

$$\mathbf{N}v(\mathbf{S}^{ss}(\mathbf{p}), \mathbf{p}) = \mathbf{0}. \quad (4.89)$$

Implicit differentiation with respect to the parameter vector \mathbf{p} yields

$$\mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \frac{\partial \mathbf{S}^{ss}}{\partial \mathbf{p}} + \mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{p}} = \mathbf{0}. \quad (4.90)$$

If we chose reaction specific parameters for perturbation, the matrix of nonnormalized parameter elasticities contains nonzero entries in the main diagonal and zeros elsewhere (compare Eq. (4.64)).

$$\frac{\partial \mathbf{v}}{\partial \mathbf{p}} = \begin{pmatrix} \frac{\partial v_1}{\partial p_1} & 0 & 0 \\ 0 & \frac{\partial v_2}{\partial p_2} & 0 \\ 0 & 0 & \frac{\partial v_r}{\partial p_r} \end{pmatrix}. \quad (4.91)$$

Therefore, this matrix is regular and has an inverse. Furthermore, we consider the Jacobian matrix

$$\mathbf{M} = \mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{S}} = \mathbf{N} \tilde{\varepsilon}. \quad (4.92)$$

The Jacobian \mathbf{M} is a regular matrix if the system is asymptotically stable and contains no conservation

relations. The case with conservation relations is considered below. Here, we may premultiply Eq. (4.90) by the inverse of \mathbf{M} and rearrange to get

$$\frac{\partial \mathbf{S}^{ss}}{\partial \mathbf{p}} = -\left(\mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{S}}\right)^{-1} \mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{p}} = -\mathbf{M}^{-1} \mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{p}} \equiv \tilde{\mathbf{R}}^S. \quad (4.93)$$

As indicated, $\partial \mathbf{S}^{ss}/\partial \mathbf{p}$ is the matrix of nonnormalized response coefficients for concentrations. Postmultiplication by the inverse of the nonnormalized parameter elasticity matrix gives

$$\frac{\partial \mathbf{S}^{ss}}{\partial \mathbf{p}} \left(\frac{\partial \mathbf{v}}{\partial \mathbf{p}} \right)^{-1} = -\left(\mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{S}}\right)^{-1} \mathbf{N} = \tilde{\mathbf{C}}^S. \quad (4.94)$$

This is the matrix of nonnormalized concentration control coefficients. The right (middle) site contains no parameters. This means, that the control coefficients do not depend on the particular choice of parameters to exert the perturbation as long as Eq. (4.64) is fulfilled. The control coefficients are only dependent on the structure of the network represented by the stoichiometric matrix \mathbf{N} , and on the kinetics of the individual reactions, represented by the nonnormalized elasticity matrix $\tilde{\mathbf{e}} = \partial \mathbf{v} / \partial \mathbf{S}$.

The implicit differentiation of

$$\mathbf{J} = \mathbf{v}(\mathbf{S}^{ss}(\mathbf{p}), \mathbf{p}), \quad (4.95)$$

with respect to the parameter vector \mathbf{p} leads to

$$\frac{\partial \mathbf{J}}{\partial \mathbf{p}} = \frac{\partial \mathbf{v}}{\partial \mathbf{p}} + \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \frac{\partial \mathbf{S}^{ss}}{\partial \mathbf{p}} = \left(\mathbf{I} - \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \left(\mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \right)^{-1} \mathbf{N} \right) \frac{\partial \mathbf{v}}{\partial \mathbf{p}} \equiv \tilde{\mathbf{R}}^J. \quad (4.96)$$

This yields, after some rearrangement, an expression for the nonnormalized flux control coefficients:

$$\frac{\partial \mathbf{J}}{\partial \mathbf{p}} \left(\frac{\partial \mathbf{v}}{\partial \mathbf{p}} \right)^{-1} = \mathbf{I} - \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \left(\mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \right)^{-1} \mathbf{N} = \tilde{\mathbf{C}}^J. \quad (4.97)$$

The normalized control coefficients are (by use of Eq. (4.69))

$$\mathbf{C}' = \mathbf{I} - (dg\mathbf{J})^{-1} \left(\frac{\partial \mathbf{v}}{\partial \mathbf{S}} \left(\mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \right)^{-1} \mathbf{N} \right) (dg\mathbf{J}) \quad \text{and}$$

$$\mathbf{C}^S = -(dg\mathbf{S}^{ss})^{-1} \left(\left(\mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \right)^{-1} \mathbf{N} \right) (dg\mathbf{J}). \quad (4.98)$$

These equations can easily be implemented for numerical calculation of control coefficients or used for analytical computation.

They are also suited for derivation of the theorems of MCA. The summation theorems for the control

coefficients follow from Eq. (4.98) by postmultiplication with the vector $\mathbf{1}$ (the row vector containing only 1s), and consideration of the relations $(dg\mathbf{J}) \cdot \mathbf{1} = \mathbf{J}$ and $\mathbf{N}\mathbf{J} = \mathbf{0}$, as shown below:

$$\begin{aligned} \mathbf{C}' \mathbf{1} &= \mathbf{I} \cdot \mathbf{1} - (dg\mathbf{J})^{-1} \left(\frac{\partial \mathbf{v}}{\partial \mathbf{S}} \left(\mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \right)^{-1} \mathbf{N} \right) (dg\mathbf{J}) \mathbf{1} \\ \mathbf{C}' \mathbf{1} &= \mathbf{1} - (dg\mathbf{J})^{-1} \left(\frac{\partial \mathbf{v}}{\partial \mathbf{S}} \left(\mathbf{N} \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \right)^{-1} \mathbf{N} \right) \mathbf{J} = \mathbf{1} - \mathbf{0} \end{aligned} \quad (4.99)$$

The connectivity theorems result from postmultiplication of Eq. (4.98) with the elasticity matrix $\mathbf{e} = (dg\mathbf{J})^{-1} \cdot (\partial \mathbf{v} / \partial \mathbf{S}) \cdot dg\mathbf{S}^{ss}$, and using that multiplication of a matrix with its inverse yields the identity matrix \mathbf{I} of respective type.

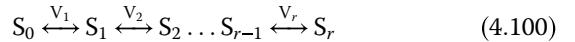
If the reaction system involves conservation relations, we eliminate dependent variables as explained in Section 1.2.4. In this case, the nonnormalized coefficients read

$$\begin{aligned} \tilde{\mathbf{C}}^J &= \mathbf{I} - \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \mathbf{L} \left(\mathbf{N}_R \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \right)^{-1} \mathbf{N}_{indep} \quad \text{and} \\ \tilde{\mathbf{C}}^S &= -\mathbf{L} \left(\mathbf{N}_R \frac{\partial \mathbf{v}}{\partial \mathbf{S}} \right)^{-1} \mathbf{N}_{indep}, \end{aligned} \quad (4.99)$$

and the normalized control coefficients are obtained by applying Eq. (4.69).

An example for calculation of flux control coefficients can be found in the web material.

To investigate implications of control distribution, we will now analyze the control pattern in an unbranched pathway:



with linear kinetics $v_i = k_i S_{i-1} - k_{-i} S_i$, the equilibrium constants $q_i = k_i/k_{-i}$, and fixed concentrations of the external metabolites, S_0 and S_r . In this case, one can calculate an analytical expression for the steady-state flux,

$$J = \frac{S_0 \prod_{j=1}^r q_j - S_r}{\sum_{l=1}^r \frac{1}{k_l} \prod_{m=l}^r q_m}, \quad (4.101)$$

as well as an analytical expression for the flux control coefficients

$$C'_i = \left(\frac{1}{k_i} \prod_{j=i}^r q_j \right) \cdot \left(\sum_{l=1}^r \frac{1}{k_l} \prod_{m=l}^r q_m \right)^{-1}. \quad (4.102)$$

Let us consider two very general cases. First assume that all reactions have the same individual kinetics, $k_i = k_+$, $k_{-i} = k_-$ for $i = 1, \dots, r$ and that the equilibrium constants, which are also equal, satisfy $q = k_+/k_- > 1$. In this case, the ratio of two subsequent flux control coefficients

is

$$\frac{C_i^J}{C_{i+1}^J} = \frac{k_{i+1}}{k_i} q_i = q > 1. \quad (4.103)$$

Hence, the control coefficients of the preceding reactions are larger than the control coefficients of the succeeding reactions and flux control coefficients are higher in the beginning of a chain than in the end. This is in agreement with the frequent observation that flux control is strongest in the upper part of an unbranched reaction pathway.

Now assume that the individual rate constants might be different, but that all equilibrium constants are equal to one, $q_i = 1$ for $i = 1, \dots, r$. This implies $k_i = k_{-i}$. Equation (4.102) simplifies to

$$C_i^J = \frac{1}{k_i} \cdot \left(\sum_{l=1}^r \frac{1}{k_l} \right)^{-1}. \quad (4.104)$$

Example 4.7

Assume that we can manipulate the pathway shown in Figure 4.12 by changing the enzyme concentration in a predefined way. We would like to explore the effect of the perturbation of the individual enzymes. For a linear pathway (see Eqs. (4.100)–(4.102)) consisting of four consecutive reactions, we calculate the flux control coefficients. For $i = 1, \dots, 4$, it shall hold that (i) all enzyme concentrations are $E_i = 1$, (ii) the rate constants are $k_i = 2, k_{-i} = 1$, and (iii) the concentrations of the external reactants are $S_0 = S_4 = 1$. The resulting flux is $J = 1$ and the flux control coefficients are $C^J = (0.533 \ 0.267 \ 0.133 \ 0.067)^T$ according to Eq. (4.98).

If we now perturb slightly the first enzyme, let's say perform a percentage change of its concentration, that is, $E_1 \rightarrow E_1 + 1\%$, then Eq. (4.54) implies that the flux increases as $J \rightarrow J + C_1^J \cdot 1\%$. In fact, the flux in the new steady state is $J^{E_1 \rightarrow 1.01 \cdot E_1} = 1.00531$. Increasing E_2, E_3 , or E_4 by 1% leads to flux values of 1.00265, 1.00132, and 1.00066, respectively. A strong perturbation would not yield similar effects. This is illustrated in Figure 4.10.

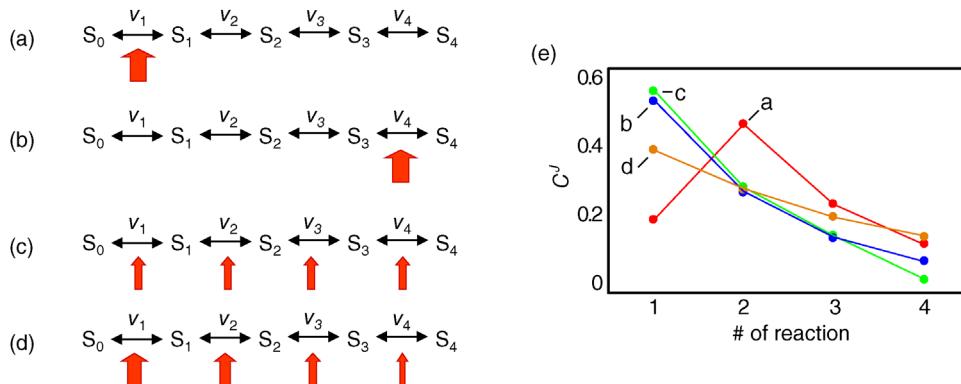


Figure 4.10 Effect of enzyme concentration change on steady-state flux and on flux control coefficients in an unbranched pathway consisting of four reactions. In the reference state, all enzymes have the concentration 1 (in arbitrary units), the control distribution is the same as in case (C), and the steady-state flux is $J = 1$. (a) Change of $E_1 \rightarrow 5E_1$ while keeping the other enzyme concentrations constant results in a remarkable drop of control of the first enzyme. The resulting flux is $J^{E_1 \rightarrow 5E_1} = 1.7741$. (b) The change $E_4 \rightarrow 5E_4$ corresponds to $J^{E_4 \rightarrow 5E_4} = 1.0563$. There is only slight change of control distribution. (c) Equal enzyme concentrations with $E_i \rightarrow 2E_i, i = 1, \dots, 4$ results in $J^{E_i \rightarrow 2E_i} = 2$. (d) Optimal distribution of enzyme concentration $E_1 = 3.124, E_2 = 2.209, E_3 = 1.562, E_4 = 1.105$ resulting in the maximal steady state flux $J^{\max} = 2.2871$.

Consider now the relaxation time $\tau_i = 1/(k_i + k_{-i})$ (see Section 4.3) as a measure for the rate of an enzyme. The flux control coefficient reads

$$C_i^J = \frac{\tau_i}{\tau_1 + \tau_2 + \dots + \tau_r}. \quad (4.105)$$

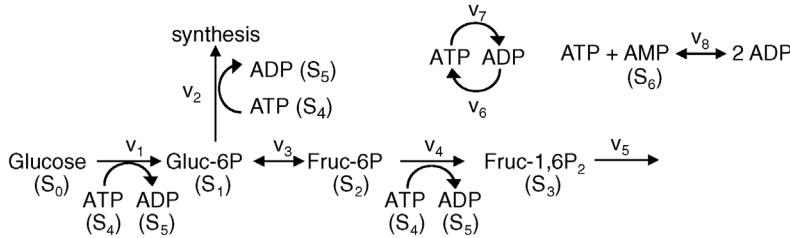
This expression helps to elucidate two aspects of metabolic control. First, all enzymes participate in the control since all enzymes have a positive relaxation time. There is no enzyme that has all control; that is, determines the flux through the pathway alone. Second, slow enzymes with a higher relaxation time exert in general more control than fast enzymes with a short relaxation time.

The predictive power of flux control coefficients for directed changes of flux is illustrated in the following example.

4.2.4

Upper Glycolysis as Realistic Model Example

Metabolic control analysis can also be easily applied to branched networks with conservation relations. Here, the matrix formulation is especially helpful. Consider the following model describing the dynamics of upper glycolysis, an essential pathway in the central carbon metabolism.



Reaction 1 denotes hexokinase, reaction 2 summarizes synthesis reactions branching off from glucose-6-phosphate. Reaction 3 is the phosphoglucoisomerase, reaction 4 the phosphofructokinase, and reaction 5 the aldolase. Reactions 6 and 7 denote other ATP consuming or producing reactions in metabolism, and reaction 8 is the adenylate kinase converting AMP and ATP into 2 ADP. The stoichiometric matrix, reduced stoichiometric matrix, and link matrix (compare Section 3.1.5) of this model read:

$$\begin{aligned} N &= \begin{pmatrix} 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & -1 & 0 & 1 & -1 & -1 \\ 1 & 1 & 0 & 1 & 0 & -1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \end{pmatrix}, \\ N_{\text{indep}} &= \begin{pmatrix} 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ -1 & -1 & 0 & -1 & 0 & 1 & -1 & -1 \\ 1 & 1 & 0 & 1 & 0 & -1 & 1 & 2 \end{pmatrix}, \\ L &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -1 & -1 \end{pmatrix}. \end{aligned} \quad (4.107)$$

Using the following kinetic expressions and kinetic parameters, we can calculate flux and concentration control coefficients:

$$v_1 = \frac{V_{\max 1} \cdot S_0 \cdot S_4}{(1 + (S_4/K_{M4}) + (S_0/K_{M0}) + (S_4/K_{M4}) \cdot (S_0/K_{M0}))}$$

with $V_{\max 1} = 1398; S_0 = 12.82; K_{M0} = 0.37; K_{M4} = 0.1$,

$$v_2 = k_2 \cdot S_1 \cdot S_4 \quad \text{with } k_2 = 0.226,$$

(4.106)

$$v_3 = \frac{V_{\max 3} \cdot ((S_1/K_{M1}) + (S_2/K_{M2}))}{(1 + (S_1/K_{M1}) + (S_2/K_{M2}))}$$

with $V_{\max 3} = 140; K_{M1} = 0.8; K_{M2} = 0.15$,

$$v_4 = \frac{V_{\max 4} \cdot S_2^2}{k_4(1 + k'_4(S_4^2/S_6^2) + S_2^2)}$$

with $V_{\max 1} = 44.7; k_4 = 0.021; k'_4 = 0.15$,

$$v_5 = k_5 \cdot S_3 \quad \text{with } k_5 = 6.05,$$

$$v_6 = k_6 \cdot S_5 \quad \text{with } k_6 = 68.48,$$

$$v_7 = k_7 \cdot S_4 \quad \text{with } k_7 = 3.21, \quad \text{and}$$

$$v_8 = k_{8f} \cdot S_4 \cdot S_6 - k_{8r} \cdot S_2^2 \quad \text{with } k_{8f} = 433; k_{8r} = 133.$$

The resulting values for the control coefficients are represented in gray scale in Figure 4.11. We see that the rates have very different control on the steady-state fluxes and steady-state concentrations. Most interesting are reaction 1 exerting positive control (due to glucose uptake) over all reactions except of the synthesis reaction (2) and the general ATP consumption (7). This is due to the fact that in this model ATP is mainly consumed for phosphorylation of glucose; ATP producing steps of lower glycolysis are only represented by reaction 6, which has therefore positive control over synthesis (2).

Reaction 6 also has positive control over S_1, S_2 , and S_4 (due to providing ATP) and negative control over S_3, S_5 , and S_6 .

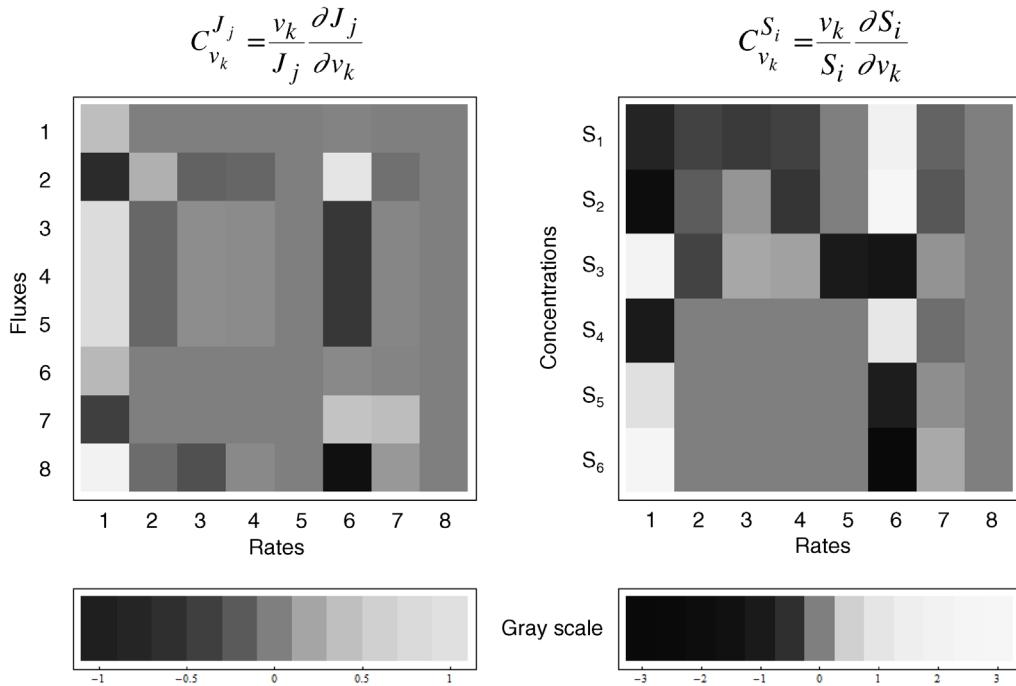


Figure 4.11 Flux and concentration control coefficients for a model of the upper glycolysis (Eqs. (4.106) and (4.107)). Gray scales express the values of the coefficients (dark gray to black – negative, light gray to white – positive).

4.2.5

Time-Dependent Response Coefficients

Metabolic control and response analysis has experienced a number of upgrades and extensions. We found that time-dependent response coefficients are especially helpful for systematic detection of the effect of a parameter change on time courses of the biochemical network, especially during parameter estimation. Time-dependent response coefficients quantify the effect of a parameter change on the dynamic concentration of a compound S , as given by

$$\tilde{R}_p^S(t) = \left. \frac{\partial S(t, p)}{\partial p} \right|_{p=p_0} \quad (4.106)$$

[26]. Again, we assume that concentration changes over time are given by the balance Eq. (3.5), that is,

$$\frac{d}{dt} S(t) = \mathbf{N}\mathbf{v}(S(t), \mathbf{p}, t). \quad (4.107)$$

To account not only for kinetic parameters, but also for initial conditions, a new vector \mathbf{q} is introduced comprising both types of quantities

$$\mathbf{q} = \begin{bmatrix} \mathbf{p} \\ S_0 \end{bmatrix}. \quad (4.108)$$

The temporal change of response coefficients (in a system without conservation relations) then reads

$$\frac{\partial}{\partial t} \frac{\partial \mathbf{S}(t)}{\partial \mathbf{q}} = \frac{\partial}{\partial t} \tilde{\mathbf{R}}_q^S(t) = \mathbf{N} \left[\frac{\partial \mathbf{v}(t)}{\partial \mathbf{S}} \frac{\partial \mathbf{S}(t)}{\partial \mathbf{q}} + \frac{\partial \mathbf{v}(t)}{\partial \mathbf{q}} \right]. \quad (4.109)$$

As before (Eq. (4.69)), the response coefficients can be normalized when conservation relations have to be respected, the expression to calculate the time-dependent response coefficients also contains the link matrix \mathbf{L} (Section 3.1.5)

$$\begin{aligned} \frac{\partial}{\partial t} \frac{\partial \mathbf{S}(t)}{\partial \mathbf{q}} &= \frac{\partial}{\partial t} \tilde{\mathbf{R}}_q^S(t) \\ &= \mathbf{N}_{\text{indep}} \left[\frac{\partial \mathbf{v}(t)}{\partial \mathbf{S}} \mathbf{L} \frac{\partial \mathbf{S}_{\text{indep}}(t)}{\partial \mathbf{q}} + \frac{\partial \mathbf{v}(t)}{\partial \mathbf{S}_{\text{dep}}} \frac{\partial \mathbf{T}}{\partial \mathbf{q}} + \frac{\partial \mathbf{v}(t)}{\partial \mathbf{q}} \right], \end{aligned} \quad (4.110)$$

where $\mathbf{T} = \mathbf{S}_{\text{dep}} - \mathbf{L}' \mathbf{S}_{\text{indep}}$ (see also Eq. (3.25)). An illustration of the behavior of time-dependent response coefficients is given in the following example:

Example 4.8

Control analysis can be applied to both metabolic and signaling networks. The small network shown in Figure 4.12a could represent both cases. It shows the production (and degradation) of compound S_1 that modifies the conversion of S_4 into S_2 . Such a reaction cycle between two components where both reactions are catalyzed by different enzymes occurs both in metabolic networks (e.g., the conversion of fructose-6-phosphate to fructose-1,6-bisphosphate by phosphofructokinase, PFK, and the reverse reaction catalyzed by fructose bisphosphatase, an important part of glycolysis, see Section 12.1) and in signaling pathways (e.g., the activation of a small G-protein by exchange of GDP with GTP catalyzed by a guanine-nucleotide exchange factor, GEF, and reversely the hydrolysis of GTP to GDP catalyzed by GTPase activating proteins, GAPs, see Section 12.2). In our case, S_2 catalyzes the formation of the next compound S_3 , which is in turn degraded. The stoichiometric matrix for this network reads

$$\mathbf{N} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 & 0 & 0 \end{pmatrix}.$$

Application of stoichiometric analysis (Chapter 3) reveals a conservation relation, that is, $S_2 + S_4 = \text{constant}$, for all times (Section 3.1.5) and three independent fluxes in steady state, that is, $\mathbf{k}_1 = (1 \ 1 \ 0 \ 0 \ 0 \ 0)^T$, $\mathbf{k}_2 = (0 \ 0 \ 1 \ 1 \ 0 \ 0)^T$, and $\mathbf{k}_3 = (0 \ 0 \ 0 \ 0 \ 1 \ 1)^T$. These independent fluxes are illustrated in Figure 4.12b by different colors. The time-dependent response depends on the initial conditions for the system, that is, whether the system is in steady state when a parameter is perturbed (Figure 4.12c) at a state far away from equilibrium (Figure 4.12d). Looking at the case where we start at steady state, middle panel, we see that a perturbing $S_3(0)$ has initially the largest effect on $S_3(t)$ as indicated by a response coefficient of 1. But this effect declines over time since systems dynamics would lead the system back to its original steady state. Increasing $S_4(0)$ would, however, increase in the long run due to an increase of the conserved moiety of S_2 and S_4 . The impact of parameter values is zero at time point $t=0$, but then increases for producing reactions such as k_1 and k_5 and decreases for degrading reactions such as k_2 and k_6 . When looking at the response of S_4 (lower panel), we find that increasing $S_3(0)$ has only temporarily a diminishing effect on $S_4(t)$, but not with respect to the new steady state. Parameters k_5 and k_6 have no effect on $S_4(t)$.

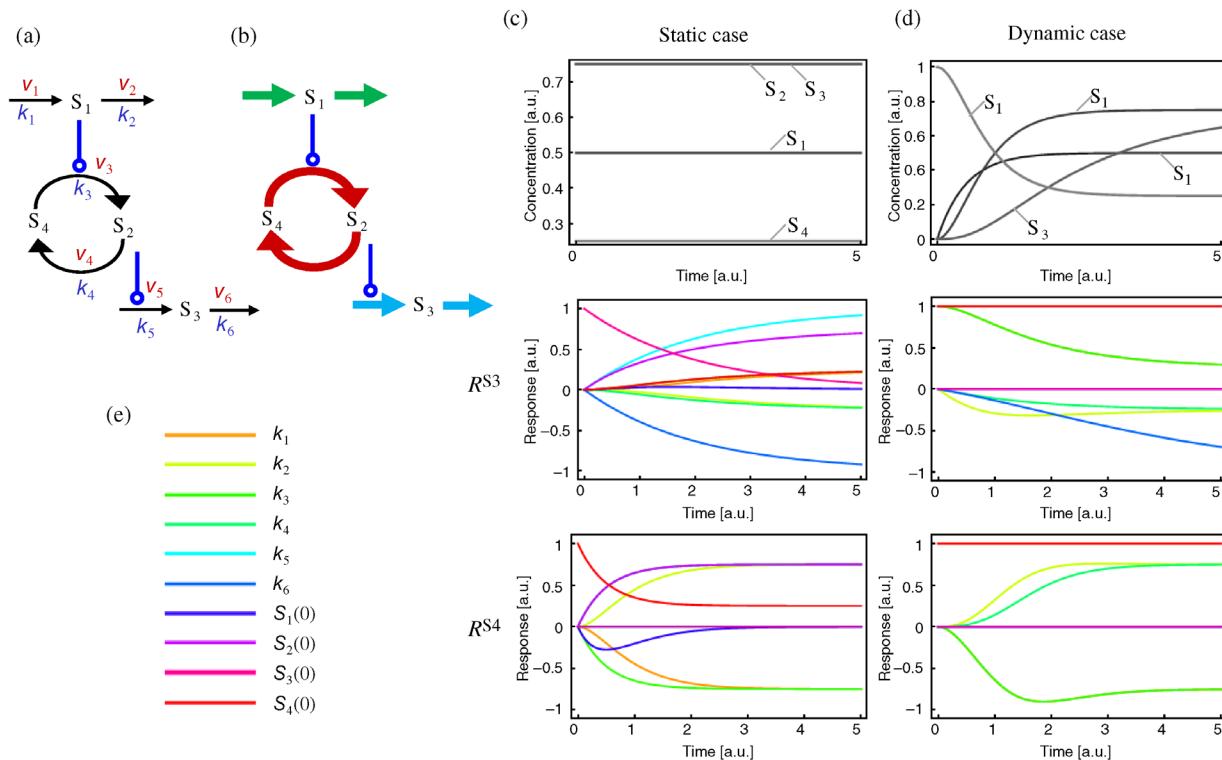


Figure 4.12 Illustration of time-dependent flux response. (a) Example network, (b) Independent steady-state fluxes for the production and degradation of S_1 , for the interconversion of S_2 and S_4 , and for the production and degradation of S_3 are shown by green, red, and blue arrows, respectively. (c) Response analysis for the network at steady state: Top panel: constant concentrations of the substrates, Middle panel: temporal behavior of all response coefficients for S_3 , and Lower panel: temporal behavior of all response coefficients for S_4 . (d) Response analysis for the system starting at the initial conditions $S_1(0) = S_2(0) = S_3(0) = 0$; $S_1(0) = 1$. Panels as in C. (e) Color code for the parameters of the system as used in the middle and lower panels.

Parameter values: $k_1 = 1$; $k_2 = 2$; $k_3 = 3$; $k_4 = k_5 = k_6 = 0.5$.

Exercises

- 1) Compare the shapes of different kinetic laws.
 - a) Create a plot of rate v versus concentration S for a reaction with mass action kinetics with $k = 1$, a reaction with Michaelis–Menten kinetics with $V_{\max} = 1, K_m = 1$ and a reaction with Hill kinetics with $V_{\max} = 1, K_m = 1$, and $n = 4$.
 - b) Create a plot v against S for a reaction with Michaelis–Menten kinetics. Vary V_{\max} and K_m .
 - c) Create a plot v against S for a reaction with Hill kinetics. Vary either K_B or n .
 - d) Create a plot v against S for a reaction with Monod–Wyman–Changeux kinetics (Eq. (4.46)). Vary K_R , n , or L . Compare to the results for Hill kinetics.
 - e) Plot reaction rates against substrate concentrations for the different types of inhibition presented in Table 4.3.
 - 2) Assign the following kinetics to network N3 in Chapter 3: $v_1 = k_1, v_2 = \frac{V_{\max 2} \cdot S_1}{K_{m2} + S_1}, v_3 = \frac{V_{\max 3} \cdot S_1}{K_{m3} + S_1}$ with $k_1 = 1.8, V_{\max 2} = 3, K_{m2} = 0.2, V_{\max 3} = 5, K_{m3} = 0.4$. Compute the steady-state concentration of S_1 and calculate the flux control coefficients.
 - 3) For the reaction system $A \xrightarrow{v_1} B, B \xrightarrow{v_2} C, C \xrightarrow{v_3} A$ with $v_1 = k_1 \cdot A, v_2 = k_2 \cdot B, v_3 = k_3 \cdot C$, and $k_1 = 2, k_2 = 2, k_3 = 1$, write down the set of systems equations.
 - a) Compute the Jacobian J !
 - b) Determine the eigenvalues and eigenvectors of the Jacobian J !
- c) What is the general solution of the ordinary differential equation system?
- d) Compute the solution with the initial condition $A(0) = 1, B(0) = 1, C(0) = 0$!
- 4) Assign following kinetics to the network given below:
- $$v_1 = k_1, v_2 = \frac{V_{\max 2} \cdot S_1}{K_{m2} + S_1} \text{ with } k_1 = 1, V_{\max 2} = 2.5, K_{m2} = 2.$$
-
- a) Calculate the concentration of S_1 in steady state!
 - b) Calculate the elasticity coefficients of v_1 and v_2 with respect to substrate S_1 !
 - c) Calculate the flux control coefficients!
 - d) Could this system attain a steady state if $k_1 = 2.5, V_{\max 2} = 1, K_{m2} = 2$?
- 5) If the rate $v = v(S)$ is given as Hill kinetics. What is the corresponding elasticity coefficient ε_S^v ?
- 6) Load a biochemical network from a suitable database (Biomodels.org or JWSonline at jjj.biochem.sun.ac.za/). Calculate steady-state concentrations and control coefficients.
- 7) What is the difference of flux control coefficients in a linear unbranched reaction pathway, if we describe all reactions either with reversible or irreversible rate laws?

References

- 1 Waage, P. and Guldberg, C.M. (1864) *Studies concerning affinity*, Forhandliger, Videnskabs-Selskabet, Christiania, pp 35.
- 2 Guldberg, C.M. and Waage, P. (1867) *Études sur les affinités chimiques*, Christiania.
- 3 Guldberg, C.M. and Waage, P. (1879) Über die chemische Affinität. *J. Prakt. Chem.*, 19, 69.
- 4 Brown., A.J. (1902) Enzyme action. *J. Chem. Soc.*, 81, 373–386.
- 5 Michaelis, L. and Menten, M.L. (1913) Kinetik der Invertinwirkung. *Biochem. Z.*, 49, 333–369.
- 6 Briggs, G.E. and Haldane, J.B.S. (1925) A note on the kinetics of enzyme action. *Biochem. J.*, 19, 338–339.
- 7 Lineweaver, H. and Burk, D. (1934) The determination of enzyme dissociation constants. *J. Am. Chem. Soc.*, 56, 658–660.
- 8 Eadie, G.S. (1942) The inhibition of cholinesterase by physostigmine and prostigmine. *J. Biol. Chem.*, 146, 85–93.
- 9 Hanes, C.S. (1932) Studies on plant amylases. I. The effect of starch concentration upon the velocity of hydrolysis by the amylase of germinated barley. *Biochem. J.*, 26, 1406–1421.
- 10 Haldane, J.B.S. (1930) *Enzymes*, Longmans, Green and Co., London.
- 11 Schellenberger, A. (ed.) (1989) *Enzymkatalyse*, VEB Gustav Fischer Verlag, Jena.
- 12 Wegscheider, R. (1902) Über simultane gleichgewichte und die beziehungen zwischen thermodynamik und reaktionskinetik homogener systeme. *Z. Phys. Chem.*, 39, 257–303.
- 13 Hill, A.V. (1910) The possible effects of the aggregation of the molecules of hemoglobin on its dissociation curves. *J. Physiol.*, 40, iv–vii.
- 14 Hill, A.V. (1913) The combinations of hemoglobin with oxygen and with carbonmonoxide. *Biochem. J.*, 7, 471–480.
- 15 Monod, J., Wyman, J., and Changeux, J.P. (1965) On the nature of allosteric transitions: a plausible model. *J. Mol. Biol.*, 12, 88–118.
- 16 Heijnen, J.J. (2005) Approximative kinetic formats used in metabolic network modeling. *Biotechnol. Bioeng.*, 91, 534–545.

- 17 Liebermeister, W. and Klipp, E. (2006) Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. *Theor. Biol. Med. Model.*, 3, 42.
- 18 Kacser, H. and Burns, J.A. (1973) The control of flux. *Symp. Soc. Exp. Biol.*, 27, 65–104.
- 19 Heinrich, R. and Rapoport, T.A. (1974) A linear steady-state treatment of enzymatic chains. General properties, control and effector strength. *Eur. J. Biochem.*, 42, 89–95.
- 20 Reder, C. (1988) Metabolic control theory: a structural approach. *J. Theor. Biol.*, 135, 175–201.
- 21 Bruggeman, F.J., Westerhoff, H.V., Hoek, J.B., and Kholodenko, B.N. (2002) Modular response analysis of cellular regulatory networks. *J. Theor. Biol.*, 218, 507–520.
- 22 Liebermeister, W., Klipp, E., Schuster, S., and Heinrich, R. (2004) A theory of optimal differential gene expression. *Biosystems*, 76, 261–278.
- 23 Westerhoff, H.V., Getz, W.M., Bruggeman, F., Hofmeyr, J.H., Rohwer, J.M. et al. (2002) ECA: control in ecosystems. *Mol. Biol. Rep.*, 29, 113–117.
- 24 Hofmeyr, J.H. and Westerhoff, H.V. (2001) Building the cellular puzzle: control in multi-level reaction networks. *J. Theor. Biol.*, 208, 261–285.
- 25 Kholodenko, B.N., Brown, G.C., and Hoek, J.B. (2000) Diffusion control of protein phosphorylation in signal transduction pathways. *Biochem. J.*, (350 Pt 3), 901–907.
- 26 Ingalls, B.P. and Sauro, H.M. (2003) Sensitivity analysis of stoichiometric networks: an extension of metabolic control analysis to non-steady state trajectories. *J. Theor. Biol.*, 222, 23–36.

Further Reading

- Enzyme kinetics:** Cornish-Bowden, A. (2012) *Fundamentals of Enzyme Kinetics*, 4th edn, Wiley-Blackwell, Weinheim.
- Enzyme kinetics:** Cornish-Bowden, A. (2013) The origins of enzyme kinetics. *FEBS Lett.*, 587 (17), 2725–2730.
- Enzyme kinetics:** Liebermeister, W. and Klipp, E. (2006) Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. *Theor. Biol. Med. Model.*, 3, 41.
- Foundations of metabolic control theory, I:** Kacser, H. and Burns, J.A. (1973) The control of flux. *Symp. Soc. Exp. Biol.*, 27, 65–104.
- Foundations of metabolic control theory, II:** Heinrich, R. and Rapoport, T.A. (1974) A linear steady-state treatment of enzymatic chains. General properties, control and effector strength. *Eur. J. Biochem.*, 42 (1), 89–95.
- Mathematical formalization of metabolic control theory:** Reder, C. (1988) Metabolic control theory: a structural approach. *J. Theor. Biol.*, 135 (2), 175–201.
- Mathematical formalization of metabolic control theory:** Fell, D. (1997) *Understanding the Control of Metabolism*, Portland, London.
- Extension to non-steady states:** Ingalls, B.P. and Sauro, H.M. (2003) Sensitivity analysis of stoichiometric networks: an extension of metabolic control analysis to non-steady state trajectories. *J. Theor. Biol.*, 222 (1), 23–36.

Data Formats, Simulation Techniques, and Modeling Tools

5

The development of mathematical models of molecular and cellular systems starts by the collection of model components and their interactions. Usually, in a first step the biochemical reaction equations that define the topological structure of the reaction network and the reaction stoichiometries are formulated. For this purpose, it is often useful to draw a diagram that illustrates the network structure either of the whole model or of a particular part. Once the reaction network and its stoichiometry are defined, the mathematical details of the model can be constructed. For this purpose, often systems of ordinary differential equations (ODEs) are used. Normally, this requires very detailed information about the kinetics of the individual reactions or appropriate assumptions have to be made.

In this chapter, databases are presented that provide information on the network structure of cellular processes such as metabolic pathways and signal transduction pathways. Moreover, data formats used for the structural, mathematical, and graphical description of biochemical reaction networks are introduced. But to begin with, we will start this chapter with an overview of simulation techniques and popular software tools that support the user during model development.

5.1 Simulation Techniques and Tools

Summary

This section gives an overview of different simulation techniques and introduces tools used in systems biology. Modeling and simulation functionalities of the tools are presented.

5.1 Simulation Techniques and Tools

- Differential Equations
- Stochastic Simulations
- Simulation Tools

5.2 Standards and Formats for Systems Biology

- Systems Biology Markup Language
- BioPAX
- Systems Biology Graphical Notation

5.3 Data Resources for Modeling of Cellular Reaction Systems

- General-Purpose Databases
- Pathway Databases
- Model Databases

5.4 Sustainable Modeling and Model Semantics

- Standards for Systems Biology Models
- Model Semantics and Model Comparison
- Model Combination
- Model Validity

References

Further Reading

5.1.1 Differential Equations

In systems biology, various simulation techniques are used, such as systems of ODEs, stochastic methods, Petri nets (see Section 7.1), Boolean models (see Section 7.1), partial differential equations (PDEs), cellular automata (see Section 7.3), agent-based systems (see Section 7.3), and hybrid approaches. The use of ODEs in biological modeling is widespread and by far the most common simulation approach in computational systems biology [1,2]. The basic structure and properties of ODEs

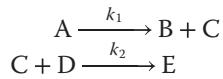
have already been introduced in Chapter 2 and more mathematical details can be found in Chapter 15. Some ODEs are simple enough to be solved analytically and have an exact solution. However, more complex ODE systems, such as those occurring in most systems biological simulations, must be solved numerically by appropriate algorithms. An early method for the numerical solution of ODEs was derived by Euler in 1768. Methods that provide improved computational accuracy are, for instance, the popular fourth-order Runge–Kutta algorithms and implicit methods that can also handle so-called “stiff” differential equations. Numerical ODE solvers advance in discrete time steps of a given step size, h . By evaluating the system of ODEs at t , $t + \frac{1}{2}h$, and $t + h$, the Runge–Kutta method achieves that the error scales with the fourth power of h . That is, if the step size is reduced by a factor of 10, the error (deviation from the analytic solution) is diminished by a factor of 10^4 . A small step size yields a high accuracy, but it also leads to longer computation times. It is therefore desirable to adjust the step size dynamically such that the error is close to a user-defined limit ϵ . A popular method that implements such an adaptive step size control is the Runge–Kutta–Fehlberg method [3]. Simulation tools for systems biology have to cope with systems of multiple reactants and multiple reactions. For the numerical integration of such complex ODE systems, they usually make use of more advanced programs such as LSODA [4,5], CVODE [6], or LIMEX [7].

ODEs can be used to describe systems in which the components are homogeneously distributed. It is, for instance, often assumed that the molecules inside a cell are homogeneously distributed, although this is clearly a simplification. In reality, molecules are synthesized or imported at a specific location and are then transported by active (e.g., along microtubules) or passive (diffusion) mechanisms within the cytosol. Furthermore, this transport can be slowed or obstructed by high viscosity or intracellular structures (e.g., nucleus, organelles, and cytoskeleton), resulting in a spatially nonhomogeneous distribution. If these effects are important, they can be modeled using PDEs. However, the construction as well as the numerical simulation of such models is significantly more difficult than ODE-based models. Later in this chapter, we will describe VCell, a tool that is designed to simulate spatial models.

5.1.2 Stochastic Simulations

Differential equations are generally used if the entities that are described by the model variables (molecules, cells, and organisms) exist in such large numbers that

they can be described by continuous values. However, if the numbers are small (i.e., dozens or a few hundred), this approximation becomes more and more unrealistic since there are, for instance, either five or six molecules or organisms and not some value in between. Furthermore, with these low numbers, random fluctuations of the reaction rates become important and can significantly influence the long-term outcome of the system. Under these conditions, it is necessary to simulate the model stochastically, which effectively means to keep track of the fate of each individual molecule. If we consider the set of chemical reactions



and denote the number of molecules of the different species X_i with $\#X_i$, then the state of the system is given by $S = (\#A, \#B, \#C, \#D, \#E)$. If the first reaction takes place, the system changes into a new state given by $S^* = (\#A - 1, \#B + 1, \#C + 1, \#D, \#E)$. The probability that a certain reaction μ occurs within the next time interval dt is given by the following equation, where a_μ is given by the product of a mesoscopic rate constant and the current particle number. The expression $o(dt)$ describes other terms that can be neglected for small dt [8].

$$P(S^*, t + dt | S, t) = a_\mu dt + o(dt).$$

One approach to proceed within this stochastic framework is to develop a so-called *master equation* (for more details see Section 7.2.2). For each possible state $(\#A, \#B, \#C, \#D, \#E)$, a probability variable is created and using the above equation and the definition of the a_μ 's, a system of coupled differential equations can be developed that describes the reaction system. The variables of this system represent transition probabilities and the equation system itself is called master equation. However, because each state requires a separate variable, the system becomes quickly intractable, as the number of chemical species and the number of each kind of molecule grow.

It was therefore a major breakthrough when Gillespie applied in the 1970s exact stochastic algorithms to the simulation of chemical reactions [9]. These algorithms are exact in the sense that they are equivalent to the results of the master equation, but instead of solving the probabilities for all trajectories simultaneously, the simulation methods calculate single trajectories. By computing many individual trajectories and studying the statistics of these trajectories, the same insights can be obtained as with the master equation. Gillespie used two exact

stochastic algorithms called *direct method* and *first reaction method*.

The direct method requires the generation of two random numbers per iteration and the computation time is proportional to the number of possible reactions. Gillespie's other algorithm, the first reaction method, works slightly different in that it generates a putative waiting time for each possible reaction (chosen from an exponential distribution). The reaction to occur next is the one with the smallest waiting time. The algorithm requires n random numbers per iteration ($n = \text{number of reactions}$) and the computation time is again proportional to the number of reactions. Gibson and Bruck [10] succeeded in considerably improving the efficiency of the first reaction method. Their elegant algorithm, called the *next reaction method*, uses only a single random number per iteration and the computation time is proportional to the logarithm of the number of reactions. This makes the stochastic approach amenable for much larger systems.

However, notwithstanding these improvements, stochastic simulations are computationally very demanding, especially if the reaction system of interest contains a mixture of species with large and small molecule numbers. In this case, the simulation will spend most of the time performing reactions of the species with large molecule numbers, although for these species a stochastic treatment is not really necessary. Even worse, the high reaction rates of those species lead to very short time intervals, τ , between individual reactions, so that the simulated reaction time advances only very slowly. Gillespie has formulated another, approximate, method that achieves significant speed improvements with only moderate losses in accuracy [11]. This τ -leap method can in principle also be used in cases of mixed reaction systems, but it loses efficiency, because the length of the appropriate time step is determined by the species with the smallest number of molecules and is on the order of the waiting times for exact algorithms. To overcome or at least mitigate this problem, hybrid systems have been developed that use approximate methods for fast reactions of a mixed system and exact methods for the slow components [12]. For a more detailed discussion of the various methods, see Section 7.2.

5.1.2.1 Stochastic and Macroscopic Rate Constants

The mesoscopic rate constants used for stochastic simulations and the macroscopic rate constants used for deterministic modeling are related, but not identical. Macroscopic rate constants depend on concentrations, while the stochastic constants depend on the number of molecules. A dimension analysis shows how the classical rate constants for reactions of first and second order have

to be transformed to be suitable for stochastic simulations.

First-Order Reaction

Let us assume that substance X decays according to a first-order reaction. The reaction rate, v , is given by $v = -k \cdot X$ with k being the rate constant. The dimensions for the deterministic description are $\frac{\text{mol}}{\text{l}\cdot\text{s}} = \frac{1}{\text{s}} \cdot \frac{\text{mol}}{\text{l}}$ and for the stochastic framework $\frac{\text{molecules}}{\text{s}} = \frac{1}{\text{s}} \cdot \text{molecules}$. The dimension of the rate constant is in both cases 1/s and thus no conversion is necessary.

Second-Order Reaction

Now let us consider a reaction in which one molecule of X and one molecule of Y interact. The reaction rate is given by $v = k \cdot X \cdot Y$. The dimensions for the deterministic equation are $\frac{\text{mol}}{\text{l}\cdot\text{s}} = \frac{1}{\text{s}\cdot\text{mol}} \cdot \frac{\text{mol}^2}{\text{l}^2}$ and for the stochastic case $\frac{\text{molecules}}{\text{s}} = \frac{1}{\text{s}\cdot\text{molecules}} \cdot \text{molecules}^2$. To convert the macroscopic rate constant from $\frac{1}{\text{s}\cdot\text{mol}}$ into $\frac{1}{\text{s}\cdot\text{molecules}}$, the numerical value has to be divided by the reaction volume and the Avogadro constant (to convert moles into molecules). Thus, a classical second-order rate constant of $1 \text{ M}^{-1} \text{s}^{-1}$ that has been measured in a reaction volume of 10^{-15} l converts to a mesoscopic rate constant of $1.66 \times 10^{-9} \text{ molecule}^{-1} \text{s}^{-1}$.

5.1.3

Simulation Tools

In the following, four different simulation tools are presented that can simulate systems of differential equations, and come along with further functionalities such as graphical visualization of the reaction network, advanced analysis techniques, and interfaces to external model and pathway databases. A compendium of further modeling and simulation tools is also given in Chapter 17 and several reviews on this topic are available [13–16].

Modeling systems have to accomplish several requirements. They must have a well-defined internal structure for the representation of model components and reactions, and optionally functionalities for the storage of a model in a well-defined structure, standardized format, or database. Further desired aspects are a user-friendly interface for model development, a graphical representation of reaction networks, a detailed description of the mathematical model, integrated simulation engines for deterministic and stochastic simulations along with a graphical representation of those simulation results, and functionalities for model analysis and model refinement. This is a very broad spectrum of functionalities. Existing tools cover different aspects of these functionalities. In the following, systems biology tools will be introduced that already accomplish several of these points.

CellDesigner, for instance, is a very popular tool in the systems biology community [16,17] and especially suitable for graphical model development. It has a user-friendly process diagram editor, uses the Systems Biology Markup Language (SBML, see Section 5.2) for model representation and exchange, and provides fundamental simulation and modeling functions. Another popular program with similar functionality is COPASI. The tool has an interface for model definition and representation and provides a multitude of methods for simulation, model analysis, and refinement such as parameter scanning, metabolic control analysis, optimization, or parameter estimation. In addition to deterministic solvers, COPASI also offers several stochastic simulation algorithms. Spatial models that are described by partial differential equations require special tools for their simulation and VCell is such a tool. It allows the construction of spatial models and is capable of simulating them deterministically or stochastically.

5.1.3.1 CellDesigner

CellDesigner provides an advanced graphical model representation along with an easy to use user interface and an integrated simulation engine [18]. The latest version of CellDesigner, released July 2014, is 4.4 (celldesigner.org). The process diagram editor of CellDesigner supports a rich set of graphical elements for the description of biochemical and gene regulatory networks. Networks can be constructed from compartments, species, and reactions. CellDesigner comes with a large number of predefined shapes that can be used for different types of molecules, such as proteins, receptors, ion channels, and small metabolites. It is also possible to modify the symbols to indicate phosphorylations or other modifications. The program also provides several icons for special reaction types such as catalysis, transport, inhibition, and activation. The graphical notation of CellDesigner almost conforms to the Systems Biology Graphical Notation (SBGN) (see Section 5.2) and SBGN compliance can also be enforced by invoking CellDesigner's SBGN Viewer, which adopts the SBGN Process Description Diagram Level 1.1. Furthermore, the graphical layout information of the model can be exported in the SBGN-ML format.

Reading and writing of the models is SBML based (see Section 5.2) and the models written by CellDesigner pass the online validation at sbml.org/validator and thus conform to the SBML standard. Models are constructed by dragging icons from the toolbar area onto the model canvas where they can be arranged as the user wishes or by applying a range of predefined layout styles. The color and size of the different icon types as well as the style of the connectors can be modified under the “Preference” menu. In addition to manually creating a model, it is also

possible to import a model from an SBML file or download it from “BioModels” (biomodels.net), “JWS Online” (jjj.biochem.sun.ac.za), or “PantherDB” (www.pantherdb.org). The tree panel on the left-hand side of the GUI (Figure 5.1) lists the compartments, species, and reactions of the model. Clicking on one of these components highlights the corresponding element in the model canvas as well as in the list area at the bottom. This tab is also the place where initial concentrations and reaction details are entered. CellDesigner allows entering arbitrary kinetic equations, but has only a very limited list of standard kinetics (mass action and irreversible Michaelis–Menten) that can be applied. However, such limitations can be eliminated by the ability to extend CellDesigner’s capabilities with the help of plug-ins. SBMLSqueezer [19], for instance, is a plug-in that generates kinetic laws for model reactions based on the SBML context of each reaction. It has a large number of options that allow the user to specify which type of kinetics is applied for which reaction type and whether the generated parameters should be global or local to the reaction. SBMLSqueezer can generate kinetics for the whole model or for single reactions (for fine tuning kinetic details). In addition, the plug-in also allows the user to convert SBML models into human-readable reports in the LaTeX or PDF format.

In the “Notes” area of CellDesigner (Figure 5.1, bottom right), it is possible to attach comments to model components. Although inconspicuous, this is a very useful feature for real-life modeling. It is often necessary to collect parameter values from diverse publications or databases (e.g., Brenda (www.brenda-enzymes.de) or Sabio-RK (sabio.villa-bosch.de)) to parameterize a model. It is then a great help to document the source for the values in the note fields attached to the different reactions.

After a model has been created, it can be analyzed using modules from the Systems Biology Workbench (SBW) (see Chapter 17), since CellDesigner can connect to and communicate with the SBW framework. If the model has been properly parameterized, it can also be numerically simulated. For this purpose, it is possible to connect to the COPASI simulation engine (see the next section) or use CellDesigner’s own solver (Figure 5.2). The interface is straightforward to use but also rather limited, since it only allows the deterministic computation of simple time curves. The connection to the COPASI engine is also restricted to the calculation of simple time curves, but allows at least to choose between deterministic and stochastic solvers. For a quick overview of the model behavior this might be sufficient, but for a more thorough understanding the better option is to save the model as SBML file and import it for further analysis into other tools such as COPASI.

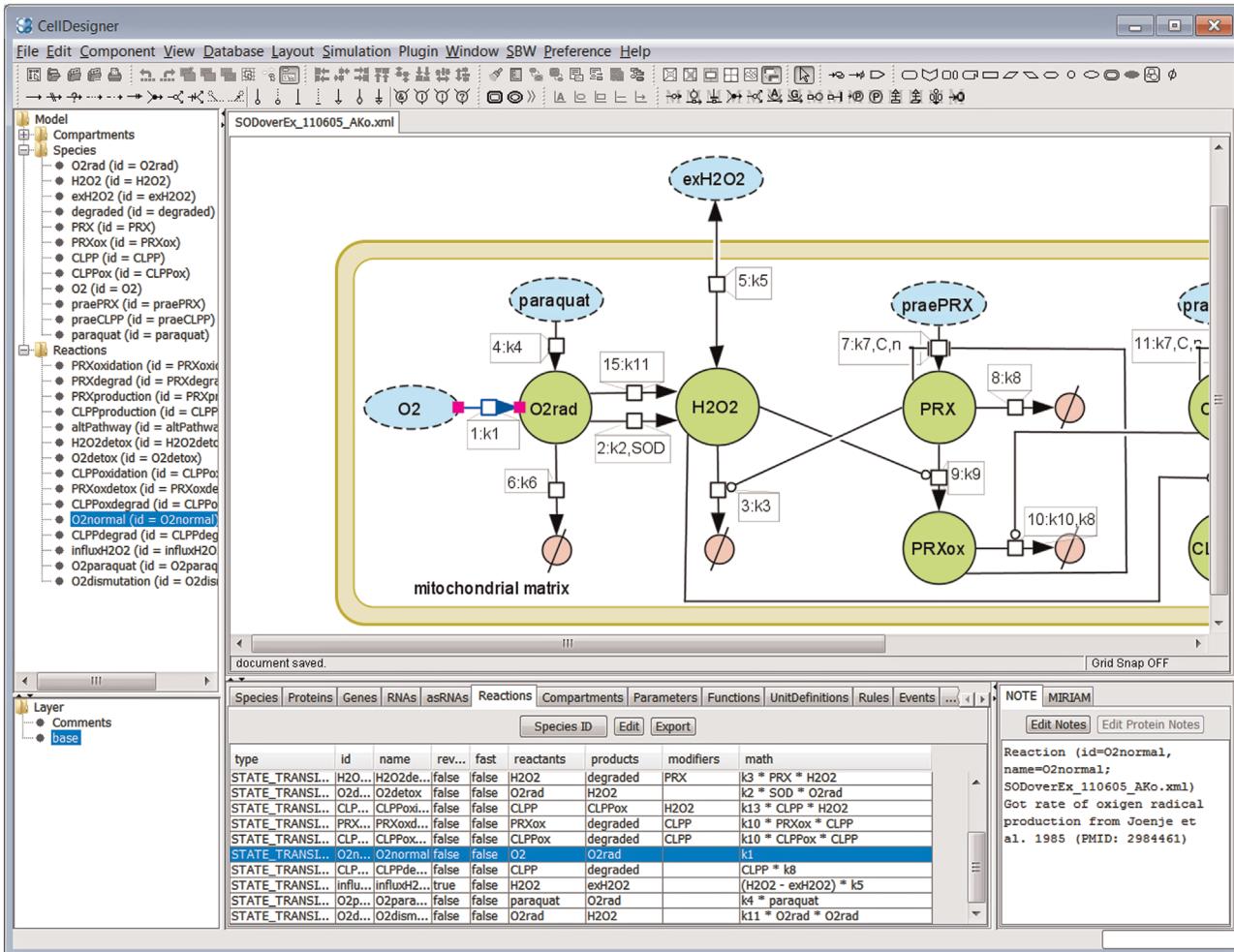


Figure 5.1 CellDesigner's GUI with several panels for the creation of SBML models. The large panel in the center contains the canvas for constructing the model by adding molecular species, reactions, and compartments. Details for elements can be specified in the panel shown at the bottom. The panel on the left-hand side allows to inspect the overall structure of the SBML model, which consists of compartments, species, and reactions.

5.1.3.2 COPASI

A powerful and user-friendly biochemical simulator that offers several unique features is COPASI [20] that currently is available as version 4.16 Build 104 (www.copasi.org/). Regarding their functionality, COPASI and CellDesigner are complementing each other. CellDesigner's drag and drop method for creating a reaction network together with its SBGN conformity for graphical representation, making it a very comfortable tool for the initial construction of the model, while its analytical and simulation capabilities are more limited. In COPASI, in contrast, reactions and species have to be entered in a textual representation (e.g., $A + B \rightarrow C$) and although COPASI allows to generate a graphical representation of the network the results are not comparable to those of

CellDesigner. If SBML models of CellDesigner are imported into COPASI, the existing layout is recognized and can be displayed. Both tools allow the definition of arbitrary kinetic laws, but COPASI provides a much larger list of predefined laws. The model structure is displayed in the usual tree-like representation, listing the typical SBML components such as compartments, species, and reactions (Figure 5.3, left-hand side). After selecting individual components, their properties such as initial concentration, type of kinetic law, or notes can be modified. For species, all reactions in which they participate are listed, which is very helpful, especially for larger models. Additionally, COPASI can also display the system of ODEs that is automatically constructed from the chemical reactions and allows them to be exported in

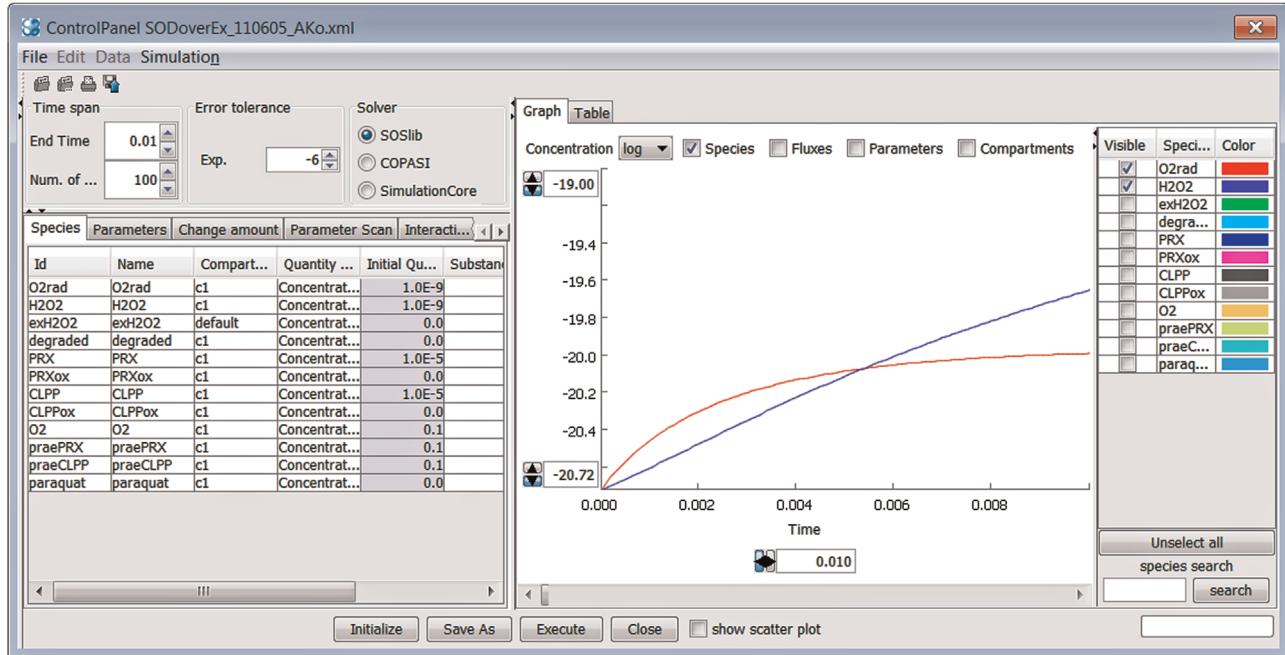


Figure 5.2 Once all species and reactions have been parameterized within CellDesigner, it is possible to perform time course simulations with the integrated numerical solver. The output can be represented as graph or table and it is possible to interactively specify the species that should be displayed.

formats for display (i.e., MathML, LaTeX, and PNG) as well as in formats for further calculation by other tools (i.e., C-source code, Berkeley Madonna format, and XPPAUT format).

However, the true strength of COPASI lies in the large number of analysis tools that are listed under the “Tasks” tab (Figure 5.3, left-hand side). Time course simulations are probably the most often used task. COPASI offers here deterministic (LSODA) as well as various stochastic algorithms (Gibson–Bruck, direct method, τ -leap, and hybrid). The results can be displayed graphically (Figure 5.4, left-hand side) or in a tabular format (called a report). The exact content of graphs or reports is very flexible and can be specified under the “Plots” and “Reports” tabs. Another important feature of COPASI is parameter scans. In its most simple form, a model parameter is varied in intervals and at each point the steady-state values of the model variables are calculated and displayed if the appropriate plot is defined (Figure 5.4, right-hand side). But this tool is much more powerful since it allows to perform nested scans of multiple parameters, and in addition to steady-state calculations also time course simulations, sensitivity analyses, or parameter estimation can be performed. Furthermore, “Repeat” and “Random Distribution” widgets can be included in the task, which allows, for instance, to perform multiple executions of

a parameter estimation task, each with slightly different initial concentrations of the model species. This is a valuable feature since parameter fitting of large models sometimes gets trapped in local minima. A variation of the starting conditions for the fitting algorithm is often used to identify such local minima. Parameter estimation, that is, adjusting model parameters such that the model output most closely resembles a set of experimental data points, is definitely one of COPASI’s strengths. It offers more than a dozen methods, including differential evolution, particle swarm, Nelder–Mead, Levenberg–Marquardt, and evolutionary programming. COPASI can use time course or steady-state data, which are imported from a file that can contain data from multiple experiments. Apart from these computationally intensive tasks, COPASI supports the analysis of the stoichiometric network (e.g., elementary modes [21]) metabolic control analysis, and can perform a sensitivity analysis.

Since some tasks of COPASI can be quite time consuming (e.g., parameter estimation), there are also options to run the tool in batch mode without the GUI. For this, either a special version of COPASI (CopasiSE) can be called directly from the command line or the COPASI API can be called from external programs. Language bindings are available for Java, Python, and C++. Finally, although COPASI models are natively saved in a

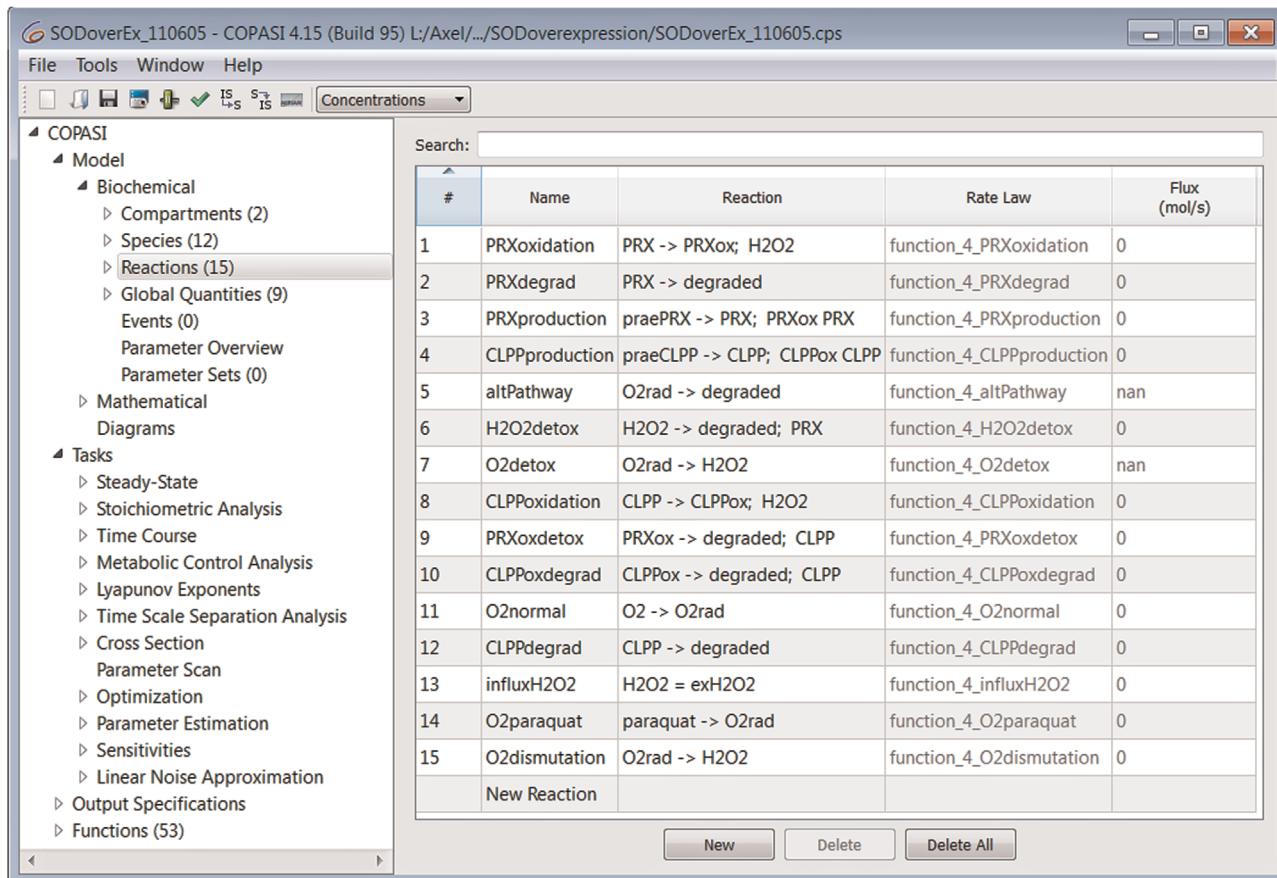


Figure 5.3 Screenshot of COPASI's GUI. A hierarchical menu (left) allows to inspect the structure of the underlying SBML model, but permits also access to a large repertoire of analysis tools as well as the different types of plots that have been created for this model.

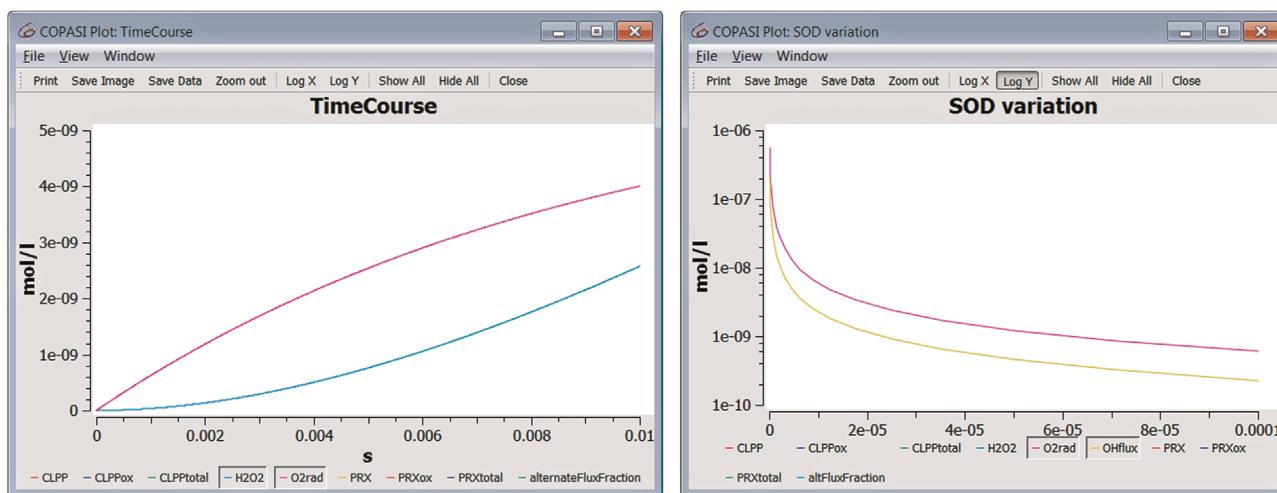


Figure 5.4 Example of different types of plots that can be generated with COPASI. In addition to the usual time course plots showing variable concentrations (left), it is also possible to construct more complex diagrams, such as the change of steady-state concentrations in response to a parameter variation (right).

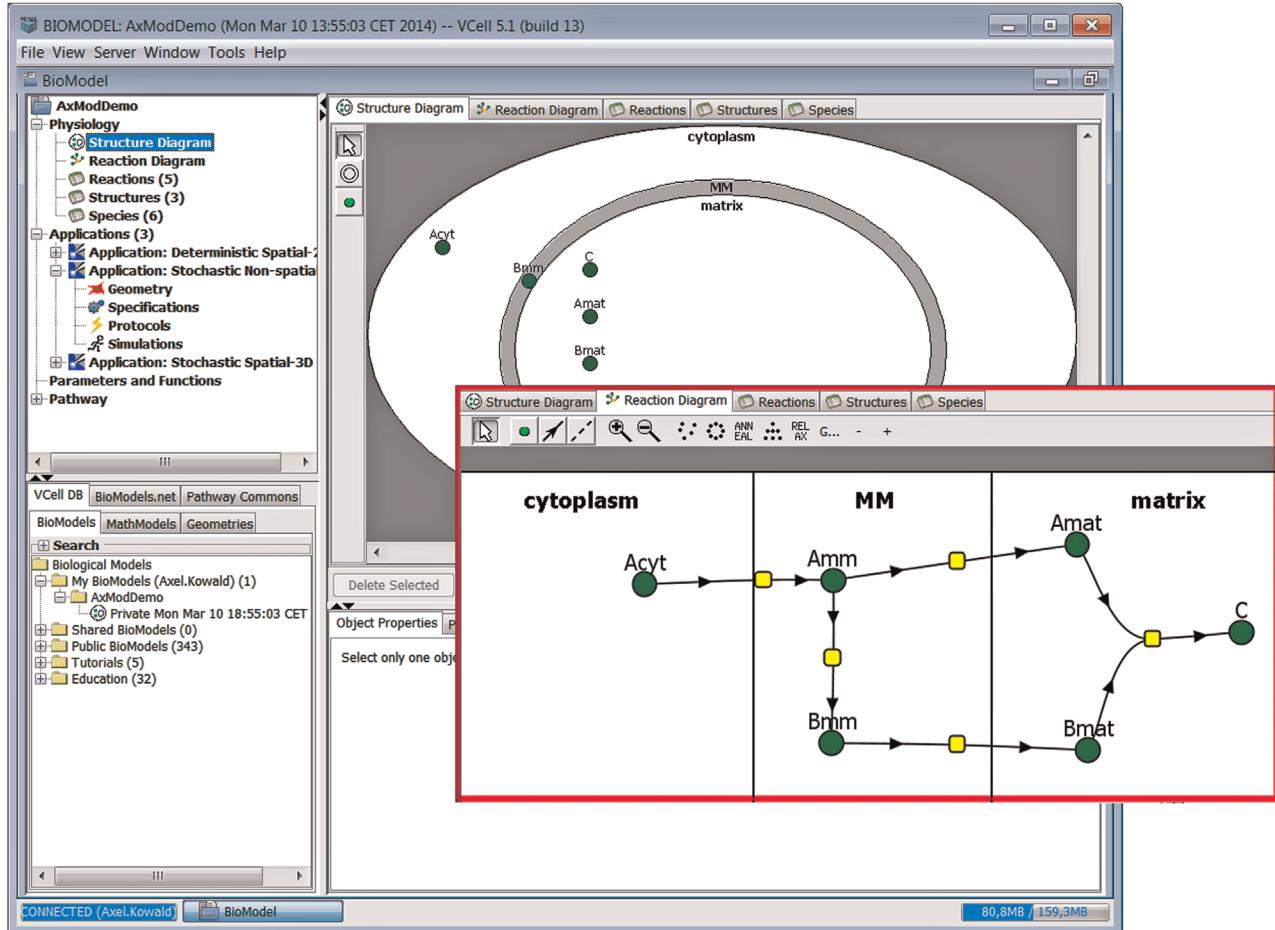


Figure 5.5 The graphical user interface of VCell 5.1, a simulation tool for compartmental as well as spatially resolved models. 2D and 3D models can be constructed using different views that show, for instance, the reactions (red inset) or the spatial location of the different species (background panel). Simulations are normally performed at the remote server and can be deterministic or stochastic.

special XML-based format, models can also be imported and exported in the SBML format.

5.1.3.3 Virtual Cell

The next software package we want to discuss is Virtual Cell, also called VCell, which is developed at the Center for Cell Analysis & Modeling of the University of Connecticut (UConn, www.vcell.org) [22]. Like both of the other tools that we had a look at, the basic purpose of VCell is the numerical simulation of models describing a system of biochemical reactions. However, the main feature that sets it apart from most other simulation tools is its ability also to handle spatially resolved models in 2D and 3D. Because even in the latest version of SBML (Level 3 Version 1) the support for spatial models is limited, VCell models are stored in an XML-based native format called VCML. Nevertheless, SBML models can be imported and nonspatial

models can also be exported as SBML. Normally, VCell models are stored in an online database at UConn and are also executed at UConn servers. Models in the online repository can be browsed and selected via the GUI (Figure 5.5, bottom left panel). However, if desired models can also be exported to the local hard drive and simulated on the local machine (via the “Quick Run” button in the “Simulation” tab).

The GUI is also the place where models are constructed. For this purpose, different views of the model are available. The “Structure Diagram” provides a schematic overview of the compartments and their species. In this view, the graphical shape and location of the components are automatically provided by VCell. Next, details of the reactions, including the kinetic law, parameter values, and the choice between concentrations or particle numbers, are specified in the “Reaction Diagram” (Figure 5.5, red box). In this view, species can be

positioned by the user (inside their compartments), but unlike CellDesigner, VCell offers only a single shape for the icons.

After the so-called “Physiology” of a model has been specified, it is then possible to add “Applications” to the model. Here the decision about the type of simulation is made. It is possible to define deterministic and stochastic simulations that can be run spatially resolved or in a homogeneous environment (compartmental mode). If a spatial model is desired, then it is necessary to define a geometry, which can be either 2D or 3D. The 2D case, however, can only be simulated deterministically. A geometry (e.g., for describing membranes, organelles, or cell shapes) can be constructed from analytical functions (e.g., for circles, cylinders, or ellipsoids) or through a series of data files containing images of cross sections of cells (z-sections). The use of image data is a very powerful and flexible but also a quite complex feature.

A very helpful video tutorial is provided at the VCell website in the User Guide area.

For the purpose of this discussion, we performed a stochastic simulation of a compartmental version of our model, as well as deterministic simulations of a spatial version based on an analytically defined 2D geometry. VCell offers one exact stochastic solver (based on Gibson & Bruck [10]) and three hybrid solvers that partition the system into subsets of fast and slow reactions to deal with models that contain a mixture of species with high and low molecule numbers (Figure 5.6a). After the simulation has completed, the results can be downloaded and viewed graphically (Figure 5.6b). Alternatively, the numerical results can be saved for further analysis by other tools. The simulation of spatial models is much more time consuming and while the simulation runs at the server the local VCell front end can be stopped if necessary.

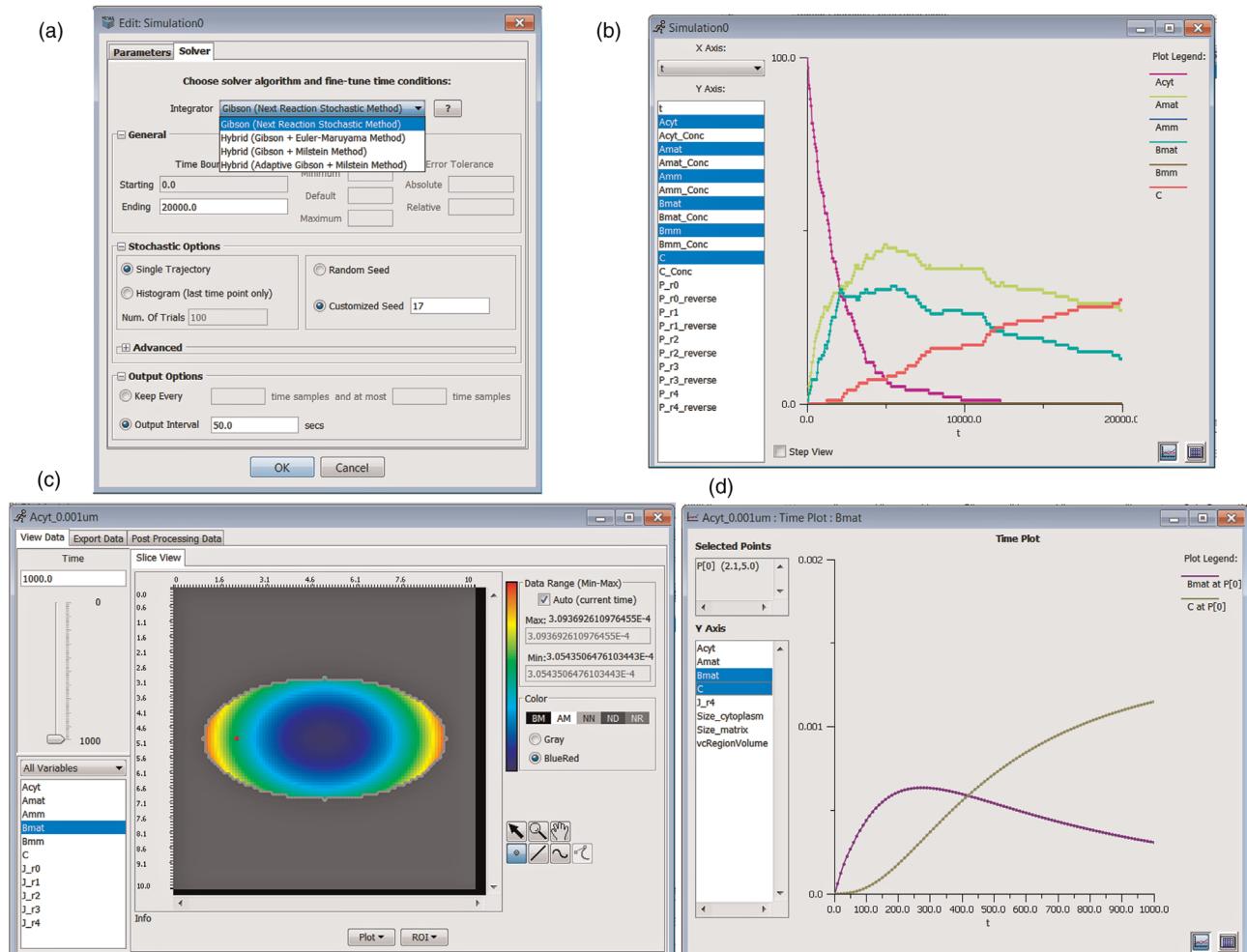


Figure 5.6 Simulation results of a VCell model. (a) Different solvers for deterministic or stochastic simulations can be selected, and panel (b) shows the results of a stochastic simulation of a compartmental model. The results of spatially resolved models are more complex and can be shown as concentration images for a given time point (c) or as a time course for a given point in space (d).

Figure 5.6c shows the results of the deterministic, 2D spatial simulation. At the left-hand side, the time point and variable that should be displayed are chosen, the middle panel shows the color-coded concentrations at each point of our 2D model (for a 3D geometry it would show a slice), and the right-hand side provides some tools for further analysis. Most importantly, the point tool can be used to specify a point (red dot in Figure 5.6c) for which the time course of the species can then be plotted (Figure 5.6d).

In case of a deterministic, compartmental model, an additional type of “Application” is possible and that is parameter estimation. Experimental results can be imported from a data file and then it is possible to select some model parameters that should be fitted to the data. VCell actually uses the COPASI engine for this fitting task, which means that all parameter estimation algorithms available in COPASI are also available in VCell.

5.2 Standards and Formats for Systems Biology

Summary

A crucial part of systems biology is data integration. With the increasing amount of data in modern biology, the requirement of standards used for different data types becomes more and more important. This section gives an overview of different standards and data formats used in systems biology.

As a matter of fact, biological systems are complex and composed of very heterogeneous data. The same applies to the techniques used in modern biology to explore and map the complexity of biological systems. A first step in the analysis of complex and heterogeneous data is data integration. Although data integration by itself does not explain the complex behavior of biological systems, it allows us to increase the information content of different experimental observations and heterogeneous data sets. Technically, data integration requires a conceptual design and the development of common standards.

The development of a standard involves four steps: an informal design of a conceptual model, a formalization, the development of a data exchange format, and the implementation of supporting tools [23]. One of the first experimental platforms in omics research for which a data standard was developed were microarray experiments. A conceptual model about the minimum information that is required for the description of a microarray experiment is specified by MIAME (Minimum Information About a Microarray Experiment [23]).

Similar specifications have, for instance, also been developed for proteomics data with MIAPe (Minimum Information About a Proteomics Experiment [24,25]), or systems biology models with MIRIAM (Minimum Information Requested in the Annotation of Biochemical Models). MIRIAM specifies a set of rules for curating quantitative models of biological systems that define procedures for encoding and annotating models represented in a machine-readable form [26].

The documentation and exchange of models need to be done in a defined way. A very straightforward way to describe and document a model of, for example, a system of biochemical reactions and its associated mathematical model is by the use of a common formalism for the representation of biochemical and mathematical equations. These conventions provide a good standard for the documentation and exchange in publications. Although these formats are very suitable for humans, they cannot be directly processed by a computer. This gave rise to the development of standards for the description of models. During the last 15 years, the eXtensible Markup Language (XML, www.w3.org/XML) has been shown to be a flexible tool for the definition of standard formats. In the following, we give a brief introduction to XML as well as a description of SBML, a standard for model description that is based on XML. Moreover, other standards, such as BioPAX, a standard for the description of cellular reaction systems, and SBGN, a standard for the graphical representation of reaction networks, will be described.

5.2.1 Systems Biology Markup Language

The Systems Biology Markup Language (sbml.org) is a free and open format for the representation of models common to research in many areas of computational biology, including cell signaling pathways, metabolic pathways, gene regulation, and others [27]. It is already supported by many software tools [28]. In July 2015, the SBML Software Matrix listed 281 software systems supporting SBML. Currently, there are three SBML specifications denoted Level 1, Level 2, and Level 3.

SBML is defined as an XML-compliant format. XML documents are written as plain text and have a very clear and simple syntax that can easily be read by both humans and computer programs; however, it is generally intended to be written and read by computers, not humans. In XML, information is associated with tags indicating the type or formatting of the information. Tags are used to delimit and denote parts of the document or to add further information to the document structure. Using miscellaneous start tags (e.g., <tag>) and end tags (e.g., </tag>), information can be structured as text blocks in a hierarchical manner.

Example 5.1

The following example of the phosphorylation reaction of aspartate catalyzed by the aspartate kinase illustrates the general structure of an SBML file.

aspartate + ATP $\xrightarrow{\text{aspartate kinase}}$ aspartyl phosphate + ADP

```

(1) <?xml version="1.0" encoding="UTF-8"?>
(2) <sbml level="2" version="1" xmlns="http://www.sbml.org/sbml/level2">
(3) <model id="AK_reaction">
(4)   <listOfUnitDefinitions>
(5)     <unitDefinition id="mmol">
(6)       <listOfUnits>
(7)         <unit kind="mole" scale="-3"/>
(8)       </listOfUnits>
(9)     </unitDefinition>
(10)    <unitDefinition id="mmol_per_litre_per_sec">
(11)      <listOfUnits>
(12)        <unit kind="mole" scale="-3"/>
(13)        <unit kind="litre" exponent="-1"/>
(14)        <unit kind="second" exponent="-1"/>
(15)      </listOfUnits>
(16)    </unitDefinition>
(17)  </listOfUnitDefinitions>
(18)  <listOfCompartments>
(19)    <compartment id="cell" name="Cell" size="1" units="volume"/>
(20)  </listOfCompartments>
(21)  <listOfSpecies>
(22)    <species id="asp" name="Aspartate" compartment="cell" initialConcentration="2"
substanceUnits="mmol"/>
(23)    <species id="aspp" name="Aspartyl phosphate" compartment="cell" initialConcentration="0"
substanceUnits="mmol"/>
(24)    <species id="atp" name="ATP" compartment="cell" initialConcentration="0"
substanceUnits="mmol"/>
(25)    <species id="adp" name="ADP" compartment="cell" initialConcentration="0"
substanceUnits="mmol"/>
(26)  </listOfSpecies>
(27)  <listOfReactions>
(28)    <reaction id="AK" reversible="false">
(29)      <listOfReactants>
(30)        <speciesReference species="asp" stoichiometry="1"/>
(31)        <speciesReference species="atp" stoichiometry="1"/>
(32)      </listOfReactants>
(33)      <listOfProducts>
(34)        <speciesReference species="aspp" stoichiometry="1"/>
(35)        <speciesReference species="adp" stoichiometry="1"/>
(36)      </listOfProducts>
(37)      <kineticLaw>
(38)        <math xmlns="http://www.w3.org/1998/Math/MathML">
(39)          <apply>
(40)            <times/>
(41)            <ci>k</ci>
(42)            <ci>asp</ci>
(43)            <ci>atp</ci>
(44)            <ci>cell</ci><ci>cell</ci>
(45)          </apply>
(46)        </math>
(47)      <listOfParameters>
(48)        <parameter id="k" value="2.25" units="per_mM_and_min"/>
(49)      </listOfParameters>
(50)    </kineticLaw>
(51)  </reaction>
(52) </listOfReactions>
(53) </model>
(54) </sbml>
```

Line 1 in Example 5.1 defines the document as an XML document. The SBML model is coded in lines 2–54. It is structured into several lists that define different properties of the model. The most important lists that are frequently used are the definition of units (lines 4–17), of compartments (lines 18–20), of species (lines 21–26), and finally of the reactions themselves (lines 27–52). Most entries in SBML have one required attribute, `id`, to give the instance a unique identifier by which other parts of the SBML model definition can refer to it. Some base units, such as gram, meter, liter, mole, and second, are already predefined in SBML. More complex units derived from the base units are defined in the list of units. For instance, mM s^{-1} that is equal to $\text{mmol l}^{-1} \text{s}^{-1}$ can be defined as shown in lines 10–16 and used by its `id` in the subsequent definition of parameters and initial concentrations. Compartments are used in SBML as a construct for the grouping of model species. They are defined in the list of compartments (lines 18–20) and can be used not only for the definition of cellular compartments but also for grouping in general. Each compartment can have a `name` attribute and defines a compartment size. Model species are defined in the list of species. Each species has a unique “`id`” attribute that can be used to refer to it and a `name` and initial value with its respective unit. Species identifiers are used in the list of reactions (lines 27–52) for the definition of the individual biochemical reactions. Reversibility of a reaction is indicated by an attribute of the reaction tag (line 28). Reactants and products of a specific reaction along with their respective stoichiometry are specified in separate lists (lines 29–36).

The kinetic law of an individual reaction (lines 37–50) is specified in MathML. MathML is an XML-based markup language especially created for the representation of complicated mathematical expressions. In the above example, the rate law reads $k \cdot [\text{asp}] \cdot [\text{atp}] \cdot \text{cell}^2$, where k is a kinetic parameter, `[asp]` and `[atp]` are the concentrations of aspartate and ATP, respectively, and `cell` is the volume of the cell. The consideration of the cell volume is needed, since rate laws in SBML are expressed in terms of amount of substance abundance per time instead of the traditional expression in terms of amount of substance concentration per time. The formulation of the rate law in the traditional way embodies the tacit assumption that the participating reaction species are located in the same, constant volume. This is done because attempting to describe reactions between species located in different compartments that differ in volume by the expression in terms of concentration per time quickly leads to difficulties.

5.2.2

BioPAX

Another standard format that is used in systems biology and designed for handling information on pathways and topologies of biochemical reaction networks is BioPAX (www.biopax.org) [29]. While SBML is tuned toward the simulation of models of molecular pathways, BioPAX is a more general and expressive format for the description of biological reaction systems even if it is lacking definitions for the representation of dynamic data such as kinetic laws and parameters. BioPAX is defined by the BioPAX working group (www.biopax.org/). The BioPAX Ontology defines a large set of classes for the description of pathways, interactions, and biological entities as well as their relations. Reaction networks described by BioPAX can be represented by the use of XML. Many systems biology tools and databases make use of BioPAX for the exchange of data.

5.2.3

Systems Biology Graphical Notation

Graphical representations of reaction networks prove as very helpful tools for the work in systems biology. The graphical representation of a reaction system is helpful not only during the design of a new model and as a representation of the model topology, but also for the analysis and interpretation, for instance, of simulation results. Traditionally, diagrams of interacting enzymes and compounds have been written in an informal manner of simple unconstrained shapes and arrows. Several diagrammatic notations have been proposed for the graphical representation [30–33]. As a consequence of the different proposals, SBGN has been set up [34]. It provides a common graphical notation for the representation of biochemical and cellular reaction networks. SBGN defines a comprehensive set of symbols, with precise semantics, together with detailed syntactic rules defining their usage. Furthermore, SBGN defines how such graphical information is represented in a machine-readable form to ensure its proper storage, exchange, and reproduction of the graphical representation.

SBGN defines three different diagram types: (i) state transition diagrams that depict all molecular interactions taking place, (ii) activity flow diagrams that represent only the flux of information going from one entity to another, and (iii) entity relationship diagrams that represent the relationships between different molecular species. In a state transition diagram, each node represents a given state of a species, and therefore a given species may appear multiple

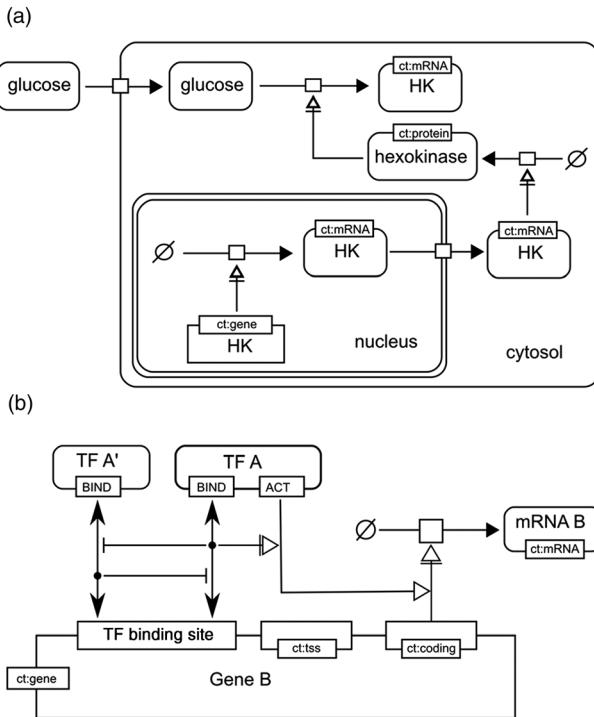


Figure 5.7 Systems Biology Graphical Notation (SBGN). (a) State transition diagram. (b) Entity relation diagram describing gene regulation and transcription of a gene. The two transcription factors TF A and TFA' compete for the same transcription factor-binding site. If one of the transcription factors is bound, the binding site is blocked for the other one, but only TF A can activate the transcription of the gene. The abbreviation “ct” indicates conceptual types of the respective entity.

times. State transition diagrams are suitable for following the temporal process of interactions. A drawback of state transition diagrams, however, is that the representation of each individual state of a species results quickly in very large diagrams, and due to this, it becomes difficult to understand what interactions actually exist for the species in question. In such a case, an entity relation diagram is more suitable. In an entity relation diagram, a biological entity appears only once. Examples of a state transition and an entity relationship diagram are given in Figure 5.7.

SBGN defines several kinds of symbols, whereas two types of symbols are distinguished: nodes and arcs. There are different kinds of nodes defined. Reacting state or entity nodes represent, for example, macromolecules, such as protein, RNA, DNA, polysaccharide, or simple chemicals, such as a radical, an ion, or a small molecule. Container nodes are defined for the representation of a complex, compartment, or module. Different transition nodes are defined for the

representation of transitions such as biochemical reactions, associations like protein complex formation, or dissociations such as the dissociation of a protein complex. The influence of a node on another is visualized by different types of arcs representing, for example, consumption, production, modulation, stimulation, catalysis, inhibition, or trigger effect. Not all node and arc symbols are defined for each of the three diagram types. A detailed description of the different nodes and arcs, and the syntax of their usage by the different diagram types is given in the specification of SBGN (see sbgn.org/).

5.3 Data Resources for Modeling of Cellular Reaction Systems

Summary

Essential for the development of models in systems biology is comprehensive information about the biological system of interest. Nowadays, a lot of information is collected in databases and data repositories. This section gives an overview of a selection of different data and information resources used in systems biology for the development of computational models.

The development of models of biological systems requires diverse types of data. This is, for instance, information about the different model components (e.g., metabolites, proteins, and genes) and their different functions and interactions. Such information can be extracted from the literature or dedicated data resources, such as pathway databases. Two pathway databases that are well known are KEGG (Kyoto Encyclopedia of Genes and Genomes) and Reactome. Both are described below in more detail. Further information about databases providing primary data is given in Chapter 16.

5.3.1 General-Purpose Databases

In the following, we introduce some databases and resources that are frequently used in systems biology for data mining beginning with two data resources of more general use, the websites PathGuide and BioNumbers.

5.3.1.1 PathGuide

Pathway-related data and information are of major importance for systems biology. PathGuide is a

pathway resource list giving an overview of web-accessible biological pathway and network databases [35]. PathGuide (www.pathguide.org) is currently listing 547 resources providing information about biological pathways and molecular interactions. These include databases on protein interactions, metabolic and signaling pathways, transcription factors and gene regulatory networks, protein–compound interactions, and pathway diagrams. PathGuide also gives details on the availability and supported exchange standards of the respective data repositories.

5.3.1.2 BioNumbers

For biological properties, numerical values are sometimes difficult to find in the literature. Most quantitative properties in biology depend on the context or the method of measurement, the organism, or the cell type. Often, however, the order of magnitude is already a very useful information for modeling. BioNumbers (www.bionumbers.hms.harvard.edu) is a database of useful biological numbers [36]. It allows you to easily browse or search for many common biological numbers that might be difficult to find but can be very important for modeling, such as the rate of translation of the ribosome or the number of bacteria in the gut. BioNumbers is a community effort to make quantitative properties of biological systems easily available together with full references.

5.3.2

Pathway Databases

The development of models of biochemical reaction networks requires information about the stoichiometry and topology of the reaction network. Such information can be found in databases such as KEGG and Reactome. Often pathway databases cover a specific scope, for example, metabolic pathways, signal transduction pathways, or gene regulatory networks. Some databases act as metadatabases that integrate pathway data from multiple sources building up a comprehensive resource of pathway and interaction data such as ConsensusPathDB.

5.3.2.1 KEGG

KEGG (<http://www.genome.ad.jp/kegg/>) is a reference knowledge base offering information about genes and proteins, biochemical compounds and reactions, as well as pathways [37]. The data are organized in three parts: the gene universe (consisting of the GENES, SSDB, and KO databases), the chemical universe (with the COMPOUND, GLYCAN, REACTION, and

ENZYME databases that are merged as LIGAND database), and the protein network consisting of the PATHWAY database [38]. Besides this, the KEGG database is hierarchically classified into categories and subcategories at four levels. The five topmost categories are metabolism, genetic information processing, environmental information processing, cellular processes, and human diseases. Subcategories of metabolism are, for example, carbohydrate, energy, lipid, nucleotide, or amino acid metabolism. These are subdivided into the different pathways, such as glycolysis, citrate cycle, and purine metabolism. Finally, the fourth level corresponds to the KO (KEGG Ontology) entries. A KO entry (internally identified by a K number, e.g., K00001 for alcohol dehydrogenase) corresponds to a group of orthologous genes that have identical functions.

The gene universe offers information about genes and proteins generated by genome sequencing projects. Information about individual genes is stored in the GENES database, which is semiautomatically generated from the submissions to GenBank, the NCBI RefSeq database, the EMBL database, and other publicly available organism-specific databases. K numbers are further assigned to entries of the GENES database. The SSDB database contains information about amino acid sequence similarities between protein-coding genes computationally generated from the GENES database. This is carried out for many complete genomes and results in a huge graph depicting protein similarities with clusters of orthologous and paralogous genes.

The chemical universe offers information about chemical compounds and reactions relevant to cellular processes. It includes more than 17 000 compounds (internally represented by C numbers, e.g., C00001 denotes water), a separate database for carbohydrates (nearly 11 000 entries; represented by a number preceded by G, e.g., G10481 for cellulose), more than 9800 reactions (with R numbers, e.g., R00275 for the reaction of the superoxide radical into hydrogen peroxide), and more than 6500 enzymes (denoted by EC numbers as well as K numbers for orthologous entries). All these data are merged as LIGAND database [39]. Thus, the chemical universe offers comprehensive information about metabolites with their respective chemical structures and biochemical reactions.

KEGG's protein network provides information about protein interactions comprising pathways and protein complexes. The 474 KEGG reference pathway diagrams (maps), offered on the website, give clear overviews of important pathways. Organism-specific pathway maps are automatically generated by coloring of organism-specific genes in the reference pathways.

5.3.2.2 Reactome

Reactome is an open, online database of fundamental human biological processes (www.reactome.org). The Reactome project is managed as a collaboration of the Cold Spring Harbor Laboratory, the European Bioinformatics Institute (EBI), and the Gene Ontology Consortium [40,41]. The database is divided into several modules of fundamental biological processes that are thought to operate in humans. Each module of the database has one or more primary authors and is further peer reviewed by experts of the specific field. Each module can also be referenced by its revision date and thus can be cited like a publication.

The current version of Reactome (version 53) describes 8128 human proteins participating in 8369 reactions. Besides the description of molecular species, reactions, and pathways, the database also annotates a broad range of major disease processes at the molecular level, such as mutations leading to the loss or gain of function of the gene product, or infectious agents such as a virus that introduces a novel gene product with a new reaction perturbing normal human processes.

On the one hand, the Reactome database is intended to offer valuable information for the wet-lab scientist, who wants to know, for example, more about a specific gene product she or he is unfamiliar with. On the other hand, the Reactome database can be used by the computational biologist to draw conclusions from large data sets such as gene or protein expression data.

The Reactome website offers several tools to analyze data, such as a tool for pathway overrepresentation analysis or a function to compare human processes with the processes of another species showing the overlap of the interaction network. Data from Reactome can be exported in various formats, such as SBML and BioPAX.

5.3.2.3 WikiPathways

Compared with KEGG and Reactome, WikiPathways (www.wikipathways.org) is a public wiki for pathway curation [42]. WikiPathways serves as a repository for biological knowledge by the use of pathway diagrams. Entities such as genes, proteins, or metabolites can be annotated with many different identifier systems, such as Ensembl or ChEBI, and the entries can be linked to other external databases or information resources, such as genome browsers, experimental platforms, Gene Ontology (GO), or Wikipedia.

5.3.2.4 ConsensusPathDB

ConsensusPathDB (consensuspathdb.org) is a database integrating human functional interactions [43–45]. Currently, the database integrates the content of 32 different

interaction databases with heterogeneous foci comprising a total of about 155 000 distinct physical entities and about 435 000 distinct functional interactions covering nearly 4400 pathways. The database comprises protein–protein interactions, biochemical reactions, and gene regulatory interactions. ConsensusPathDB has a sophisticated interface for the visualization of the functional interaction networks. Furthermore, ConsensusPathDB offers two statistical approaches to analyze lists of genes or metabolites, for example, as coming from omics analysis. The first approach is overrepresentation analysis, where sets of predefined functionally associated components, such as pathways or GO categories, are tested for overrepresentation in the user-specified list based on the hypergeometric test. The second approach is called enrichment analysis that uses the Wilcoxon signed-rank test taking as an input genes with exactly two measurement values, typically expression in two different phenotypes or conditions.

5.3.3 Model Databases

Due to the advent of systems biology, mathematical models are frequently available following common standards, such as SBML, or are carefully reimplemented in a computer-readable format according to published descriptions. During the last few years, huge efforts have been made on the gathering and implementation of existing models in databases. Two well-known databases on this are BioModels and JWS, which are described in the following.

5.3.3.1 BioModels

The BioModels project is an international effort to (i) define agreed-upon standards for model curation, (ii) define agreed-upon vocabularies for annotating models with connections to biological data resources, and (iii) provide a free, centralized, publicly accessible database of annotated, computational models in SBML, and other structured formats [38]. The 29th release of the BioModels Database (biomodels.org) provides access to more than 144 000 models, of which a large number of models are automatically generated from pathway resources. 1296 models correspond to models published in the literature, of which 575 models have been manually curated. Models can be browsed in the web interface, online simulations can be performed via the external simulation engine of JWS Online (see below), or they can be exported in several prominent file formats (e.g., SBML, CellML, and BioPAX) for external usage by other programs.

5.3.3.2 JWS Online

Another model repository that is providing kinetic models of biochemical systems is JWS Online [46]. As of July 2015, this model repository provides more than 190 models (jjj.biochem.sun.ac.za). Models in JWS Online can be interactively run and interrogated over the Internet.

5.4 Sustainable Modeling and Model Semantics

Summary

The computer-assisted processing of models is greatly facilitated by practices of sustainable model building, including the use of standard formats, a clear documentation, a central and open storage of models, and semantic annotations that describe the biological meaning of model elements. An infrastructure including model formats, ontologies, and public data and model repositories has been established by the community. The extra effort for model annotation pays off: it enables an intuitive understanding of the model at hand and allows us to match models element by element with respect to their biological interpretation, to cluster them by similarity, and to use software tools for semiautomatic model merging.

Over the years, the increasing amounts of biological high-throughput data have allowed systems biology models to become larger, more complex, and more precise, covering a variety of cellular pathways. Aside from comprehensive, organism-specific cell models [47], the future will bring us various tailored models of differing scopes (e.g., in the resolution chosen) and foci (e.g., on different pathways). These models can then be flexibly adjusted and combined to address specific questions, and be used for different applications. Making models modifiable and recombinable by software poses challenges, including model versioning, a central storage, and sophisticated methods for model retrieval, alignment, and combination, as well as automated validity checks. An important technical prerequisite are semantic annotations in the models, which we shall discuss in this chapter. However, modelers' awareness and foresight are just as key to sustainable model building as technology is. Modelers should anticipate that their models may be reused, and adopt techniques of sustainable modeling when developing models, designing experiments, and collaborating within research projects [48].

5.4.1

Standards for Systems Biology Models

Computer-assisted modeling relies crucially on standard data formats. While shifting from paper-and-pencil models to computer-assisted modeling, systems biologists have established standards, such as SBML, and software that supports these standards, such as CellDesigner [49] or COPASI [50]. The success of SBML has triggered developments such as the MIRIAM guidelines for published models [51] and SBGN [52], now the standard for depicting biochemical network models. These standards are further developed by the COMBINE initiative ([co.mbine.org/](http://combine.org/)), an open community of developers and users who devise interoperable standards for modeling in biology, centered around the core standards SBML, SBGN, SED-ML (Simulation Experiment Description Markup Language) [53], BioPAX, CellML, and SBOL (Synthetic Biology Open Language). A set of conventions for structured data tables – including models stored in spreadsheet files – can be found at www.sbtab.net. Generally, the development of a technical infrastructure concerns three main areas: definitions (guidelines, ontologies, identifiers, and data schemas), representations (formats or languages for encoding data), and access to data and software services (through standardized interfaces) [54]. For instance, the formal description of simulation experiments is based on good practice guidelines (MIASE: Minimal Information About a Simulation Experiment) [55] and implemented in the XML data format SED-ML, which in turn relies on terms and relationships defined in a specific ontology (KiSAO: Kinetic Simulation Algorithm Ontology). “Minimal requirement” guidelines for different fields of application are made available through the MIBBI (Minimum Information for Biological and Biomedical Investigations) consortium [56].

Thanks to standardization efforts, a large number of models are now stored in the public repository BioModels Database [57], which includes more than 500 manually curated models and more than 100 000 network models adopted from other sources (e.g., the KEGG Pathways [58]) and translated into SBML.

5.4.2

Model Semantics and Model Comparison

Semantic annotations describe the meaning of models in a computer-processable way. Using these annotations, modeling tools can do more than just numerical integration: they can generate selective visualizations, check and compare models, or assist users in model merging. In principle, programs could even use models as sources of

biochemical information and perform automatic reasoning about their contents. As a prerequisite, models must contain explicit statements about the biological meaning of their components, for example, state that a certain variable represents the concentration of a specific substance. One of the basic tasks in model processing is to decide whether variables from different models (for instance, “glucose” and “GLC”) refer to the same entity. If models stem from different sources and use different naming conventions, semantic annotations must be consulted. Semantic annotations link the models to controlled biological vocabularies or ontologies [59]. In computer science, ontologies define a set of basic notions and relations relevant to a certain field of knowledge. A prominent example is the Gene Ontology, which provides terms and hierarchical relationships to describe the roles of gene products in cells. The development of interoperable ontologies is supported by the OBO (Open Biomedical Ontologies) effort, and ontologies can be accessed through the bioportal [60] or tools such as the EBI Ontology Lookup Service (OLS).

5.4.2.1 Semantics Annotations in SBML

SBML represents models neither as pure biochemical networks nor as pure mathematical equation systems, but as a mixture of both. Some semantic information is hard-coded in the SBML syntax (e.g., the links between reactions and their reactants) or expressed by XML attributes (e.g., the fact that a substance is represented by its concentration and not by an amount). Other semantic information, for example, the fact that a model parameter represents a Michaelis–Menten value, is given by terms from the Systems Biology Ontology [59]. The biological meaning of elements (e.g., which substance is represented by some `<species>` element) is specified by MIRIAM-compliant RDF annotations [61] that point to entries in public web resources (e.g., the ChEBI database for biochemical compounds [62]). Resolvable persistent identifiers can be obtained by using the [Identifiers.org](#) system [63], which is based on the unique and perennial identifiers provided by the MIRIAM Registry. Aside from stating an exact identity between model element and database element, annotations can also establish more complex logical relations, for example, state that the model element “D-glucose” is a type of glucose. A mechanism for this is provided by Biomodels.net qualifiers (in this case, the qualifier `IsVersionOf`). Annotations of this type enable software tools to resolve standardized identifiers (e.g., associate identifiers with readable names and additional information) and use ontologies to connect biochemical concepts, for example, to infer whether a substance is a subtype of another one.

5.4.2.2 Element Similarities

Comparing models on the basis of their biological elements can be complicated. For instance, should a variable annotated with “glucose” and one annotated with “D-glucose” – the form of glucose typically used by cells – be treated as equivalent or not during model combination? This distinction may be unintended and can arise if models (or experimental data to be mapped onto models) are annotated by different curators. Even if a software cannot decide on this, it should be able to spot the problem and warn the user.

To recognize that two elements are almost identical, modeling tools must relate the elements by an ontology. Similarities between model elements can be quantified by similarity scores ranging between 0 (for very different elements) and 1 (for identical elements). These scores can, for instance, be defined based on distances in the ontology in which the elements appear [64] (see Figure 5.8b). More complex scores can also account for the ontology structure as a whole [65]. Qualifiers in the annotations (e.g., `IsVersionOf`) can be reflected by reduced similarity scores. An example of a software that can compute element similarities is semanticSBML.

Figure 5.9 shows a large set of models and the biological entities appearing in them in the form of a matrix. Each row (“semantic annotation vector”) characterizes one model, stating whether each of the semantic elements appears in the model (1) or not (0). Annotation vectors allow for simple semantic comparisons between models: a scalar product between two vectors yields the number of shared semantic elements; the cosine of the angle between the vectors expresses this overlap on a scale between 0 and 1. After the similarities between biochemical entities have been quantified, the matrix of similarity scores (e.g., with a large entry between glucose and D-glucose) can be used to define a non-Euclidean metric in the space of annotation vectors. With this new metric, vector elements for glucose and D-glucose will have practically the same effects when scalar products are computed [64].

5.4.2.3 Model Alignment and Model Similarities

Semantic comparisons can be used to align models, to cluster them by similarity, or to automatically retrieve models that resemble or overlap with a given query model. Various similarity scores, both for models and for individual model elements, have been proposed [64] because what aspects should be scored by a similarity measure and how they should be weighted depend on the purpose of the comparison. If the purpose of comparison is model combination, models should be counted as

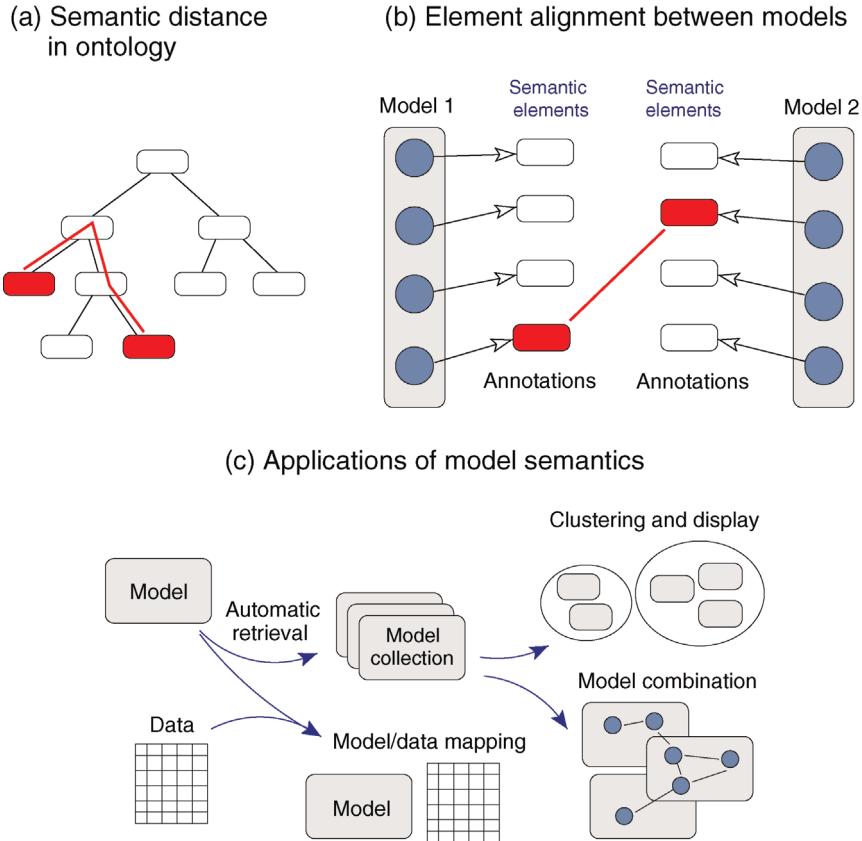


Figure 5.8 Semantic annotations. (a) The semantic distance between biological elements can be defined by the shortest weighted path in an ontology. (b) To define similarities between model elements, we look up the biological concepts referenced in the annotations (formal identifiers describing substances, reactions, or other elements), connect them by paths in an ontology, and compute a similarity score from the semantic relationships along these paths. Identity (or, more generally, similarity) between semantic elements enables us to match model elements. (c) Possible applications of model semantics. (Redrawn from Ref. [64].)

similar if they share biochemical elements. Aspects that can be disregarded in this case, for example, mathematical formulation or numerical parameter values, may become relevant in other contexts.

Similarity scores can be classified according to the information they draw upon. Some scores disregard model structure and consider only the semantic elements referenced (“bag of annotations”). Others compare model elements individually and align them by annotation similarities (see Figure 5.8). As an example, Figure 5.10 shows two MAP kinase cascade models of different resolution: the automatic alignment makes the correspondence between the models easy to see. Finally, in more sophisticated scores, elements lacking annotations are compared through annotations of their neighboring elements by a mechanism called “semantic propagation”: reactions, for instance, may be compared based on annotations of their reactants. Elaborate similarity scores, however, are not needed for all applications. A simple model comparison by feature vectors, which only considers which

biochemical elements are common to both models, yields already good results.

5.4.3 Model Combination

A promising way to construct large, possibly whole-cell, models is to combine existing pathway models [69–73]. Conceptually, model combination relies on the (reductionist) assumption that a model’s correctness or incorrectness does not depend on context, that is, on the environment in which the biological system resides. Technically, successful model combination depends on well-formed, well-described input models. Model combination is thus the moment where sustainable model building pays off [48]. Joining two models can be difficult if they are based on different mathematical formalisms, for example, FBA and kinetic models. However, even models based on the same formalism can be hard to combine because they may overlap and contradict each

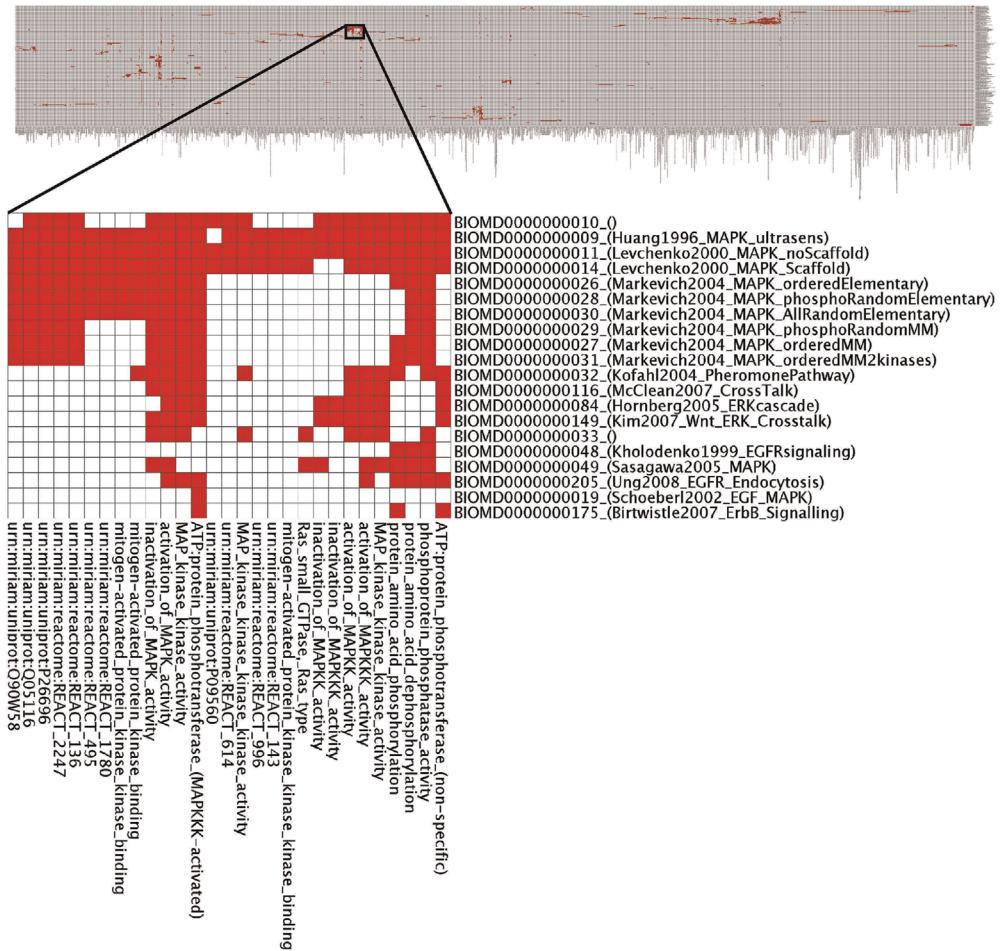


Figure 5.9 Annotations in systems biology models in BioModels Database. An annotation matrix (top) shows which biological concepts (columns) appear in which models (rows). After sorting the matrix by hierarchical two-way clustering, similar concepts or models appear close to each other. The close-up shows a selection of MAP kinase models and the concepts typically associated with them. (From Ref. [64].)

other; in such cases, we first need to bring both model structures into agreement [73].

There are several types of model combination: in model composition, models remain intact and are coupled through communicating variables (see Section 6.4); in model aggregation, the submodels are deliberately coupled by defined interfaces, while their internal variables are shielded [74]; in model fusion or merging, a new “flat” model is built from elements of the original models. However, the difficulties mentioned above, for example, the problem of checking the input models for redundant or conflicting elements, may occur in all cases.

A relatively simple basic procedure of model merging is as follows [72]. Formally, a biochemical network model can be seen as a list of compounds and reactions, each with a number of properties (stoichiometries and rate laws, for instance, can be listed as properties of the reactions). To merge two or more

models, we can join their element lists, find duplicate elements, replace the duplicates by single elements, and translate the resulting, nonredundant list into a syntactically valid model. This procedure entails the following subtasks: (i) Model elements (variables, parameters, and chemical reactions) must be compared by their biological meaning as shown in Figure 5.11; as we saw before, this requires a clear description by computer-readable annotations. (ii) Measurement units must be compared and unified. (iii) Explicit conflicts between models, for example, different rate laws assumed for the same reaction, must be detected and resolved. (iv) Implicit conflicts, which may arise if the input models make contradicting assumptions or obey contradicting constraints (e.g., thermodynamic relationships between rate constants), must be resolved. This can be done by the user or automatically based on predefined rules. (v) If the original model parameters had been obtained by

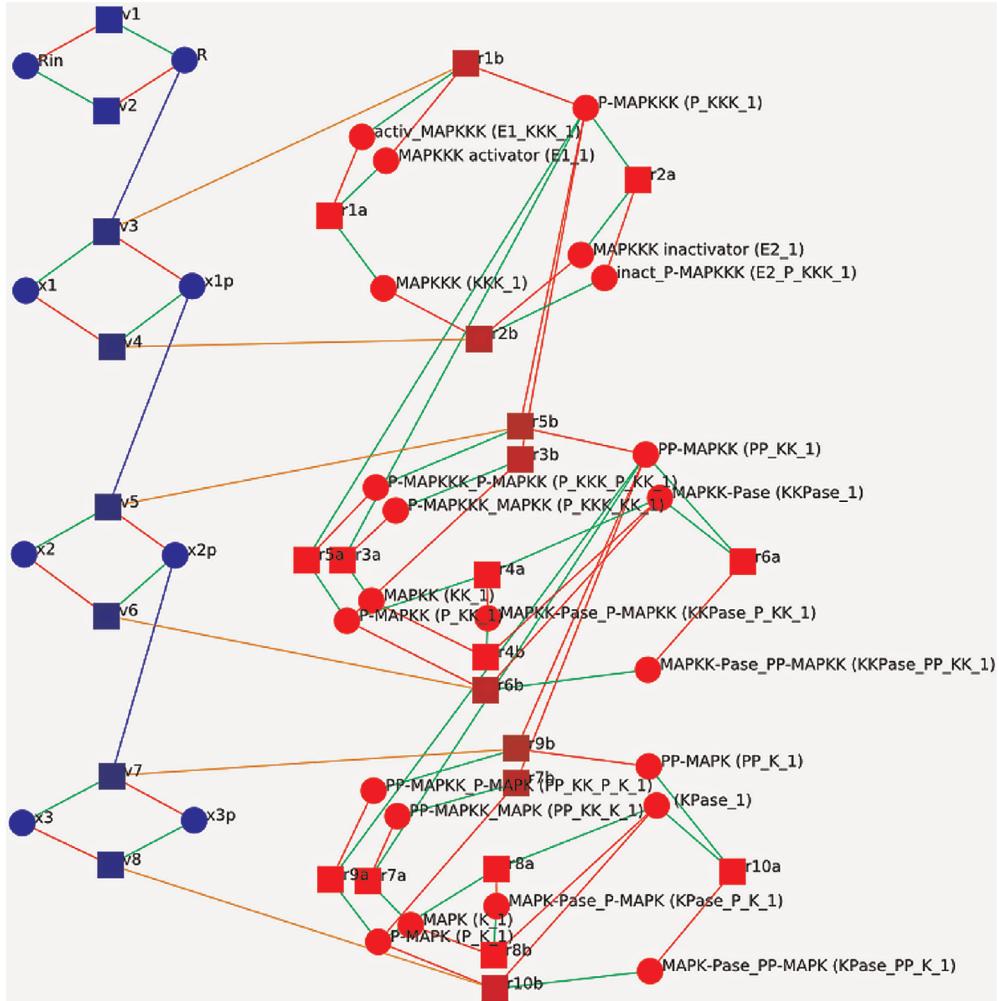


Figure 5.10 In mitogen-activated kinase (MAPK) cascades, kinases transmit signals by phosphorylating each other subsequently. To align two MAP kinase cascade models from Refs. [67] (BioModel 84; left, in blue) and [68] (BioModel 9; right, in red), model elements were automatically matched by semanticSBML. Network nodes represent species (circles) and reactions (squares). Corresponding model elements are linked by orange lines. (From Ref. [67].)

fitting, the parameters in the merged model may have to be refitted.

This procedure tends to become laborious for modelers, so automation can be helpful. However, if redundant elements come with conflicting mathematical statements (e.g., two concentration values of the same substance), a software may not be able to resolve these conflicts automatically (see Figure 5.11). Apart from difficulties in correctly matching the elements, there may be conflicts in model assumptions, or conflicts between different levels of resolution in the models. Finally, changes in the model structure during merging may cause new conflicts: for instance, a combination of metabolic pathways can lead to new thermodynamic loops and therefore to new Wegscheider conditions to be satisfied (see Section 15.6). Such conflicts cannot be resolved by a simple element-by-element merging.

5.4.4 Model Validity

When models are processed by software, they can be routinely checked for validity. Formal concepts for model validity, implemented in standard formats, help modelers avoid mistakes and can capture knowledge and intuition about the mathematical, physical, and biochemical aspects of models. What do we mean by model validity? Models cannot depict reality in all details and are, in this sense, necessarily incomplete. In Section 6.2, we will discuss how plausible models can be chosen based on data. However, there are also general validity criteria that should always hold (e.g., that equations be mathematically valid and completely given) or that in many cases may be required by the user (e.g., mass conservation in chemical reactions). What

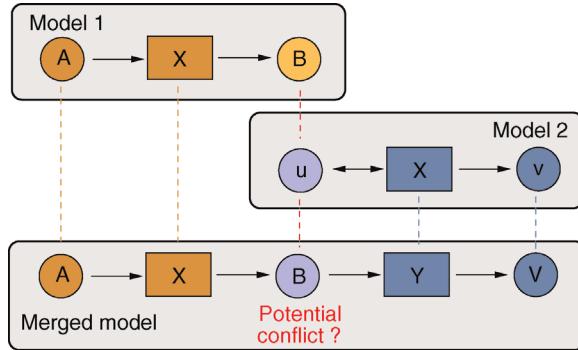


Figure 5.11 Model merging. Two models (top and center) are merged into one (bottom). Circles and boxes represent substances and reactions, respectively. Model elements are aligned (dashed lines) according to their biological meaning as indicated by annotations. A comparison by names would be unreliable because naming conventions can vary across models. When merging the models, duplicate elements (here: B and u) must be collapsed to avoid redundancies. If these elements have different features (e.g., concentrations for a compound or rate laws for a reaction), the conflicts need to be resolved manually or according to default rules.

validity criteria apply depends on the model's purpose and on the modeler's expectations: thermodynamic correctness, for instance, may be important in metabolic models, but irrelevant in signal transduction models where energy conversion is not in the focus of interest. We can conclude that there is not one absolute criterion for model validity, but many possible criteria, which may or may not be relevant depending on the circumstances [72]. General types of validity criteria, relying on different types of information in the models and used to define different sorts of conflicts, are listed in Table 5.1. If enough information is provided in a computer-readable form, such validity criteria can be tested by software.

Table 5.1 Types of validity criteria for systems biology models.

Criteria	Test requires	Example conflict
Syntax	SBML syntax rules	References to undefined model elements
Computation	Equation system encoded in model	Equations underdetermined
Semantics	Semantic annotations and ontologies	Negative Michaelis–Menten constants
Physics	Checks for model equations	Mass balances violated
Biochemistry	Data about realistic values	Unrealistically high concentrations
Facts about organism	Databases/network reconstructions	Genes not present in organism

Explicit validity criteria can help us develop tests for meaningful models and devise safe algorithms for model merging. For a detailed description, see Ref. [72].

Thinking about validity criteria is also an occasion to rethink fundamental concepts in modeling, such as model assumptions, context independence, and the fact that the same reality can be formally described in multiple ways. Moreover, model validity is also of practical relevance, for example, in computer-assisted model combination: valid models, when combined, should again result in a valid model. Formal validity criteria can be useful in model merging. First, sanity checks can be applied during or after merging: in case of conflicts, the software can warn the user and suggest ways to resolve the conflicts. Second, merging procedures can be designed such that certain validity criteria are preserved by construction [72]. Third, appropriate model formulations can avoid conflicts a priori, thus preventing conflicts from even arising. Here are two examples: (i) Many potential conflicts between model elements can be avoided if the input models are all based on one common network scheme (with a fixed resolution and defined names and semantic annotations). (ii) Some physical laws can be directly implemented in the model formulation. For instance, the equilibrium constants in metabolic models cannot be freely chosen, but depend on each other via Wegscheider conditions (see Section 6.1). If the same model is parameterized by Gibbs energies of formation, the resulting equilibrium constants will automatically satisfy all constraints, even after parameter changes or model combination. Finally, validity criteria and safe model formulations can inform the development of data formats such as SBML, helping modelers to create complete and meaningful models.

References

- 1 de Jong, H. (2002) Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.*, 9(1), 67–103.
- 2 Kitano, H. (2002) Computational systems biology. *Nature*, 420 (6912), 206–210.
- 3 Hairer, E., Norsett, S., and Wanner, G. (1993) *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer, Berlin.
- 4 Hindmarsh, A.C. (1983) ODEPACK, a systematized collection of ODE solvers, in *Scientific Computing* (eds. R.S. Stepleman *et al.*), North-Holland, Amsterdam, pp. 55–64.
- 5 Petzold, L. (1983) Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM J. Sci. Stat. Comput.*, 4, 136–148.
- 6 Cohen, S.D. and Hindmarsh, A.C. (1996) CVODE, a stiff/nonstiff ODE solver in *C. Comput. Phys.*, 10(2), 138–143.
- 7 Deuflhard, P. and Nowak, U. (1987) Extrapolation integrators for quasilinear implicit ODEs, in *Large Scale Scientific Computing* (eds P. Deuflhard and B. Engquist), Birkhäuser, pp. 37–50.
- 8 Gillespie, D.T. (1992) A rigorous derivation of the chemical master equation. *Physica A*, 188, 404–425.
- 9 Gillespie, D.T. (1977) Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81, 2340–2361.

- 10** Gibson, M.A. and Bruck, J. (2000) Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem.*, 104, 1876–1889.
- 11** Gillespie, D.T. (2001) Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 115, 1716–1733.
- 12** Puchalka, J. and Kierzek, A.M. (2004) Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks. *Biophys. J.*, 86, 1357–1372.
- 13** Klipp, E., Liebermeister, W., Helbig, A., Kowald, A., and Schaber, J. (2007) Systems biology standards – the community speaks. *Nat. Biotechnol.*, 25(4), 390–391.
- 14** Materi, W. and Wishart, D.S. (2007) Computational systems biology in drug discovery and development: methods and applications. *Drug Discov. Today*, 12 (7–8), 295–303.
- 15** Wierling, C., Herwig, R., and Lehrach, H. (2007) Resources, standards and tools for systems biology. *Brief. Funct. Genomics Proteomics*, 6(3), 240–251.
- 16** Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K.Y., and Kitano, H. (2011) Software for systems biology: from tools to integrated platforms. *Nat. Rev. Genet.*, 12(12), 821–832.
- 17** Klipp, E., Liebermeister, W., Helbig, A., Kowald, A., and Schaber, J. (2007) Systems biology standards – the community speaks. *Nat. Biotechnol.*, 25(4), 390–391.
- 18** Funahashi, A., Tanimura, N., Morohashi, M., and Kitano, H. (2003) CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO*, 1, 159–162.
- 19** Drager, A., Hassis, N., Supper, J., Schroder, A., and Zell, A. (2008) SBMLSqueezer: a CellDesigner plug-in to generate kinetic rate equations for biochemical networks. *BMC Syst. Biol.*, 2, 39.
- 20** Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N. *et al.* (2006) COPASI – a COmplex PAthway SImlator. *Bioinformatics*, 22(24), 3067–3074.
- 21** Schuster, S., Dandekar, T., and Fell, D.A. (1999) Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. *Trends Biotechnol.*, 17 (2), 53–60.
- 22** Cowan, A.E., Moraru, I.I., Schaff, J.C., Slepchenko, B.M., and Loew, L.M. (2012) Spatial modeling of cell signaling networks. *Methods Cell Biol.*, 110, 195–221.
- 23** Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C. *et al.* (2001) Minimum Information About a Microarray Experiment (MIAME) – toward standards for microarray data. *Nat. Genet.*, 29(4), 365–371.
- 24** Taylor, C.F., Paton, N.W., Lilley, K.S., Binz, P.-A., Julian, R.K., Jones, A.R. *et al.* (2007) The Minimum Information About a Proteomics Experiment (MIAPE). *Nat. Biotechnol.*, 25(8), 887–893.
- 25** Martínez-Bartolomé, S., Binz, P.-A., and Albar, J.P. (2014) The Minimal Information About a Proteomics Experiment (MIAPE) from the Proteomics Standards Initiative. *Methods Mol. Biol.*, 1072, 765–780.
- 26** Le Novère, N., Finney, A., Hucka, M., Bhalla, U.S., Campagne, F., Collado-Vides, J. *et al.* (2005) Minimum Information Requested in the Annotation of Biochemical Models (MIRIAM). *Nat. Biotechnol.*, 23(12), 1509–1515.
- 27** Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J.C., Kitano, H. *et al.* (2003) The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. *Bioinforma.*, 19(4), 524–531.
- 28** Hucka, M., Finney, A., Bornstein, B.J., Keating, S.M., Shapiro, B.E., Matthews, J. *et al.* (2004) Evolving a lingua franca and associated software infrastructure for computational systems biology: the Systems Biology Markup Language (SBML) project. *Syst. Biol.*, 1(1), 41–53.
- 29** Demir, E., Cary, M.P., Paley, S., Fukuda, K., Lemer, C., Vastrik, I. *et al.* (2010) The BioPAX community standard for pathway data sharing. *Nat. Biotechnol.*, 28(9), 935–942.
- 30** Kitano, H. (2003) A graphical notation for biochemical networks. *BIOSILICO*, 1(5), 169–176.
- 31** Kitano, H., Funahashi, A., Matsuoka, Y., and Oda, K. (2005) Using process diagrams for the graphical representation of biological networks. *Nat. Biotechnol.*, 23(8), 961–966.
- 32** Kohn, K.W. (1999) Molecular interaction map of the mammalian cell cycle control and DNA repair systems. *Mol. Biol. Cell*, 10(8), 2703–2734.
- 33** Pirson, I., Fortemaison, N., Jacobs, C., Dremier, S., Dumont, J.E., and Maenhaut, C. (2000) The visual display of regulatory information and networks. *Trends Cell Biol.*, 10(10), 404–408.
- 34** Le Novère, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A. *et al.* (2009) The Systems Biology Graphical Notation. *Nat. Biotechnol.*, 27(8), 735–741.
- 35** Bader, G.D., Cary, M.P., and Sander, C. (2006) PathGuide: a pathway resource list. *Nucleic Acids Res.*, 34 (Database issue), D504–D506.
- 36** Milo, R., Jorgensen, P., Moran, U., Weber, G., and Springer, M. (2010) BioNumbers – the database of key numbers in molecular and cell biology. *Nucleic Acids Res.*, 38 (Database issue), D750–D753.
- 37** Kanehisa, M., Goto, S., Sato, Y., Furumichi, M., and Tanabe, M. (2012) KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Res.*, 40 (Database issue), D109–D114.
- 38** Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., and Hattori, M. (2004) The KEGG resource for deciphering the genome. *Nucleic Acids Res.*, 32 (Database issue), D277–D280.
- 39** Goto, S., Okuno, Y., Hattori, M., Nishioka, T., and Kanehisa, M. (2002) LIGAND: database of chemical compounds and reactions in biological pathways. *Nucleic Acids Res.*, 30(1), 402–404.
- 40** Croft, D., Mundo, A.F., Haw, R., Milacic, M., Weiser, J., Wu, G. *et al.* (2014) The Reactome pathway knowledgebase. *Nucleic Acids Res.*, 42 (Database issue), D472–D477.
- 41** Milacic, M., Haw, R., Rothfels, K., Wu, G., Croft, D., Hermjakob, H. *et al.* (2012) Annotating cancer variants and anti-cancer therapeutics in Reactome. *Cancers*, 4(4), 1180–1211.
- 42** Kelder, T., van Iersel, M.P., Hanspers, K., Kutmon, M., Conklin, B.R., Evelo, C.T. *et al.* (2012) WikiPathways: building research communities on biological pathways. *Nucleic Acids Res.*, 40 (Database issue), D1301–D1307.
- 43** Kamburov, A., Wierling, C., Lehrach, H., and Herwig, R. (2009) ConsensusPathDB – a database for integrating human functional interaction networks. *Nucleic Acids Res.*, 37 (Database issue), D623–D628.
- 44** Kamburov, A., Pentchev, K., Galicka, H., Wierling, C., Lehrach, H., and Herwig, R. (2011) ConsensusPathDB: toward a more complete picture of cell biology. *Nucleic Acids Res.*, 39 (Database issue), D712–D717.
- 45** Kamburov, A., Stelzl, U., Lehrach, H., and Herwig, R. (2013) The ConsensusPathDB interaction database: 2013 update. *Nucleic Acids Res.*, 41 (Database issue), D793–D800.
- 46** Olivier, B.G. and Snoep, J.L. (2004) Web-based kinetic modelling using JWS Online. *Bioinforma.*, 20(13), 2143–2144.
- 47** Karr, J.R., Sanghvi, J.C., Macklin, D.N., Gutschow, M.V., Jacobs, J.M., Bolival, B., Assad-Garcia, N., Glass, J.I., and Covert, M.W. (2012) A whole-cell computational model predicts phenotype from genotype. *Cell*, 150(2), 389–401.

- 48** Krause, F., Schulz, M., Swainston, N., and Liebermeister, W. (2011) Sustainable model building: the role of standards and biological semantics. *Methods Enzymol.*, 500, 371–395.
- 49** Funahashi, A., Matsuoka, Y., Jouraku, A., Morohashi, M., Kikuchi, N., and Kitano, H. (2008) CellDesigner 3.5: a versatile modeling tool for biochemical networks. *Proc. IEEE*, 96(8), 1254–1265.
- 50** Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. (2006) COPASI – a COmplex PATHway SIMulator. *Bioinformatics*, 22(24), 3067–3074.
- 51** Le Novère, N., Finney, A., Hucka, M., Bhalla, U.S., Campagne, F., Collado-Vides, J., Crampin, E.J., Halstead, M., Klipp, E., Mendes, P., et al. (2005) Minimum Information Requested in the Annotation of Biochemical Models (MIRIAM). *Nat. Biotechnol.*, 23(12), 1509–1515.
- 52** Le Novère, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M.I., Wimalaratne, S.M., Bergman, F.T., Gauges, R., Ghazal, P., Kawaji, H., Li, L., Matsuoka, Y., Villéger, A., Boyd, S.E., Calzone, L., Courtot, M., Dogrusoz, U., Freeman, T.C., Funahashi, A., Ghosh, S., Jouraku, A., Kim, S., Kolpakov, F., Luna, A., Sahle, S., Schmidt, E., Watterson, S., Wu, G., Goryanin, I., Kell, D.B., Sander, C., Sauro, H., Snoep, J.L., Kohn, K., and Kitano, H. (2009) The Systems Biology Graphical Notation. *Nat. Biotechnol.*, 27(8), 735–741.
- 53** Waltemath, D., Adams, R., Bergmann, F.T., Hucka, M., Kolpakov, F., Miller, A.K., Moraru, I.I., Nickerson, D., Snoep, J.L., and Le Novère, N. (2011) Reproducible computational biology experiments with SED-ML – the Simulation Experiment Description Markup Language. *BMC Syst. Biol.*, 5, 198.
- 54** Chelliah, V., Endler, L., Juty, N., Laibe, C., Li, C., Rodriguez, N., and Le Novère, N. (2009) Data integration and semantic enrichment of systems biology models and simulations, in *Data Integration in the Life Sciences – 6th International Workshop (DILS 2009)*, Lecture Notes in Computer Science, vol. 5647, Springer, pp. 5–15.
- 55** Waltemath, D., Adams, R., Beard, D.A., Bergmann, F.T., Bhalla, U.S. et al. (2011) Minimum Information About a Simulation Experiment (MIASE). *PLoS Comput. Biol.*, 7(4), e1001122.
- 56** Taylor, C.F. et al. (2008) Promoting coherent minimum reporting guidelines for biological and biomedical investigations: the MIBBI project. *Nat. Biotechnol.*, 26(8), 889–896.
- 57** Le Novère, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., Li, L., Sauro, H., Schilstra, M., Shapiro, B., Snoep, J.L., and Hucka, M. (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.*, 34 (Database issue), D689–D691.
- 58** Büchel, F. et al. (2013) Path2Models: large-scale generation of computational models from biochemical pathway maps. *BMC Syst. Biol.*, 7, 116.
- 59** Courtot, M. et al. (2011) Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol.*, 7, 543.
- 60** Noy, N.F., Shah, N.H., Whetzel, P.L., Dai, B., Dorf, M., Griffith, N., Jonquet, C., Rubin, D.L., Storey, M.A., Chute, C.G., and Musen, M.A. (2009) BioPortal: ontologies and integrated data resources at the click of a mouse. *Nucleic Acids Res.*, 37, W170–173.
- 61** Le Novère, N., Finney, A., Hucka, M., Bhalla, U.S., Campagne, F., Collado-Vides, J., Crampin, E.J., Halstead, M., Klipp, E., Mendes, P., Nielsen, P., Sauro, H., Shapiro, B., Snoep, J.L., Spence, H.D., and Wanner, B.L. (2005) Minimum Information Requested in the Annotation of Biochemical Models (MIRIAM). *Nat. Biotechnol.*, 23(12), 1509–1515.
- 62** Degtyarenko, K., de Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcantara, R., Darsow, M., Guedj, M., and Ashburner, M. (2008) ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Res.*, 36 (Database issue), D344.
- 63** Juty, N., Le Novère, N., and Laibe, C. (2012) [Identifiers.org](#) and MIRIAM Registry: community resources to provide persistent identification. *Nucleic Acids Res.*, 40(D1), D580–D586.
- 64** Schulz, M., Krause, F., Le Novère, N., Klipp, E., and Liebermeister, W. (2011) Retrieval, alignment, and clustering of computational models based on semantic annotations. *Mol. Syst. Biol.*, 7, 512.
- 65** Trißl, S., Hussels, P., and Leser, U. (2012) InterOnto – ranking inter-ontology links, in *Data Integration in the Life Sciences*, Lecture Notes in Computer Science, vol. 7348 (eds O. Bodenreider and B. Rance), Springer, pp. 5–20.
- 66** Schulz, M., Klipp, E., and Liebermeister, W. (2012) Propagating semantic information in biochemical network models. *BMC Bioinform.*, 13, 18.
- 67** Hornberg, J.J., Bruggeman, F.J., Binder, B., Geest, C.R., de Vaate, A.J.M.B., Lankelma, J., Heinrich, R., and Westerhoff, H.V. (2005) ERK phosphorylation and kinase/phosphatase control. *FEBS J.*, 272 (1), 244–258.
- 68** Huang, C.Y. and Ferrell, J.E. (1996) Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proc. Natl. Acad. Sci. USA*, 93(19), 10078.
- 69** Bhalla, U.S. and Iyengar, R. (1999) Emergent properties of networks of biological signaling pathways. *Supramol. Sci.*, 283(5400), 381–387.
- 70** Snoep, J.L., Bruggeman, F., Olivier, B.G., and Westerhoff, H.V. (2006) Towards building the silicon cell: a modular approach. *Biosystems*, 83, 207–216.
- 71** Schulz, M., Uhendorf, J., Klipp, E., and Liebermeister, W. (2006) SBMLmerge, a system for combining biochemical network models. *Genome Inform.*, 17(1), 62–71.
- 72** Liebermeister, W. (2008) Validity and combination of biochemical models. *Proceedings of 3rd International ESCEC Workshop on Experimental Standard Conditions on Enzyme Characterizations*.
- 73** Krause, F., Uhendorf, J., Lubitz, T., Klipp, E., and Liebermeister, W. (2010) Annotation and merging of SBML models with semanticSBML. *Bioinformatics*, 26(3), 421–422.
- 74** Randhawa, R., Shaffer, C.A., and Tyson, J.J. (2009) Model aggregation: a building-block approach to creating large macromolecular regulatory networks. *Bioinformatics*, 25(24), 3289–3295.

Further Reading

MIRIAM rules: Le Novère, N., Finney, A., Hucka, M., Bhalla, U.S., Campagne, F., Collado-Vides, J., Crampin, E.J., Halstead, M., Klipp, E., Mendes, P. et al. (2005) Minimum Information Requested in the Annotation of Biochemical Models (MIRIAM). *Nat. Biotechnol.*, 23 (12), 1509–1515.

Systems Biology Markup Language: Hucka, M. et al. (2003) The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. *Bioinforma*, 19(4), 524–531.

SBGN: Le Novère, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M.I., Wimalaratne, S.M., Bergman, F.T., Gauges, R., Ghazal, P., Kawaji, H., Li, L., Matsuoka, Y., Villéger, A., Boyd, S.E., Calzone, L., Courtot, M., Dogrusoz, U., Freeman, T.C., Funahashi, A., Ghosh, S., Jouraku, A., Kim, S., Kolpakov, F., Luna, A., Sahle, S., Schmidt, E., Watterson, S., Wu, G., Goryanin, I., Kell, D.B., Sander, C., Sauro, H., Snoep, J.L., Kohn, K., and Kitano, H. (2009) The Systems Biology Graphical Notation. *Nat. Biotechnol.*, 27(8), 735–741.

BioModels Database: Le Novère, N., Bornstein, B., Broicher, A., Courtot, M., Donizelli, M., Dharuri, H., Li, L., Sauro, H., Schilstra, M., Shapiro, B., Snoep, J.L., and Hucka, M. (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.*, 34 (Database issue), D689–D691.

Model semantics: Courtot, M. *et al.* (2011) Controlled vocabularies and semantics in systems biology. *Mol. Syst. Biol.*, 7, 543.

Semantic model comparison: Schulz, M., Krause, F., Le Novère, N., Klipp, E., and Liebermeister, W. (2011) Retrieval, alignment, and clustering of computational models based on semantic annotations. *Mol. Syst. Biol.*, 7, 512.

Sustainable modeling: Krause, F., Schulz, M., Swainston, N., and Liebermeister, W. (2011) Sustainable model building: the role of standards and biological semantics. *Methods Enzymol.*, 500, 371–395.

Model Fitting, Reduction, and Coupling

6

Introduction

Cells and organisms are incredibly complex, and the only way to understand them is through simplified pictures. Simplicity comes from omitting irrelevant details. For instance, we may picture a pathway in isolation, pretending that its environment is given and constant, or we may neglect microscopic details within a system: Instead of considering all reaction events between single molecules, we average over them and see a smooth behavior of macroscopic concentrations.

Simplified models will be approximations, but approximations are exactly what we are aiming at: models that provide a good level of details and are easy to work with. Keeping in mind that cells are much more complex than our models (and that effective quantities like free energies summarize a lot of microscopic complexity in a single number), we can move between models of different scope whenever necessary. If models turn out to be too simple, we zoom in and consider previously neglected details, or zoom out and include parts of the environment into our model.

Early biochemical models were simple and mostly used for studying general principles, for example, the emergence of oscillations from feedback regulation. With increasing amounts of data, models of metabolism, cell cycle, or signaling pathways have become more complex, more accurate, and more predictive. As more data are becoming available, biochemical models are increasingly used to picture biological reality and specific experimental situations.

Building a useful model is usually a lengthy process. Based on literature studies and collected data, one starts by developing hypotheses about the biological system. Which objects and processes (e.g., substances, reactions, cell compartments) are relevant? Which mathematical

Introduction

6.1 Parameter Estimation

- Regression, Estimators, and Maximal Likelihood
- Parameter Identifiability
- Bootstrapping
- Bayesian Parameter Estimation
- Probability Distributions for Rate Constants
- Optimization Methods

6.2 Model Selection

- What Is a Good Model?
- The Problem of Model Selection
- Likelihood Ratio Test
- Selection Criteria
- Bayesian Model Selection

6.3 Model Reduction

- Model Simplification
- Reduction of Fast Processes
- Quasi-Equilibrium and Quasi-Steady State
- Global Model Reduction

6.4 Coupled Systems and Emergent Behavior

- Modeling of Coupled Systems
- Combining Rate Laws into Models
- Modular Response Analysis
- Emergent Behavior in Coupled Systems
- Causal Interactions and Global Behavior

Exercises

References

Further Reading

framework is appropriate (continuous or discrete, kinetic or stochastic, spatial or nonspatial model)? Sometimes, model structures must meet specific criteria, for example, known robustness properties. The bacterial chemotaxis system, for instance, shows a precise adaptation to external stimuli, and a good model should account for this

fact; models that implement perfect adaptation by their network structure will be even more plausible than models that require a fine-tuning of parameters (see Section 10.2).

Modeling often involves several cycles of model generation, fitting, testing, and selection [1,2]. To choose between different model variants, new experiments may be required. In practice, the choice of a model depends on various factors: Does a model reflect the basic biological facts about the system? Is it simple enough to be simulated and fitted? Does it explain known observations and can it predict previously unobserved behavior? Sometimes, a model's structure (e.g., stoichiometric matrix and rate laws) is already known, while the parameter values (e.g., rate constants or external concentrations) still have to be determined. They can be obtained by fitting the model outputs (e.g., concentration time series) to a set of experimental data. If the model is structurally identifiable (a property that will be explained below), if enough data are available, and if the data are free of errors, this procedure should allow us to determine the true parameter set: For all other parameter sets, data and model output would differ. In reality, however, these conditions are rarely met. Therefore, modelers put large efforts in parameter estimation and in developing efficient optimization algorithms.

Without comprehensive data, many parameters remain nonidentifiable and there will be a risk of overfitting. This calls for models that are simple enough to be fully validated by existing data, which limits the degree of details in realistic models. The process of model simplification can be systematized: Different hypotheses and simplifications may lead to different model variants, and statistical methods can help us select based on data. Using optimal experimental design [3], and based on preliminary models, one may devise experiments that are most likely to yield the information needed to select between possible models. This cycle of experiments and modeling enables us to overcome many limitations of one-step model selection. Model selection can be insightful: For instance, the fact that fast equilibrated processes cannot be resolved using data can tell us that also the cellular machinery may take these fast processes "for granted" when controlling processes on slower time scales.

6.1 Parameter Estimation

Summary

Parameter values can be obtained by fitting a model to experimental data, that is, by minimizing the mismatch between predictions and data. Uncertainties in estimated

parameters can be assessed by bootstrapping, that is, repeated estimation with resampled data, and the quality of model predictions can be tested by cross-validation. In Bayesian statistics, parameters are characterized by a posterior parameter distribution based on available data and prior knowledge. Measured rate constants, even incomplete, redundant, and contradictory ones, can be used to estimate consistent parameter sets for kinetic models by parameter balancing. Parameter estimation leads to optimization problems. Global optimization, for example, by simulated annealing or genetic algorithms, can evade local minima, but can be numerically demanding.

In mathematical models, for example, the kinetic models discussed before, each possible parameter set θ is mapped to a set of predicted data values x via a function $x(\theta)$. If measurement data are given, we can try to invert this procedure and reconstruct the unknown parameter set θ from the data. There are three possible cases: (i) the parameters are uniquely determined; (ii) the parameters are overdetermined (i.e., no parameter set corresponds to the data) because data are noisy or the model is wrong; (iii) the parameters are underdetermined, that is, several or many parameter sets explain the data equally well.

Case (i) is mainly theoretical; in reality, when simple models are applied to complex phenomena, we do not expect perfect predictions. Cases (ii) and (iii) are what we normally encounter. Pragmatically, it is common to assume that a model is correct and that discrepancies between data and model predictions are caused by measurement errors. This resolves the problem of overdetermined parameters (case (ii)) and allows us to fit model parameters and to compare different models by their data fits. We can use data and statistical methods to obtain *parameter estimates* that approximate the true parameter values, as well as the uncertainty of the estimates [4]. Underdetermined parameters (case (iii)) can either be accepted as a fact or specific parameters can be chosen by additional regularization criteria. Common modeling tools such as COPASI [5] support parameter estimation.

6.1.1

Regression, Estimators, and Maximal Likelihood

6.1.1.1 Regression

Linear regression is a typical example of parameter estimation. A linear regression problem is shown in Figure 6.1: A number of data points (t_m, y_m) have to be approximated by a straight line $x = f(t, \theta) = \theta_1 t + \theta_2$. If the data points are indeed located exactly on a line, we can choose a parameter vector $\theta = (\theta_1, \theta_2)^T$ such that

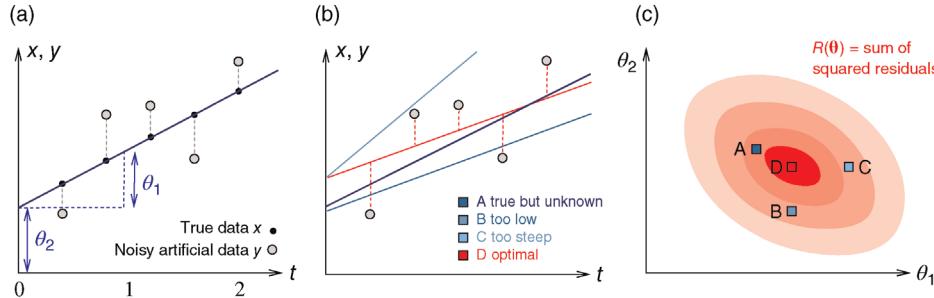


Figure 6.1 Linear regression. (a) Artificial data \$(t_m, y_m)\$ (gray) are generated by computing data points (black) from a model \$x(t) = \theta_1 t + \theta_2\$ (straight line) and adding Gaussian noise. Each possible line is characterized by two parameters, slope \$\theta_1\$ and offset \$\theta_2\$. The aim in linear regression is to reconstruct the unknown parameters from noisy data (in this case, artificial data \$y\$). (b) The deviation between a possible line (four lines A, B, C, and D are shown) and data can be measured by the sum of squared residuals. Residuals are shown for line D (dashed lines). (c) Each of the lines A, B, C, and D corresponds to a point \$(\theta_1, \theta_2)\$ in parameter space. The SSR as a function \$R(\theta)\$ in parameter space can be pictured as a landscape (shades of pink; dark pink indicates small SSR, that is, a good fit). Line D minimizes the SSR value. Small SSR values correspond to a large likelihood.

\$y_m = f(t_m, \theta)\$ holds for all data points \$m\$. In practice, measured data will scatter, and we search for a regression line as close as possible to the data points. If the deviation between line and data is scored by the sum of squared residuals (SSR), \$R(\theta) = \sum_m (y_m - f(t_m, \theta))^2\$, the regression problem can be seen as a minimization problem for the function \$R(\theta)\$.

6.1.1.2 Estimators and Maximal Likelihood

The use of the SSR as a distance measure is justified by statistical arguments, assuming that the data stem from some generative model with unknown parameters and additional random errors. As an example, we consider a curve \$f(t, \theta)\$ with an independent variable \$t\$ (e.g., time) and curve parameters \$\theta_1, \dots, \theta_N\$. For a number of time points \$t_m\$, the model yields the output values \$x_m = f(t_m, \theta)\$, which can be seen as a vector \$\mathbf{x} = \mathbf{x}(\theta)\$. By adding random errors \$\xi_m\$, we obtain the noisy data:

$$y_m = f(t_m, \theta) + \xi_m. \quad (6.1)$$

If the errors follow independent Gaussian distributions with mean 0 and variance \$\sigma_m^2\$, each data point \$y_m\$ is a Gaussian-distributed random number with mean \$f(t_m, \theta)\$ and variance \$\sigma_m^2\$.

In parameter estimation, we revert this process: Starting with a model (characterized by a function \$\mathbf{x}(\theta)\$ and noise variances \$\sigma_m^2\$) and a set of noisy data \$\mathbf{y}\$ (some realization of Eq. (6.1)), we try to infer the unknown parameter set \$\theta\$. A function \$\hat{\theta}(\mathbf{y})\$ mapping each data set \$\mathbf{y}\$ to an estimated parameter vector \$\hat{\theta}\$ is called an *estimator*.

Practical and fairly simple estimators follow from the *principle of maximum likelihood*. The likelihood of a model is the conditional probability to observe the actually observed data \$\mathbf{y}\$ if the model is correct. For instance, if a model predicts “Tomorrow, the sun will shine with 80% probability,” and sunshine is indeed observed, the

model has a likelihood of 0.8. Accordingly, if a model structure is given, each possible parameter set \$\theta\$ has a likelihood \$L(\theta|\mathbf{y}) = p(\mathbf{y}|\theta)\$. To compute likelihood values for a biochemical model, we need to specify our assumptions about possible measurement errors. From a generative model like Eq. (6.1), the probability density \$p(\mathbf{y}|\theta)\$ to observe the data \$\mathbf{y}\$, given the parameters \$\theta\$, can be computed. The maximum likelihood estimator

$$\hat{\theta}_{\text{ML}}(\mathbf{y}) = \arg \max_{\theta} L(\theta|\mathbf{y}) \quad (6.2)$$

yields the parameter set that would maximize the likelihood given the observed data. If the maximum point is not unique, the model parameters cannot be identified by maximum likelihood estimation.

6.1.1.3 Method of Least Squares

To see how likelihood functions can be computed in practice, we return to the model Eq. (6.1) with additive Gaussian noise. If our model yields a true value \$x_m\$, the observable noisy value \$y_m\$ has a probability density \$p_\xi(y_m - x_m)\$, where \$p_\xi(\xi)\$ is the probability density of the error term. If the variables \$\xi_m\$ are independently Gaussian-distributed with mean 0 and variance \$\sigma_m^2\$, their density reads

$$p_{\xi_m}(\xi) = \frac{1}{\sqrt{2\pi\sigma_m^2}} e^{-(\xi^2/2\sigma_m^2)}. \quad (6.3)$$

Given one data point \$y_m\$, we obtain the likelihood function:

$$L(\theta|y_m) = p(y_m|\theta) = p_{\xi_m}(y_m - x_m(\theta)). \quad (6.4)$$

Given all data points and assuming that random errors are independent, the probability to observe the data set \$\mathbf{y}\$ is the product of all probabilities for individual data points. Hence, the likelihood reads

$$L(\theta|\mathbf{y}) = p(\mathbf{y}|\theta) = \prod_m p_{\xi_m}(y_m - x_m(\theta)). \quad (6.5)$$

By inserting the Gaussian probability density (6.3) and taking the logarithm, we obtain

$$\ln L(\boldsymbol{\theta}|\mathbf{y}) = \sum_m -\frac{(y_m - x_m(\boldsymbol{\theta}))^2}{2\sigma_m^2} + \text{const.} \quad (6.6)$$

If all random errors ξ_m have the same variance σ^2 , the logarithmic likelihood reads

$$\begin{aligned} \ln L(\boldsymbol{\theta}|\mathbf{y}) &= -\frac{1}{2\sigma^2} \sum_m (y_m - x_m(\boldsymbol{\theta}))^2 + \text{const.} \\ &= -\frac{R(\boldsymbol{\theta})}{2\sigma^2} + \text{const.}, \end{aligned} \quad (6.7)$$

where $R(\boldsymbol{\theta}) = \|\mathbf{y} - \mathbf{x}(\boldsymbol{\theta})\|^2$ is the sum of squared residuals. Thus, maximum likelihood estimation with the error model (6.3) is equivalent to a minimization of weighted squared residuals. This justifies the method of least squares. The same argument holds for data on logarithmic scale. Additive Gaussian errors on logarithmic scale correspond to multiplicative log-normal errors for the original, nonlogarithmic data. In this case, a uniform variance σ^2 for all data points means that all nonlogarithmic data points have the same range of *relative errors*. According to the Gauss–Markov theorem, the least-squares estimator is a best linear unbiased estimator if the model is linear and if the errors ξ_m for different data points are linearly uncorrelated and have the same variance. The error distribution $p_\xi(\xi)$ need not be Gaussian.

6.1.2 Parameter Identifiability

If a likelihood function is seen as a landscape in parameter space, the maximum likelihood estimate $\hat{\boldsymbol{\theta}}$ is the maximum point in this landscape. According to Eq. (6.6), this point corresponds to a minimum of the weighted SSR (compare with Figure 6.1). In a single isolated maximum, the logarithmic likelihood function $\ln L(\boldsymbol{\theta}|\mathbf{y})$ has strictly

negative curvatures and can be approximated using the local curvature matrix $\partial^2 \ln L(\boldsymbol{\theta}|\mathbf{y}) / \partial \boldsymbol{\theta}_i \partial \boldsymbol{\theta}_k$. Directions with weak curvatures correspond to parameter variations that have little effect on the likelihood.

If several parameter sets fit the data equally well, the maximum likelihood criterion does not yield a unique solution, and the estimation problem is underdetermined (Figure 6.2). In particular, the likelihood function may be maximal on an entire curve or surface in parameter space. This can have different reasons: structural or practical nonidentifiability.

6.1.2.1 Structural Nonidentifiability

If two parameters θ_a and θ_b in a model appear only as a product $c = \theta_a \theta_b$, they are underdetermined: For any choice of λ , the pair $\theta_a' = \lambda \theta_a$ and $\theta_b' = \theta_b / \lambda$ will yield the same result $\theta_a' \theta_b' = \theta_a \theta_b$ and lead to the same model predictions. Thus, maximum likelihood estimation (which compares model predictions with data) can determine the product $c = \theta_a \theta_b$, but not the individual values θ_a and θ_b . Such models are called *structurally nonidentifiable*. To resolve this problem, we may replace the product $\theta_a \theta_b$ by a new identifiable parameter θ_c . Structural nonidentifiability can arise in various ways and may be difficult to detect and resolve.

6.1.2.2 Practical Nonidentifiability

Even structurally identifiable models may be *practically nonidentifiable* with respect to some parameters if data are insufficient, that is, if too few or the wrong kind of data are given. In particular, if the number of parameters exceeds the number of data points, the parameters cannot be identified. Let us assume that each possible parameter set $\boldsymbol{\theta}$ would yield a prediction $\mathbf{x}(\boldsymbol{\theta})$ and that the mapping $\boldsymbol{\theta} \rightarrow \mathbf{x}$ is continuous. If the dimensionality of $\boldsymbol{\theta}$ is larger than the dimensionality of \mathbf{x} , it is impossible to invert the function $\mathbf{x}(\boldsymbol{\theta})$ and to reconstruct the

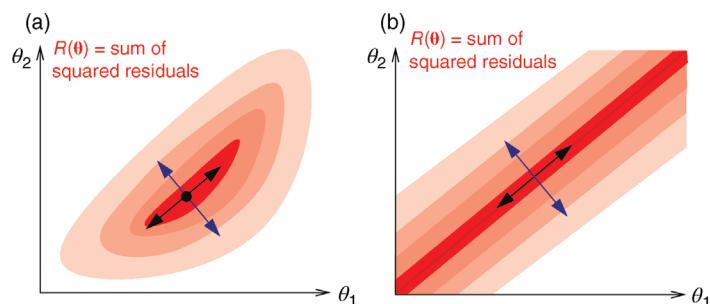


Figure 6.2 Identifiability. (a) In identifiable models, the sum of squared residuals (SSR, shown in pink) has a single minimum point (dot). The curvatures (i.e., second derivatives) of the SSR form a matrix. Its eigenvectors point toward directions of maximal (blue) and minimal curvatures (black). (b) In the nonidentifiable model shown, the SSR is minimized on a line in parameter space. A linear combination of parameters (blue arrow) is identifiable from the data, while another one (black arrow) is nonidentifiable, visible from the vanishing curvatures along this direction.

parameters from the given data. Rules for the minimum numbers of experimental data needed to reconstruct differential equation models can be found in Ref. [6].

If a model is nonidentifiable, numerical parameter optimization with different starting points will yield different estimates $\hat{\theta}$, all located on the same manifold in parameter space. In the example above, the estimates for θ_a and θ_b would differ, but they would all satisfy the relation $\ln \theta_a + \ln \theta_b = \ln c$ with a fixed value for c . On logarithmic scale, the estimates would lie on a straight line (provided that all other model parameters are identifiable).

Problems like parameter identification in which the procedure of model simulation is reverted are called *inverse problems*. If the solution of an inverse problem is not unique, the problem is *ill-posed* and additional assumptions are required for a unique solution. For instance, we may postulate that the sum of squares of all parameter values should be minimized. Such requirements can help us select a particular solution; the method is called *regularization*.

6.1.3 Bootstrapping

From a noisy data set $\mathbf{y} = \mathbf{x}(\boldsymbol{\theta}) + \xi$, we cannot determine the true model parameters $\boldsymbol{\theta}$; we only obtain an estimate $\hat{\boldsymbol{\theta}}(\mathbf{y})$. Different realizations of the random error ξ lead to different possible sets of noisy data, and thus different estimates $\hat{\boldsymbol{\theta}}$. Ideally, the mean value $\langle \hat{\boldsymbol{\theta}} \rangle$ of these estimates should yield the true parameter value (in this case, the estimator is called “unbiased”), and their variance should be small. In practice, only a single data set is available, and we obtain a single point estimate $\hat{\boldsymbol{\theta}}$. *Bootstrapping* [7] is a way to assess, at least approximately, the statistical distributions of such estimates. First, hypothetical data sets of the same size as the original data set are generated from the original data by resampling with replacement (see Figure 6.3). Then, a parameter estimate $\hat{\boldsymbol{\theta}}$ is calculated for each of these data sets. The empirical distribution of these estimates is taken as an approximation of the true distribution of $\hat{\boldsymbol{\theta}}$. The bootstrapping method is asymptotically consistent, that is, the approximation becomes exact as the size of the original data set goes to infinity.

6.1.3.1 Cross-Validation

There is a fundamental difference between model fitting and prediction. If a model has been fitted to a given data, we enforce an agreement with exactly these *training data*, and the model is likely to fit them better than it will fit *test data* that have not been used during model fitting. The fitted model will fit the data even better than the true model itself – a phenomenon called *overfitting*. However,

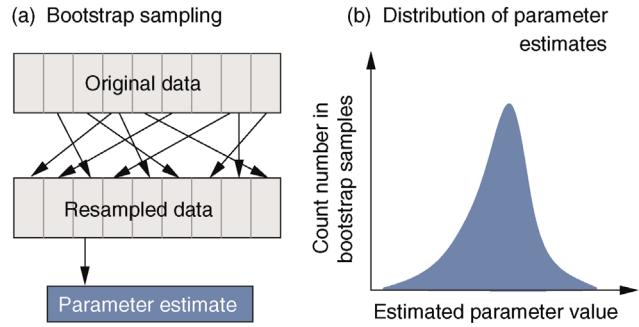


Figure 6.3 Bootstrapping with resampled data. (a) A hypothetical data set is generated by sampling values with replacement from the original data. Each resampled data set yields one parameter estimate $\hat{\boldsymbol{\theta}}$. (b) The distribution of parameter estimates obtained from bootstrap samples approximates the true distribution of the estimator $\hat{\boldsymbol{\theta}}$. For a good approximation, the original data set should be large.

overfitted models will predict new data less reliably than the true model, and their parameter values may differ strongly from the true parameter values. Therefore, overfitting should be avoided.

We have seen an example of overfitting in Figure 6.1: The least-squares regression line yields a lower SSR than the true model – because a low SSR is what it is optimized for. The apparent improvement is achieved by fitting the noise, that is, by adjusting the line to the specific realization of random errors in the data. However, this precise fit does not lead to better predictions; here, the true model will be more reliable.

How can we test whether models are overfitted? In theory, we would need new data that have not been used during model fitting. In *cross-validation* (see Figure 6.4), to mimic this test, an existing data set is split into two parts: a training set for fitting and a test set for evaluating the model predictions. This procedure is repeated many times with different parts of the data used as test sets.

Example 6.1 Bootstrapping Used for Estimating a Mean Value

Ten numbers (x_1, \dots, x_{10}) are drawn from a random distribution. We use their empirical mean value $\bar{x} = 1/10 \sum_{m=1}^{10} x_m$ to estimate the true mean value $\langle x \rangle$ of the underlying random variable X . Our aim is to assess mean and variance of the estimator \bar{x} . In bootstrapping, we randomly draw numbers z from the given sample (x_1, \dots, x_{10}) , form new tuples (bootstrap samples) $z^{(k)} = (z_1^{(k)}, \dots, z_{10}^{(k)})$, and compute the empirical mean $\bar{z}^{(k)} = 1/10 \sum_{i=1}^{10} z_i^{(k)}$ for each of them. After repeating this many times, a statistics of the values $\bar{z}^{(k)}$ can be used to approximate the true distribution of \bar{x} .

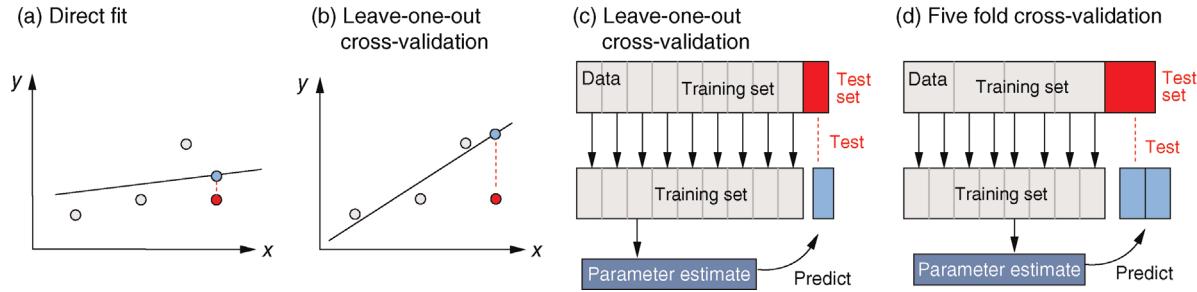


Figure 6.4 Cross-validation as a way to detect overfitting. (a) In linear regression, a straight line is fitted to data points (gray and red). The fitting error (dotted red line) is the distance between a data point (red) and the corresponding height of the regression line (blue). The regression line is chosen such that fitting errors are minimized. (b) In a leave-one-out cross-validation, we pretend that one point (red) is unknown and attempt to predict it from the model. The regression line is fitted to the remaining (gray) points, and the prediction error for the red data point will be larger than the fitting error shown in part (a). (c) Scheme of leave-one-out cross-validation. The model is fitted to all data points except for one (training set) and the remaining data point (test set) is predicted. This procedure is repeated for each data point to be predicted and yields an estimate of the average prediction error. (d) In k -fold cross-validation, the data are split into k subsets. In every run, $k-1$ subsets serve as training data, while the remaining subset is used as test data.

From the prediction errors, we can judge how the model predicts new data on average, when fitted to n data points (size of the training set). The average prediction error is an important quality measure and allows us to reject models prone to overfitting. Cross-validation, just like bootstrapping, can be numerically demanding because many estimation runs must be performed.

6.1.4

Bayesian Parameter Estimation

In maximum likelihood estimation, we search for a single parameter set representing the true parameters. Bayesian parameter estimation, an alternative approach, is based on a different premise: The parameter set θ is formally treated as a random variable. In this context, randomness describes a subjective uncertainty due to lack of information. Once more data become available, the parameter distribution can be updated and uncertainty will be reduced. By choosing a *prior* parameter distribution, we

state how plausible certain parameter sets appear in advance. Given a parameter set θ , we assume that a specific data set y will be observed with a probability density (likelihood) $p(y|\theta)$. Hence, parameters and data are described by a joint probability distribution with density $p(y, \theta) = p(y|\theta)p(\theta)$ (see Figure 6.5).

Given a data set y , we can determine the conditional probabilities of θ given y , called posterior probabilities. According to the Bayes formula,

$$p(\theta|y) = \frac{p(y|\theta)p(\theta)}{p(y)}, \quad (6.8)$$

the posterior probability density $p(\theta|y)$ is proportional to likelihood and prior density. Since the data y are given, the denominator $p(y)$ is a constant; it is used for normalization only. Bayesian estimation can also be applied recursively, that is, the posterior from one estimation can serve as a prior for another estimation with new data.

Maximum likelihood estimation and Bayesian parameter estimation differ in their practical use and in how

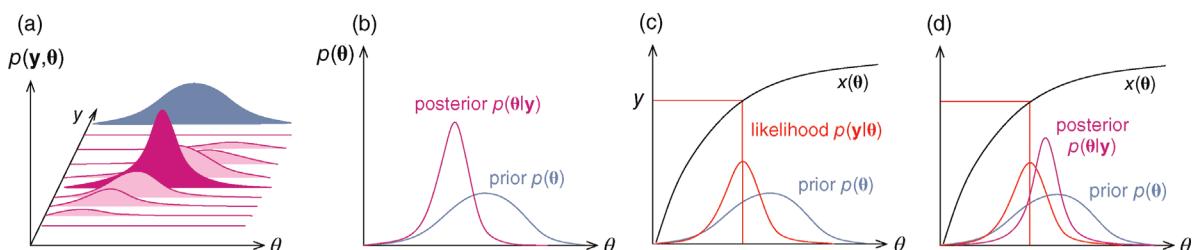


Figure 6.5 Bayesian parameter estimation. (a) In Bayesian estimation, parameters θ and data y are described by a joint probability distribution with density $p(y, \theta)$. The marginal density $p(\theta)$ of the parameters (blue) is called prior, while the conditional density $p(\theta|y)$ given a certain data set (magenta) is called posterior. (b) The posterior is narrower than the prior (blue), reflecting the information gained by the data. (c) Prior and likelihood. A variable y is given by the output $x(\theta)$ of a generative model (black line) plus Gaussian measurement errors. A given observed value y gives rise to the likelihood function $L(\theta|y) = p(y|\theta)$ in parameter space. (d) The posterior is the product of prior and likelihood function, normalized to a total probability of 1.

probabilities are interpreted. In the former, we ask: Under which hypothesis about the parameters would the data appear most probable? In the latter, we directly ask: How probable does each parameter set appear given the data? Moreover, the aim in Bayesian statistics is not to determine parameters precisely, but to characterize their posterior $p(\boldsymbol{\theta}|\mathbf{y})$ (e.g., to compute marginal distributions for individual parameters or probabilities for quantitative predictions). For complicated models, the posterior cannot be computed analytically; instead, it is common to sample parameter sets from the posterior, for instance, by using the Metropolis–Hastings algorithm described below.

Bayesian priors are used to encode general beliefs or previous knowledge about the parameter values. In practice, they can serve as regularization terms that make

Example 6.2 Combining Gaussian Priors and Posteriors

Bayesian estimation can be used to combine measured parameter values with general prior expectations. Assume that a continuous parameter $x \in \mathbb{R}$ has been measured (mean value \bar{x}^{data} , error variance $\text{var}(x^{\text{data}})$) and that a prior distribution (mean value \bar{x}^{prior} , prior variance $\text{var}(x^{\text{prior}})$) describes our general knowledge about this parameter type (from previous measurements or measurements of similar parameters). If both distributions are Gaussian, the posterior will also be Gaussian. Its standard variance and mean value read

$$\begin{aligned}\text{var}(x^{\text{post}}) &= \left(\frac{1}{\text{var}(x^{\text{prior}})} + \frac{1}{\text{var}(x^{\text{data}})} \right)^{-1}, \\ \bar{x}^{\text{post}} &= \text{var}(x^{\text{post}}) \left[\frac{\bar{x}^{\text{prior}}}{\text{var}(x^{\text{prior}})} + \frac{\bar{x}^{\text{data}}}{\text{var}(x^{\text{data}})} \right].\end{aligned}\quad (6.10)$$

The posterior precision (i.e., reciprocal variance) is the sum of the original precisions, and the posterior mean value is an average of the original mean values, weighted by their relative precisions. This approach also works for multivariate distributions and for model variables that are not measured directly, but through observables that linearly depend on them. Given data \mathbf{y} with mean vector $\bar{\mathbf{y}} = \mathbf{R}\mathbf{x}$ and covariance matrix \mathbf{C}_{data} , we obtain the posterior covariance matrix and mean vector [8]:

$$\begin{aligned}\mathbf{C}_{\text{post}} &= \left(\mathbf{C}_{\text{prior}}^{-1} + \mathbf{R}^T \mathbf{C}_{\text{data}}^{-1} \mathbf{R} \right)^{-1}, \\ \bar{\mathbf{x}}_{\text{post}} &= \mathbf{C}_{\text{post}} \left[\mathbf{C}_{\text{prior}}^{-1} \bar{x}_{\text{prior}} + \mathbf{R}^T \mathbf{C}_{\text{data}}^{-1} \bar{\mathbf{y}} \right].\end{aligned}\quad (6.11)$$

These formulas can be used, for instance, to obtain kinetic model ensembles [9].

models identifiable. By taking the logarithm of Eq. (6.8), we obtain the logarithmic posterior:

$$\ln p(\boldsymbol{\theta}|\mathbf{y}) = \ln L(\boldsymbol{\theta}|\mathbf{y}) + \ln p(\boldsymbol{\theta}) + \text{const.} \quad (6.9)$$

If the logarithmic likelihood has no unique maximum (as in Figure 6.2b), the model will not be identifiable by maximum likelihood estimation. By adding the logarithmic prior $\ln p(\boldsymbol{\theta})$ in Eq. (6.9), we can obtain a unique maximum in the posterior density.

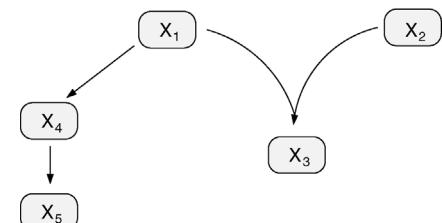
6.1.4.1 Bayesian Networks

Bayesian statistics is commonly used in probabilistic reasoning [10,11] to study relationships between uncertain facts. Facts are described by random variables (binary or quantitative), and their (assumed) probabilistic dependencies are represented as a directed acyclic graph $G(V, E)$, called Bayesian network [10,12,13]. Vertices $v \in V$ correspond to random variables x_i and edges $e \in E$ encode conditional dependencies between these variables: A variable, given the variables represented by its parent vertices, is conditionally independent on all other variables. Bayesian networks can also define a joint probability distribution precisely. Each variable x_i is associated with a conditional probability $p(x_i|L(x_i))$, where $L(x_i)$ denotes the parents of variable i , that is, the set of variables having a direct probabilistic influence on i . Together, these conditional probabilities define a joint probability distribution via the formula $p(\mathbf{x}) = \prod_{i=1}^n p(x_i|L(x_i))$. Notably, the shape of a Bayesian network is not uniquely defined by a dependence structure: Instead, the same probability distribution can be represented by different equivalent Bayesian networks, which cannot be distinguished by observation of the variables \mathbf{x} [13].

Bayesian networks have, for instance, been used to infer gene regulation networks from gene expression data. Variables x_i belonging to the vertices i are a measure of

Example 6.3 Bayesian Network

For the network shown below, the conditional independence relations read $i(x_a; x_b)$ and $i(x_d; x_a, x_b|x_c)$. The joint probability distribution of the network is $p(x_a, x_b, x_c, x_d) = p(x_a) \cdot p(x_b) \cdot p(x_c|x_a, x_b) \cdot p(x_d|x_c)$.



gene activity, for example, the expression level of a gene or the amount of active protein. The assumption is that statistical information (given the expression of gene i , the expression of gene j is independent of other genes) reflects mechanistic causality (gene product i is the only regulator of gene j).

Generally, Bayesian conditioning can be applied in different ways: to search for networks or network equivalence classes that best explain measured data; to learn the parameters of a given network; or a network can be used to predict data, for example, possible gene expression profiles. In any case, to resolve causal interactions, time series data or data from intervention experiments are needed. In dynamic Bayesian networks, time behavior is modeled explicitly: In this case, the conditional probabilities describe the state of a gene in one moment given the states of genes in the moment before [14].

6.1.5 Probability Distributions for Rate Constants

The values of rate constants, an important requisite for kinetic modeling, can be obtained in various ways: from the literature, from databases like Brenda [15] or Sabio-RK [16], from fits to dynamic data, from optimization for other objectives, or from simple guesses. Based on data, we can obtain parameter distributions. Parameters that have been measured can be described by a log-normal distribution representing the measured value and error bar; if experimental conditions are not exactly comparable, the error bar may be artificially increased. If a parameter has not been measured, one may describe it by a broader distribution. The empirical distribution of Michaelis constants in the Brenda database, which is roughly log-normal, can be used to describe a specific, but unknown Michaelis constant. Such probability distributions can be helpful for both

uncertainty analysis and parameter estimation. In maximum likelihood estimation, one can restrict the parameters to biologically plausible values and reduce the search space for parameter fitting. In Bayesian estimation, knowledge encoded in prior distributions is combined with information from the data.

6.1.5.1 Distributions of Enzymatic Rate Constants

If we neglect all possible interdependencies, we can assume separate distributions for all parameters in a kinetic model. According to Bayesian views, a probability distribution reflects what we can know about some quantity, based on data or prior beliefs. The shapes of such distributions may be chosen by the principle of minimal information (see Section 10.1) [17]: For instance, given a known mean value and variance (and no other known characteristics), we should assume a normal distribution. If we apply this principle to logarithmic parameters, the parameters themselves will follow log-normal distributions.

What could constitute meaningful mean values and variances, specifically for rate constants? This depends on what knowledge we intend to represent, namely, values for specific parameters (e.g., the K_m value of some enzyme, with measurement error) or types of parameters (K_m values in general, characterized by median value and spread). In the latter case, we may consult the distributions of measured rate constants, possibly broken down to specific enzyme classes (see Figure 6.6). Thermodynamic and enzyme kinetic parameters can be found in publications or databases like NIST [18], Brenda [15], or Sabio-RK [16]. With more information about model elements, for example, molecule structures, improved parameter estimates can be obtained from molecular modeling or machine learning [19].

6.1.5.2 Thermodynamic Constraints on Rate Constants

Whenever kinetic constants are fitted, optimized, or sampled, we should ensure that only valid parameter combinations are

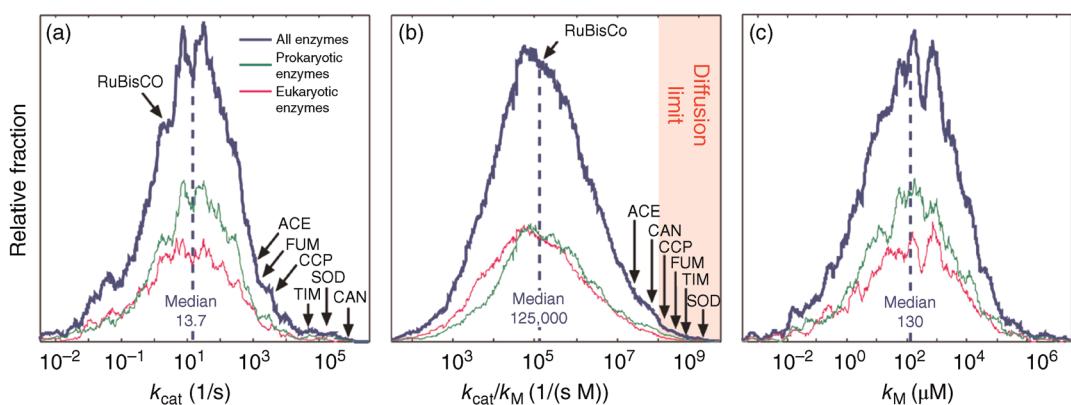


Figure 6.6 Distribution of enzymatic rate constants from the Brenda database [15]. (a) k_{cat} values in the Brenda database. (b) k_{cat}/K_m ratios. (c) K_m values. The values of RuBisCo and some prominent enzymes in *E. coli* are indicated. From Ref. [20].

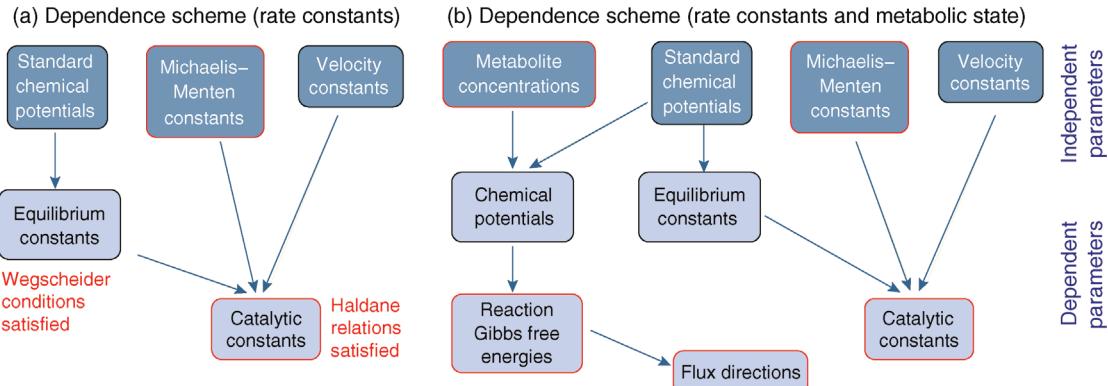


Figure 6.7 Dependence scheme for rate constants and metabolic state. (a) Dependence scheme for rate constants. To obtain consistent Michaelis–Menten constants and catalytic constants k^{cat} for a kinetic model, one treats them as part of a larger dependence scheme. In the scheme, basic parameters (top) can be chosen at will and derived parameters (center and bottom) are computed from them by linear equations. Parameters appearing in the model are marked by red frames. (b) Dependence scheme for a kinetic model in a specific metabolic state. The scheme additionally includes metabolite concentrations, chemical potentials, and reaction Gibbs free energies, which predefine the flux directions.

used. In metabolic networks with mass-action kinetics, for instance, the rate constants $k_{\pm j}$ and the equilibrium constant $K_{\text{eq},j}$ in a reaction j are related by $k_{+j}/k_{-j} = K_{\text{eq},j}$. The equilibrium constants, in turn, depend on standard chemical potentials $\mu_i^{(0)}$ via $\ln K_{\text{eq},j} = -\beta \sum_i n_{ij} \mu_i^{(0)}$, with $\beta = 1/(RT)$, which yields the condition

$$k_{+j}/k_{-j} = e^{-\beta \sum_i n_{ij} \mu_i^{(0)}}. \quad (6.12)$$

Thus, a choice of rate constants $k_{\pm j}$ will only be feasible if there exists a set of standard chemical potentials $\mu_j^{(0)}$ satisfying Eq. (6.12). If a metabolic network contains loops, it is unlikely that this test will be passed by randomly chosen rate constants.

Instead of excluding parameter sets by hindsight, based on condition (6.12), one may directly construct parameter sets that satisfy it automatically [21,22]. The equation itself shows us how to proceed: The chemical potentials $\mu_j^{(0)}$ are calculated or sampled from a Gaussian distribution and used to compute the equilibrium constants. Prefactors r_l are then sampled from a log-normal distribution, and the kinetic constants are set to $k_{\pm j} = r_j (K_{\text{eq},j})^{\pm 1/2}$. This procedure yields rate constants $k_{\pm j}$ with dependent log-normal distributions, which are feasible by construction. For Michaelis–Menten-like rate laws (e.g., convenience kinetics [23] or the modular rate laws [24]), this works similarly: Velocity constants, defined as the geometric means $k_j^V = \sqrt{k_j^{\text{cat}+} k_j^{\text{cat}-}}$ of k_{cat} values, can be used as independent basic parameters; and feasible k_{cat} values, satisfying the Haldane relationships, can be computed from equilibrium constants, Michaelis constants, and velocity constants [24].

6.1.5.3 Dependence Scheme for Model Parameters

Dependencies between rate constants can be an obstacle in modeling, but they can also be helpful: On the one

hand, measured values, when inserted into models, can lead to contradictions. On the other hand, dependencies reduce the number of free parameters, allowing us to infer unknown parameters from the other given parameters. How can we account for parameter dependencies in general, for example, when formulating a joint parameter distribution? Parameters that satisfy linear equality constraints can be expressed in terms of independent basic parameters, and their dependencies can be depicted in a scheme with separate layers for basic and derived parameters (see Figure 6.7). For instance, the vector of logarithmic rate constants in a model (vector x) can be computed from a set of basic (logarithmic) parameters (in a vector θ) by a linear equation:

$$\mathbf{x} = \mathbf{R}_\theta^x \boldsymbol{\theta}. \quad (6.13)$$

The dependence matrix \mathbf{R}_θ^x follows from the structure of the network [21,23,24,25]. In kinetic models with reversible standard rate laws (e.g., convenience kinetics [23] or thermodynamic–kinetic rate laws [22]), this parametrization can be used to guarantee feasible parameter sets. Dependence schemes cover not only rate constants but also other quantities, in particular metabolite concentrations and thermodynamic driving forces (see Figure 6.7).

A dependence scheme can help us define a joint probability distribution of all parameters. We can describe the basic parameters by independent normal distributions; the dependent parameters (which are linear combinations of them) will then be normally distributed as well. We obtain a multivariate Gaussian distribution for logarithmic parameters, and each parameter, on nonlogarithmic scale, will follow a log-normal distribution. The chosen distribution should match our knowledge about rate constants. But what can we do if standard chemical

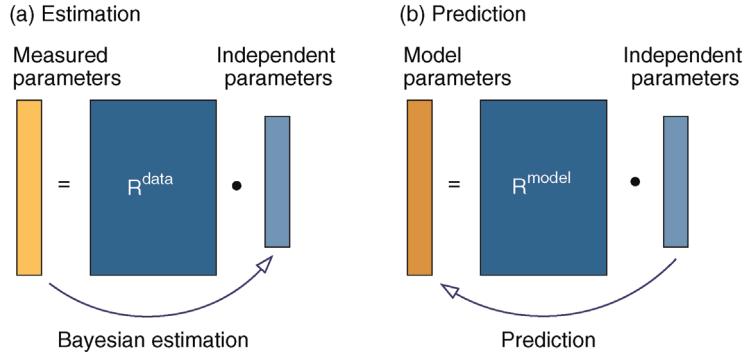


Figure 6.8 Parameter balancing. (a) Based on measured rate constants, and using a dependence scheme represented by a matrix R^{data} , the independent parameters are estimated by Bayesian multivariate regression. (b) From the posterior distribution of the basic parameters, a joint posterior of all model parameters is obtained.

potentials (as basic parameters) are poorly determined, although equilibrium constants (as dependent parameters) are almost precisely known? To express such knowledge by a joint distribution, we need to assume correlated distributions for the basic parameters. Such distributions can be determined from collected kinetic data by parameter balancing.

6.1.5.4 Parameter Balancing

Parameter balancing [9,25] is a method for estimating consistent parameter sets in kinetic models based on kinetic, thermodynamic, and metabolic data as well as prior assumptions and parameter constraints. Based on a dependence scheme, all model parameters are described as linear functions of some basic parameters with known coefficients. Using this as a linear regression model, the basic parameters can be estimated from observed parameter values using Bayesian estimation (Figure 6.8). General expectations about parameter ranges can be formulated in two ways: as prior distributions (for basic parameters) or as pseudo-values (for derived parameters). Pseudo-values appear formally as data points, but represent prior assumptions [25], providing a simple way to define complex correlated priors, for example, priors for standard chemical potentials that entail a limited variation of equilibrium constants.

To set up the regression model, we use Eq. (6.13), but consider on the left only parameters for which data exist: $\mathbf{x}^{data} = \mathbf{R}_{\theta}^{data}\boldsymbol{\theta}$. Inserting this relation into Eq. (6.11) and assuming priors for the basic parameters, we obtain the posterior covariance and mean for the basic parameters:

$$\begin{aligned}\mathbf{C}_{\theta,post} &= \left(\mathbf{C}_{prior}^{-1} + \mathbf{R}^{data T} \mathbf{C}_{data}^{-1} \mathbf{R}^{data} \right)^{-1}, \\ \bar{\boldsymbol{\theta}}_{post} &= \mathbf{C}_{post} \left[\mathbf{C}_{prior}^{-1} \bar{\boldsymbol{\theta}}_{prior} + \mathbf{R}^{data T} \mathbf{C}_{data}^{-1} \bar{\mathbf{x}}^{data} \right].\end{aligned}\quad (6.14)$$

Using Eq. (6.13) again, we obtain the posterior distribution for all parameters, characterized by

$$\begin{aligned}\mathbf{C}_{x,post} &= \mathbf{R}^{all T} \mathbf{C}_{\theta,post} \mathbf{R}^{all}, \\ \bar{\mathbf{x}}_{post} &= \mathbf{R}^{all} \bar{\boldsymbol{\theta}}_{post}.\end{aligned}\quad (6.15)$$

Parameter balancing does not yield a point estimate, but a multivariate posterior distribution for all model parameters. The posterior describes typical parameter values, uncertainties, and correlations between parameters, based on what is known from data, constraints, and prior expectations. Priors and pseudo-values keep the parameter estimates in meaningful ranges even when few data are available. Parameter balancing can also account for bounds on single parameters and linear inequalities for parameter combinations. With such constraints, the posterior becomes restricted to a region in parameter space.

Model parameters for simulation can be obtained in two ways: by using the posterior mode (i.e., the most probable parameter set) or by sampling them from the posterior distribution. By generating an ensemble of random models and assessing their dynamic properties, we can learn about potential behavior of such models, given all information used during parameter balancing. Furthermore, the posterior can be used as a prior in subsequent Bayesian parameter estimation, for example, when fitting kinetic models to flux and concentration time series [9].

Being based on linear regression and Gaussian distributions, parameter balancing is applicable to large models with various types of parameters. Flux data cannot be directly included because rate laws, due to their mathematical form, do not fit into the dependence scheme. However, if a thermodynamically feasible flux distribution is given, one can impose its signs as constraints on the reaction affinities and obtain solutions in which rate

laws and metabolic state match the given fluxes (see Section 6.4) [26].

6.1.6 Optimization Methods

Parameter fitting typically entails optimization problems:

$$\min_! f(\mathbf{x}). \quad (6.16)$$

In the method of least squares, for instance, \mathbf{x} denotes the parameter vector $\boldsymbol{\theta}$ and f is the sum of squared residuals. The allowed choices of \mathbf{x} may be restricted by constraints such as $x_i^{\min} \leq x_i \leq x_i^{\max}$. Global and local minima of f are defined as follows. A parameter set \mathbf{x}^* is a *global minimum point* if no allowed parameter set \mathbf{x} has a smaller value for f . A parameter set \mathbf{x}^* is a *local minimum point* if no other allowed parameter set \mathbf{x} in a neighborhood around \mathbf{x}^* has a smaller value. To find such optimal points numerically, algorithms evaluate the objective function f (and possibly its derivatives) in a series of points \mathbf{x} , leading to increasingly better points until a chosen convergence criterion is met.

6.1.6.1 Local Optimization

Local optimizers are used to find a local optimum in the vicinity of some starting point. *Gradient descent methods* are based on the local gradient $\nabla f(\mathbf{x})$, a vector that indicates the direction of the strongest increase of f . A sufficiently small step in the opposite direction will lead to lower function values:

$$f(\mathbf{x} - c \nabla f(\mathbf{x})) < f(\mathbf{x}) \quad (6.17)$$

for sufficiently small coefficients c . In gradient descent methods, we iteratively jump from the current point $\mathbf{x}^{(n)}$ to a new point by

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - c \nabla f(\mathbf{x}). \quad (6.18)$$

The coefficient c can be adapted in each step, for example, by a numerical line search:

$$c = \arg \min_c f(\mathbf{x} - c' \nabla \mathbf{x}). \quad (6.19)$$

Newton's method is based on a local second-order approximation of the objective function:

$$f(\mathbf{x} + \Delta \mathbf{x}) \approx f(\mathbf{x}) + \nabla f(\mathbf{x}) \Delta \mathbf{x} + \frac{1}{2} \Delta \mathbf{x}^T H(\mathbf{x}) \Delta \mathbf{x}, \quad (6.20)$$

with the curvature matrix $H_{ij} = \partial^2 f / \partial x_i \partial x_j$ (also called Hessian matrix). If we neglect the approximation error in Eq. (6.20), a direct jump $\Delta \mathbf{x}$ to an optimum would require that

$$\nabla f(\mathbf{x}) + H(\mathbf{x}) \Delta \mathbf{x} = 0. \quad (6.21)$$

In the iterative Newton method, we approximate this and jump from the current point $\mathbf{x}^{(n)}$ to a new point:

$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} - H(\mathbf{x}^{(n+1)})^{-1} \nabla f(\mathbf{x}^{(n+1)}). \quad (6.22)$$

The second term can be multiplied by a relaxation coefficient $0 < c < 1$: With smaller jumps, the iteration process will converge more stably.

6.1.6.2 Global Optimization

Theoretically, global optimum points of a function $f(\mathbf{x})$ can be found by scanning the space of \mathbf{x} values with a very fine grid. However, for a problem with n parameters and m grid values per parameter, this would require m^n function evaluations, which soon renders the problem intractable. In practice, most global optimization algorithms scan the parameter space by random jumps (Figure 6.9). The aim is to find high-quality solutions (preferably, close to a global optimum) in a short computation time or with an affordable number of function evaluations. To surmount the basins of attraction of local solutions, algorithms should allow for jumps toward

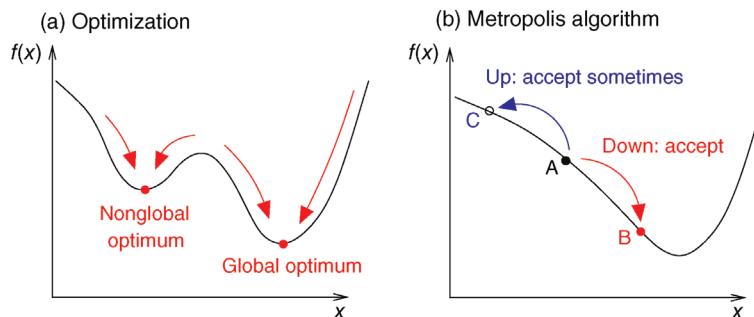


Figure 6.9 Global optimization. (a) A function may have different local minima with different function values. (b) The Metropolis-Hastings algorithm samples points \mathbf{x} by an iterative jump process: Points are sampled with probabilities $\sim \exp(-f(\mathbf{x}))$, reflecting their function values $f(\mathbf{x})$. Tentative jumps toward lower f values ($A \rightarrow B$) are always accepted, while upward jumps ($A \rightarrow C$) are accepted only with probability $p = \exp(f(x_A) - f(x_C))$.

worse solutions during the search. Among the many global optimization algorithms [27,28], popular examples are *simulated annealing* and *genetic algorithms*.

Aside from purely local and global methods, there are hybrid methods [29,30] that combine the robustness of global optimization algorithms with the efficiency of local methods. Starting from points generated in the global phase, they apply local searches that can accelerate the convergence to optimal solutions up to some orders of magnitude. Hybrid methods usually prefilter their candidate solutions to avoid local searches leading to known optima.

6.1.6.3 Sampling Methods

Simulated annealing, a popular optimization method based on sampling, is inspired by statistical thermodynamics. To use a physical analogy, we consider a particle with position x (scalar or vectorial) that moves by stochastic jumps in an energy landscape $E(x)$. In a thermodynamic equilibrium ensemble at temperature T , the particle position x follows a *Boltzmann distribution* with density

$$p(x) \sim e^{-E(x)/(k_B T)}, \quad (6.23)$$

where k_B is Boltzmann's constant (see Section 16.6). In the *Metropolis–Hastings algorithm* [31,32], this Boltzmann distribution is realized by a jump process (Monte Carlo Markov chain):

- 1) Given the current position $x^{(n)}$ with energy $E(x^{(n)})$, choose a new potential position x^* at random.
- 2) If x^* has an equal or a lower energy $E(x^*) \leq E(x^{(n)})$, accept the jump and set $x^{(n+1)} = x^*$.
- 3) If x^* has a higher energy $E(x^*) > E(x^{(n)})$, accept the jump with probability

$$p = \exp\left(\frac{E(x^{(n)}) - E(x^*)}{k_B T}\right).$$

To accept or reject a potential jump, we draw a uniform random number z between 0 and 1; if $z < p$, we accept the jump and set $x^{(n+1)} = x^*$; otherwise, we set $x^{(n+1)} = x^{(n)}$, keeping the particle at its old position.

Programming the Metropolis–Hastings algorithm is easy. The random rule for jumps in step 1 can be chosen at will, with only one restriction: The transition probability for potential jumps from state x' to state x'' must be the same as for potential jumps from x'' to x' . If this condition does not hold, the unequal transition probabilities must be compensated by a modified acceptance function in step 3. Nevertheless, the rule in step 1 must be carefully chosen: Too large jumps will mostly be rejected; too small jumps will cause the particle to stay close to its

current position; in both cases, the distribution will converge very slowly to the Boltzmann distribution. Convergence can be improved by adapting the jump rule to previous movements and, thus, to the energy landscape itself [8].

According to the Boltzmann distribution (6.23), a particle will spend more time and yield more samples in positions with low energies: The preference for low energies becomes more pronounced if the temperature is low. At temperature $T = 0$, only jumps to lower or same energies will be accepted and the particle reaches, possibly after a long time, a global energy minimum. The Metropolis–Hastings algorithm has two important applications:

- 1) *Sampling from given probability distributions* In Bayesian statistics, a common method for sampling the posterior distribution is Metropolis–Hastings sampling with fixed temperature (6.8): We set $k_B T = 1$ and choose $E(\theta) = p(y|\theta)p(\theta)$, ignoring the constant factor $1/p(y)$. From the resulting samples, we can compute, for instance, the posterior mean values and variances of individual parameters θ_i .
- 2) *Global optimization by simulated annealing* For global optimization by simulated annealing [33], $E(x)$ is replaced by some function $f(x)$ to be minimized, k_B is set to 1, and the temperature is varied during the optimization process. Simulated annealing starts with a high temperature, which is then continuously lowered during the sampling process. If the temperature falls slowly enough, the system will reach a global optimum almost certainly (i.e., with probability 1). In practice, finite run times require a faster cooling, so convergence to a global optimum is not guaranteed.

6.1.6.4 Genetic Algorithms

Genetic algorithms like *differential evolution* [34] are inspired by biological evolution. As we shall see in Section 11.1, genetic algorithms do not iteratively improve a single solution (as in simulated annealing), but simulate a whole population of possible solutions (termed “individuals”). In each step, the function value of each individual is evaluated. Individuals with high values (in relation to other individuals in the population) can have offspring, which form the following generation. In addition, mutations (i.e., small random changes) or crossover (i.e., random exchange of properties between individuals) allow the population to explore large regions of the parameter space in a short time. In problems with constraints, the *stochastic ranking* method [35] provides an efficient way to trade the objective function against the need to satisfy the constraints. Genetic algorithms are not proven to find global optima, but they are popular and have been successfully applied to various optimization problems.

6.2 Model Selection

Summary

Systems biology models must meet multiple requirements: They should fit experimental data, allow for predictions of biological behavior, represent the biological mechanisms in question, and describe them in an understandable way. Different models of the same system may comprise different levels of details or implement different biological hypotheses. If a model is too complicated, it may overfit the data. To avoid this and to select reliable models from a range of available model variants, the number of free model parameters must be restricted. Model selection can be based on the likelihood ratio test, on selection criteria like the Akaike criterion, or on Bayesian model selection.

A biochemical system can be described by multiple model variants, and choosing the right variant is a major challenge in modeling. Different model variants may cover different pathways, different substances or interactions within a pathway, different descriptions of the same process (e.g., different kinetic laws, fixed or variable concentrations), and different levels of details (e.g., subprocesses or time scales). Together, such choices can lead to a combinatorial explosion of model variants. To choose between models in a justified way, statistical methods for model selection are needed [36,37]. In fact, model fitting and model selection are very similar tasks: In both cases, we look for models that agree with biological knowledge and match experimental data; in one case, we choose between possible parameter sets, and in the other case between model structures. Moreover, model selection can involve parameter estimation for each of the candidate models.

A philosophical principle called *Ockham's razor* (*Entia non sunt multiplicanda praeter necessitatem*: Entities should not be multiplied without necessity) claims that theories should be free of unnecessary elements. Statistical model selection relies on a similar principle: Complexity in models – for example, additional substances or interactions in the network – should be avoided unless it is supported, and thus required, by data. If two models achieve equally good fits, the one with fewer free parameters should be chosen. With limited and inaccurate data, we may not be able to pinpoint a single model, but we can at least rule out models that contradict the data or for which there is no empirical evidence. With data values being uncertain, notions like “contradiction” and “evidence” can only be understood in a probabilistic sense, which calls for statistical methods.

6.2.1 What Is a Good Model?

Good models need not describe a biological system in all details. J.L. Borges writes in a story [38]: “In that empire, the art of cartography attained such perfection that the map of a single province occupied the entirety of a city, and the map of the empire, the entirety of a province. In time, those unconscionable maps no longer satisfied, and the cartographers guilds struck a map of the empire whose size was that of the empire, and which coincided point for point with it.” Similarly, systems biology models range from simple to complex “maps” of the cell, and like in real maps, details must be omitted. Otherwise, models would be as hard to understand as the biological systems described.

Models are approximations of reality or, as George Box put it, “Essentially, all models are wrong, but some are useful” [39]. In model selection, we have to make this specific and ask: useful for what? Depending on their purpose, models will have to meet various, sometimes contrary, requirements (see Figure 6.10):

- 1) In *data fitting*, we start from data and describe them by some mathematical function. A reason to do this can be simple economy of description: Instead of specifying many data pairs (x, y) on a curve, we specify much fewer curve parameters (e.g., offset and slope for a straight line). If data points deviate from our curve, we may attribute the discrepancy to measurement errors. In the context of model fitting, a dynamical model is just one specific way to define predicted curves. Given a model structure, for example, a differential equation system, we can fit the model parameters, for instance, by minimizing the sum of squared residuals.
- 2) To serve for *prediction*, a model should not just fit existing data, but also remain valid for future observations. In the language of statistical learning, it should *generalize well* to new data.

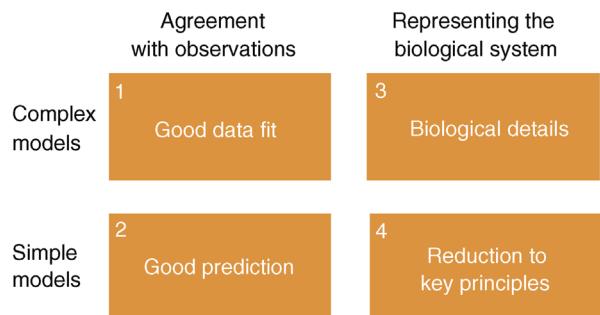


Figure 6.10 Possible requirements for good models.

- 3) *Realistic* models are supposed to picture processes “as they happen in reality.” Since there is a limit to detailing, models have to focus on certain pathways and describe them to reasonable details. Simplifying assumptions and model reduction (see Section 6.3) can be used to simplify models to a tractable level.
- 4) To highlight *key principles* of biological processes, models need to be simple. Simplicity makes models useful as didactic or prototypic examples. This holds not only for computational models but also for example cases in general, for instance, the *lac* operon as an example of microbial gene regulation.

Only few of these requirements can be tested formally, as described in Section 5.4. Moreover, different criteria can either entail or compromise each other. A good data fit, for instance, may suggest that a model is mechanistically correct and sufficiently complete. However, it may also arise in models with a very implausible structure, as long as they are flexible enough to fit various data. As a rule of thumb, models with more free parameters can fit data more easily (“With four parameters I can fit an elephant, and with five I can make him wiggle his trunk.” J. von Neumann, quoted in Ref. [40]). In this case, even though the fit becomes better, the average amount of experimental information per parameter decreases and the parameter estimates become poorly determined. This, in turn, compromises prediction. The overfitting problem notoriously occurs when many free parameters are fitted to few data points or when a large set of models is prescreened for good data fits (Freedman’s paradox [41]).

6.2.2

The Problem of Model Selection

In modeling, one typically gets to a point at which a number of model variants have been proposed and the best model is chosen by comparing model predictions with experimental data. The procedure resembles parameter fitting, as described in the previous chapter; now it is the model structure that needs to be chosen – which possibly entails parameter estimation for each of the model variants.

6.2.2.1 Likelihood and Overfitting

The quality of models can be assessed by their *likelihood*, that is, the probability that a model assigns to experimental observations. Let us assume that a model is correct in its structure and parameters; then, data values y_{tm} (component m , at time point t) will be Gaussian random variables with mean values x_{tm} and variances σ_{tm}^2 . According

Example 6.4 Reversible or Irreversible Reaction?

As a running example, we consider two models describing a reaction $S \rightleftharpoons P$. Model A assumes mass-action kinetics with rate constants k_+ and k_- and a fixed product concentration c . The substrate concentration s follows a rate equation:

$$\frac{ds}{dt} = -k_+ s + k_- c. \quad (6.24)$$

Model B assumes that the reaction is irreversible, that is, the second term vanishes (or equivalently, $k_- = 0$). By selecting this model, we would state that there is no considerable backward flux from P to S . For the concentration of S , the two models predict different curves:

$$\begin{aligned} \text{Model A : } s(t) &= s^{st} + (s_0 - s^{st}) e^{-k_+ t}, \\ \text{Model B : } s(t) &= s_0 e^{-k_+ t}, \end{aligned} \quad (6.25)$$

where s_0 is the initial concentration and $s^{st} = ck_-/k_+$ is the steady-state concentration in model A. The solution of model A depends on the values of k_+ , k_- , s_0 , and c . However, the parameters k_- and c only appear as a product $a = k_- c$, that is, they are not identifiable (see Section 6.1.3). We, therefore, use three model parameters, k_+ , s_0 , and the effective parameter a . Model B contains only two parameters, k_+ and s_0 . For model selection, we compare both models with experimental data for S . Our concentration time series consists of triples (t_i, y_i, σ_i) for the i th measurement, each containing the time point t_i , a measured concentration value y_i , and a standard error σ_i .

to Eq. (6.6), the logarithmic likelihood is related to the weighted sum of squared residuals (wSSR) as

$$\begin{aligned} -2 \ln L(\boldsymbol{\theta} | \mathbf{y}) &= -2 \ln p(\mathbf{y} | \boldsymbol{\theta}) \\ &= \sum_t^n \sum_{m=1}^n \frac{(y_{tm} - x_{tm}(\boldsymbol{\theta}))^2}{\sigma_{tm}^2}. \end{aligned} \quad (6.26)$$

The wSSR itself, as a sum of standard Gaussian distributions, follows a χ^2 -distribution with n degrees of freedom. This fact can be used for a statistical test: If the weighted SSR for the given data falls in the upper 5% quantile of the χ^2_n -distribution, we can reject the model on a 5% confidence level. If we do reject it, we conclude that the model is wrong. Importantly, a negative test – one in which the model is *not* rejected – does not prove that the model is correct; it only shows that there is not enough evidence to disprove the model. Also, note that this test works only for data that have not previously been used to fit the model.

A high likelihood can serve as a criterion for model selection *provided that* the likelihood is evaluated with

Example 6.5 Likelihood Values

In Eq. (6.25), we consider a true model of the form A with parameters $k_{\pm} = 1$, $s_0 = 1$, and $c = 0.1$. Figure 6.11 shows a simulation run of this model. To generate artificial data, we added Gaussian random numbers with a standard deviation of 10% of the true value. To reconstruct the model parameters, the data are compared with potential candidate models using the wSSR from Eq. (6.26) to measure the goodness of fit. For models A and B with predefined parameter values (see Figure 6.11a), the fits are rather poor. After maximum likelihood parameter estimation (i.e., by minimizing the weighted SSR), the fit is much closer (Figure 6.11b, numerical values in Figure 6.11c). The resulting models fit the data even better than the original model does – a clear case of overfitting. As expected, model A (with three parameters) performs better than model B (with two parameters). The question remains: Which of the two should we choose?

new data, that is, data that have not been used for fitting the model before. For instance, the statement “Tomorrow, the sun will shine with 80% probability” (obtained by some model A) can be compared with the statement “Tomorrow, the sun will shine with 50% probability” (obtained from another model B). If sunshine is observed, model A has a higher likelihood ($\text{Prob}(\text{data}|A) = 0.8$) than model B ($\text{Prob}(\text{data}|B) = 0.5$) and will be chosen by the likelihood criterion.

However, there is one big problem: When posing the question “which model fits our data best,” we usually do not compare models with *given* parameters, but fit candidate models to data and then compare them by their *maximized likelihood* values. In doing so, we use the same data twice: first for fitting the model and then for assessing its quality. This would be a wrong usage of statistics and leads to overfitting. A direct comparison by likelihood values is justified only if model parameters have been fixed in advance and without using the test data.

6.2.2.2 Methods for Model Selection

This is the central problem in model selection: We try to infer model details based on data fits, but we know that good data fits can also arise from overfitting. To counter this risk, models with more free parameters must satisfy stricter requirements regarding the goodness of fit; or, following Ockham’s principle, we should choose the most simple model variant unless others are clearly better supported by data. The problem of overfitting in model selection can be addressed in a number of ways:

- 1) *Cross-validation* In cross-validation, models are tested with data that have not been used for fitting. Like a normal goodness of fit, the mean prediction error from cross-validation can be used as a criterion to select parameters or structures of models. Notably, the cross-validation error *after this selection* will again be biased. To prove that the selected model is free of overfitting, one would need to run a nested cross-validation: an inner cross-validation loop for selecting a model, and an outer cross-validation loop to check the selected model for overfitting in an unbiased way.
- 2) *Statistical tests* In statistical tests, we compare a more complex model with a simpler background model. Our null hypothesis states that both models perform equally well. Only if the predictions of the more complex model are strikingly better, we reject this hypothesis. If we perform a test with confidence level α , and if the null hypothesis is indeed correct, there will be an $\alpha\%$ chance that we wrongly reject it.
- 3) *Selection criteria* A given set of candidate models can be scored by selection criteria [42,43,44,45]. Selection criteria are mathematical score functions that trade agreement with experimental data against model complexity: To compensate the advantage of complex models in fitting, they penalize high numbers of free parameters. Selection criteria can be used to rank models, to choose between them, and to weight predictions from different models when computing weighted averages.

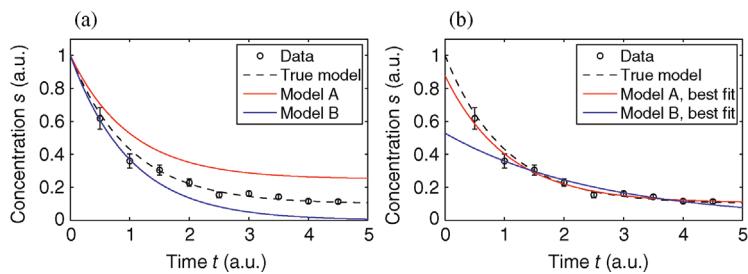


Figure 6.11 Fit of the example models (Eq. (6.25)). (a) Artificial data (black dots) are generated by adding Gaussian noise to results of the true model (dashed line). Solid curves show simulations from models A (red) and B (blue) with arbitrary parameters. (b) After parameter estimation, both models fit the data much better than before. (c) Parameter sets and goodness of fit for variants of the example model. The goodness of fit is described by the weighted sum of squared residuals, $\sum_i(y_i - x_i)^2 / \sigma_i^2$ (last column).

4) *Tests with artificial data* It can be helpful to test model fitting and selection procedures with artificial data obtained from model simulations. Knowing the true model behind the data, we can judge more easily if a model selection method is able to recover the original model.

A second problem arises when we have several models to compare. If we test many models, we are likely to find some models that fit the data well just by chance. In such cases of multiple testing, stricter significance criteria must be applied: Predefining the false discovery rate is a good strategy for choosing sensible significance levels. But let us now describe likelihood ratio test and selection criteria in more detail.

6.2.3 Likelihood Ratio Test

The *likelihood ratio test* [46] compares two models A and B (with k_A and k_B free parameters) by their maximized likelihood values L_A and L_B . Like in Eq. (6.25), the models must be nested, that is, model B must be a special case of model A with some of the parameters fixed. As a null hypothesis, we assume that both models explain the data equally well. However, even if the null hypothesis is true, model A is likely to show a higher observed likelihood because its additional parameters make it easier to fit the noise. The statistical test accounts for this fact. As a test statistic, we consider the expression $r = 2 \ln(L_A/L_B)$. Assuming that the number of data points is large and that measurement errors are independent and Gaussian distributed, this statistic will asymptotically follow a χ^2 -distribution with $k_A - k_B$ degrees of freedom. In the test, we consider the empirical value of r ; if it is significantly high, we reject the null hypothesis and accept model A. Otherwise, we accept the simpler model B. The likelihood ratio test can be sequentially applied to more than two models, provided that they are subsequently nested. Likelihood ratio tests, like other statistical tests, can only tell us whether to reject the null hypothesis model: If the test result is negative, we cannot reject the null model, that is, model A may still be the better one, but we cannot prove it.

The likelihood values and, thus, the result of the likelihood ratio test depend on the data values and their standard errors. If standard errors are known to be small, the data require an accurate fit. Accordingly, if we simply shrink all error bars, the likelihood ratio becomes larger and complex models have better chances to be selected.

Example 6.6 Likelihood Ratio Test

In example 6.5, with the values from Fig. 6.11(c), the test statistic has a value of $2 \ln(L_A/L_B) \approx 6.13 - 4.98 = 1.15$, which is well within the χ^2 -distribution: The 95% quantile for a distribution with $3 - 2 = 1$ degree of freedom is about 3.84. Therefore, the likelihood ratio test does not allow us to reject model B. However, the likelihood values depend on the noise levels $\sigma_i(t)$ in the likelihood function: If we assume a smaller noise level (but use the same artificial data, with a noise level corresponding to 10% of the original values), the weighted SSR read approximately 5.0 (model A) and 19.8 (model B). Since the test statistics has the highly significant value of $19.8 - 5.0 = 14.8$, the data support model A.

6.2.4 Selection Criteria

Nonnested models can be compared by using *selection criteria*. As we saw, if the likelihood value were not biased, it could directly be used as a criterion for model selection. However, the likelihood after fitting is biased due to overfitting. The average bias ΔL is unknown, but estimates for it have been proposed. By adding these estimates to the log-likelihood, we obtain supposedly less biased score functions, the so-called *selection criteria*. By minimizing these functions instead of the log-likelihood, we can reduce the impact of overfitting. The Akaike information criterion [47]

$$AIC = -2 \ln L(\hat{\theta}|\mathbf{y}) + 2k \quad (6.27)$$

directly penalizes the number k of free parameters. If we assume additive Gaussian measurement noise of width 1, the term $-2 \ln L(\hat{\theta}|\mathbf{y})$ in Eq. (6.27) equals the sum of squared residuals $R(\theta)$ and we obtain

$$AIC = R(\theta) + 2k. \quad (6.28)$$

A correction for small sample sizes [48] leads to

$$AICc = AIC + \frac{2k(k+1)}{n-k-1}, \quad (6.29)$$

where n is the number of data. The Schwarz criterion (Bayesian information criterion) [49]

$$BIC = -2 \ln L(\hat{\theta}|\mathbf{y}) + k \ln n \quad (6.30)$$

penalizes free parameters more strongly. In contrast to AIC, the BIC is consistent, that is, as the number of data n goes to infinity, the true model will be selected with probability 1. Selection criteria can be used for ranking and selecting models, but the statistical significance of such a model selection cannot be assessed.

Example 6.7 Selection Criteria

Table 6.1 compares different selection criteria for Example 6.5. To generate artificial data, the noise width in each data point was set to 10% of the true data value. With these noise levels, all selection criteria favor the simpler model B. However, if we refit the models to the same data and assume a smaller noise level (10% of the original values), model A is favored because the likelihood term is weighted more strongly, which then necessitates a good fit.

In some cases, selection criteria may suggest that none of the models is considerably better than the others. In such cases, instead of selecting a single model, we may combine predictions from several models. For example, to estimate a model parameter θ , we may average over the estimates θ_i obtained from several models M_i , giving higher weights to estimates from more reliable models. Weighting factors can be constructed from the selection criteria by heuristic methods like the following [45]. We compute, for each model M_i , the Akaike information criterion a_i and translate it into a likelihood score $L(M_i) = e^{-\frac{1}{2}(a_i - a_0)}$, where a_0 is the minimum value among the a_i . Models can be compared by their evidence ratios $L(M_i)/L(M_j)$ and the normalized scores $w_i = L(M_i)/\sum_j L(M_j)$, called Akaike weights, can be used to weight numerical results. If no single model is strongly supported by the data (e.g., $w_i \leq 0.9$ for all models), the Akaike weights may be used to compute weighted parameter averages $\bar{\theta} = \sum_i w_i \theta_i$ or variances. Numerical model

predictions (e.g., concentration curves) can be scored or averaged accordingly.

6.2.5

Bayesian Model Selection

The logic of maximum likelihood estimation contradicts our everyday reasoning. We do not usually ask: "Under which explanation do our observations appear most likely?" Rather we ask: "What is the most plausible explanation for our observations?" Furthermore, the maximum likelihood criterion may force us to accept explanations even if they are implausible: Imagine that you toss a coin, you obtain heads, and are asked to choose between statement A: "The coin always shows heads" and statement B: "It shows heads and tails with equal probability." According to the maximum likelihood criterion, you should choose explanation A, even if you know that usual coins do not always show heads.

Bayesian statistics provides model selection methods that are more intuitive [50,51]. We saw in Section 6.1.6 how model parameters can be scored by posterior probabilities. This concept can be extended to entire models (considering their structure \mathcal{M} and possibly their parameters Θ). Our hypotheses about plausible model structures – before observing any data – are expressed by prior probabilities $p(\mathcal{M}, \Theta)$. In the previous example, for instance, the prior could define the (very small) probability to encounter a coin that always shows heads. The posterior follows from Bayes' theorem:

$$p(\mathcal{M}, \Theta | \mathbf{y}) = \frac{p(\mathbf{y} | \mathcal{M}, \Theta)p(\mathcal{M}, \Theta)}{p(\mathbf{y})} \quad (6.31)$$

as the product of likelihood $L(\mathcal{M}, \Theta | \mathbf{y}) = p(\mathbf{y} | \mathcal{M}, \Theta)$ (which quantifies how well the model explains the data) and prior (which describes how probable the model appears in general). The marginal probability $p(\mathbf{y})$ acts as a normalization constant. After taking the logarithm, we can rewrite Eq. (6.31) as a sum:

$$\ln p(\mathcal{M}, \Theta | \mathbf{y}) = \ln p(\mathbf{y} | \mathcal{M}, \Theta) + \ln p(\mathcal{M}, \Theta) + \text{const.} \quad (6.32)$$

The purpose of Bayesian estimation is not to select a single model, but to assign probabilities to different models. From these probabilities, we can obtain distributions of single parameters, probabilities for structural features in different models, or probabilities for quantitative model predictions.

In practice, it is often impossible to compute the posterior density (Eq. (6.31)) analytically. However, Monte Carlo methods [50] allow us to sample representative models and parameter sets from the posterior. From a

Table 6.1 Selection criteria calculated for Example 6.5.

	σ Large		σ Small	
	Model A	Model B	Model A	Model B
n	3	2	...	
k	9	9		
$2k$	6	4	...	
$2k + \frac{2k(k+1)}{n-k-1}$	4.67	2.33		
$k \ln n$	6.59	4.39		
wSSR	4.98	6.13	4.99	19.81
AIC	10.98	10.13	10.99	23.81
AICc	9.64	8.46	9.66	22.14
BIC	11.57	10.52	11.58	24.20

The upper rows show characteristics of the models compared. For each criterion (weighted sum of squared residuals (SSR), Akaike information criteria (AIC and AICc) and Schwarz criterion (BIC)), the selected model variant is marked in bold.

sufficient number of samples, all statistical properties of the model can be estimated to arbitrary accuracy. By considering many possible models and weighting them by their probabilities, we can obtain reliable probabilities for structural model features even if the model cannot be precisely determined in all its details. As a criterion for model selection, we can compare two models by the ratio of their posterior probabilities, given all data. With equal priors $p(\mathcal{M}_1) = p(\mathcal{M}_2)$ for models \mathcal{M}_1 and \mathcal{M}_2 , this so-called *Bayes' factor* reads

$$\frac{p(\mathcal{M}_2|\mathbf{y})}{p(\mathcal{M}_1|\mathbf{y})} = \frac{p(\mathbf{y}|\mathcal{M}_2)}{p(\mathbf{y}|\mathcal{M}_1)}. \quad (6.33)$$

Unlike the likelihood ratio, the Bayes factor does not score models based on a single optimized parameter set; instead, it is computed from weighted averages over all possible parameter vectors $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$:

$$\frac{p(\mathbf{y}|\mathcal{M}_2)}{p(\mathbf{y}|\mathcal{M}_1)} = \frac{\int p(\mathbf{y}|\boldsymbol{\theta}_2, \mathcal{M}_2)p(\boldsymbol{\theta}_2|\mathcal{M}_2)d\boldsymbol{\theta}_2}{\int p(\mathbf{y}|\boldsymbol{\theta}_1, \mathcal{M}_1)p(\boldsymbol{\theta}_1|\mathcal{M}_1)d\boldsymbol{\theta}_1}. \quad (6.34)$$

For complicated models, these integrals can be approximated by Monte Carlo sampling.

Our choice of priors (for model structures and parameter values) can strongly affect the posterior distribution. The choice of the prior can reflect both our expectations about biological facts and our demands for simplicity. In any case, Bayesian statistics forces modelers to explicitly state their assumptions about system structure. With a uniform prior (i.e., using the same prior probability for all models), the posterior will be proportional to the likelihood. Priors can also be used, like the above-mentioned selection criteria, to penalize models with many parameters.

Pragmatically, the prior probability can also be used as a regularization term. When the data are not sufficient to identify parameter sets or model structures, model selection becomes an ill-posed problem; however, if we employ a prior, the posterior may have, at least, a unique optimum point. If we select this single model – which is, in fact, a rather “non-Bayesian” use of the posterior – the problem of model selection becomes well determined.

6.3 Model Reduction

Summary

Model reduction is the attempt to simplify complex models, that is, to reduce the number of equations and parameters while maintaining a model's key dynamical properties. Elements can be omitted, lumped, or replaced by effective variables, and global model behavior can be approximated by global modes or simplified black-box

models. A reduced model should emulate the behavior of relevant variables under relevant conditions and on a relevant time scale. Common methods for model reduction, like the quasi-equilibrium or quasi-steady-state approximation, can be justified by a distinction between fast and slow processes. Simplified models can facilitate understanding, numerical and analytical calculations, and model fitting.

6.3.1 Model Simplification

In modeling, we must find a good compromise between biological complexity and the simplicity needed for understanding and practical use. The choice of models depends on both the data and biological knowledge and the questions to be addressed. Smaller models – with fewer equations and parameters – provide some advantages: They are better to understand, easier to simulate, and make model fitting more reliable. It is always good to follow A. Einstein's advice: “Make everything as simple as possible, but not simpler.”

Some systematic ways to simplify models [52] are shown in Figure 6.12. A first rule is to omit all elements that have little influence on model predictions, for instance, reactions with negligible rates. Some elements that cannot be omitted may be described, for the conditions or time scales of interest, by effective variables. For instance, we may assume constant concentrations or flux ratios, use effective rate laws fitted to measured data, or linearize nonlinear kinetics for usage in the physiological range. All such simplifications need to be justified: A reduced model should yield a good approximation of the original model for certain quantities of interest, a certain time scale, and certain conditions (for a range of parameter values, in the vicinity of a certain steady state, or regarding some qualitative behavior under study).

Model simplification facilitates model building and simulation because it reduces the number of equations, variables, and parameters, replaces differential equations by algebraic equations, and can remove stiff differential equations. But apart from being a practical tool, model reduction also brings up fundamental questions about what models are and how they can represent reality. The same simplifications that we apply to existing models, in model reduction, can also be directly employed as model assumptions. In fact, any model can be seen as a simplified form of more complex, hypothetical models, which, although faithfully representing reality, would be intractable in practice. Accordingly, the methods for model reduction discussed in this chapter can also be read as a justification of model assumptions in general and, thus, of our mental pictures of reality.

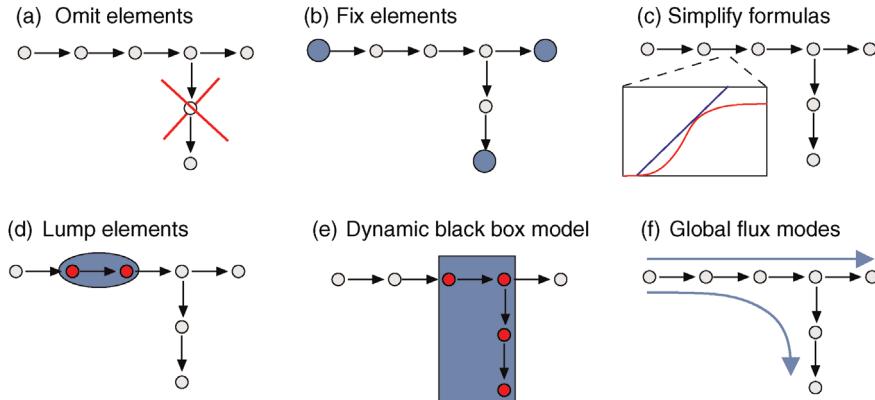


Figure 6.12 Simplification of biochemical models. The scheme shows different types of model reduction in a branched pathway (metabolites shown as circles). (a) Omitting substances and reactions. (b) Predefining concentration or flux values or relations between them. (c) Simplifying mathematical expressions (e.g., omitting terms in a rate law, using simplified rate laws [53], neglecting insensitive parameters [54]). (d) Lumping substances (e.g., similar metabolites, protonation states of a metabolite, or metabolite concentrations in different compartments). Likewise, subsequent reactions in a pathway or elementary steps in a reaction can be replaced by a single reaction of the same velocity; for parallel reactions, for example, reactions catalyzed by isoenzymes, the velocities are summed up. (e) Replacing parts of a model by a dynamic black-box model that mimics the input–output behavior [55]. (f) Describing dynamic behavior by global modes (e.g., elementary flux modes or eigenvectors of the Jacobian matrix).

Another important lesson can be learned from model reduction. If we omit all unnecessary details from a model, the resulting model will describe our biological systems in two ways: in a positive way, by the choice of processes described and the details given in the model; and in a negative way, by the choice of processes omitted, simplified, or treated as constant. The positive facts about the system are those stated explicitly; the negative ones – which may be just as important, and possibly much harder to prove when a model is built – are those implicit in the model assumptions.

6.3.2 Reduction of Fast Processes

Cell processes occur on time scales from microseconds to hours. The time scales of enzymatic reactions, for instance, can differ strongly due to different enzyme concentrations and kinetic constants. Time scales in biochemical systems leave their mark on both the system's internal dynamics (e.g., relaxation to steady state, periodic oscillations) and their susceptibility to external fluctuations. If processes occupy very different time scales, the number of differential equations can be reduced with acceptable errors: Slow processes are approximately treated as constant and fast processes can be effectively described by time averages or fixed relationships between quantities, for example, a quasi-equilibrium between metabolites or quasi-steady-state value for an individual metabolite. For instance, in a model of gene expression, binding and unbinding of transcription factors can occur

on the order of microseconds, changes in transcription factor activity on the order of minutes, and culture conditions may change much more slowly. In the model, we may assume a fast equilibrium for transcription factor binding, a dynamical model for signal transduction and gene expression, and constant values for the culture conditions.

6.3.2.1 Time Scale Separation

In numerical simulations, a fast process in a model, for example, a rapid conversion between two substances $A \rightleftharpoons B$ as shown in Figure 6.13b, can force numerical solvers to use very small time steps during integration. If the same model also contains slow processes, a long total time interval may have to be simulated and the numerical effort can become enormous. Luckily, the dynamics of fast reactions can often be neglected: The concentration ratio s_B/s_A will remain close to the equilibrium constant and if we assume an exact equilibrium in every moment in time, we can replace the reaction by the algebraic relation $s_B/s_A = K_{eq}$, thereby omitting the stiff differential equation that caused the big numerical effort.

A justification for such an algebraic equation is shown in Figure 6.13: In state space, fast processes may bring the system state rapidly close to a submanifold on which certain relationships hold (e.g., an equilibrium between different concentrations). After a short relaxation phase, the system state remains close to this manifold and changes slowly. In our approximation, the system moves exactly on the manifold. In general, there can be a hierarchy of such manifolds related to different time scales [56].

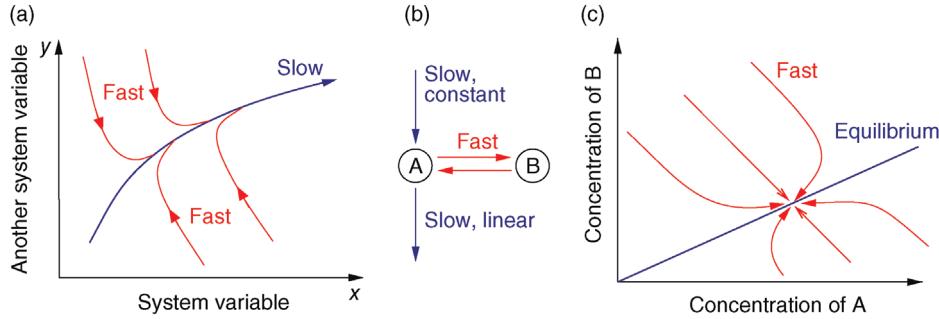
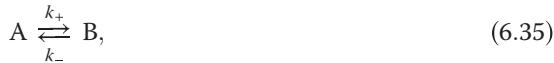


Figure 6.13 Time scale separation. (a) A system's dynamics can be depicted by its trajectories in state space. If the system state is attracted by a submanifold (in the two-dimensional case, a curve), all trajectories (red) will rapidly approach this manifold (blue). Later, the system moves slowly along the manifold, thus satisfying an algebraic equation. (b) A small reaction system with different time scales. A fast conversion between metabolites A and B keeps their concentration ratio s_B/s_A close to the equilibrium constant K_{eq} , while slow production and degradation of A only change the sum $s_A + s_B$. (c) Schematic trajectories for the system shown in part (b). For any initial condition, the concentrations s_A and s_B rapidly approach the line $s_B/s_A = K_{\text{eq}}$ and then move slowly toward the steady state.

6.3.2.2 Relaxation Time and Other Characteristic Time Scales

The time scales of biochemical processes can differ strongly depending on the rate constants involved. To capture them by numbers, we can define characteristic times for different kinds of processes. Such time constants typically describe how fast systems return to equilibrium or steady states after a perturbation. Let us start with the return to equilibrium. As an example, we consider an isolated first-order reaction:



with rate constants k_{\pm} and equilibrium concentrations satisfying $b^{\text{eq}}/a^{\text{eq}} = K_{\text{eq}} = k_+/k_-$. If the concentrations $a = a^{\text{eq}} + x$ and $b = b^{\text{eq}} - x$ are out of equilibrium (e.g., after a small shift in temperature), the deviation x will decrease in time as

$$\frac{d(b^{\text{eq}} - x)}{dt} = k_+(a^{\text{eq}} + x) - k_-(b^{\text{eq}} - x), \quad (6.36)$$

which results in

$$\frac{dx}{dt} = -(k_+ + k_-)x. \quad (6.37)$$

Integration leads to

$$x(t) = x(0) e^{-(k_+ + k_-)t} = x(0) e^{-t/\tau}, \quad (6.38)$$

with a *relaxation time* $\tau = (k_+ + k_-)^{-1}$ for the decrease of x from its initial value to the 1/e-fold value. Accordingly, we can define $\tau_{1/2} = (\ln 2)\tau$ for a decrease to 1/2 of the initial value. Similarly, we can observe how systems relax to steady state. As an example, consider a substance

produced at a rate v and linearly degraded with rate constant λ ; its concentration s satisfies the rate equation:

$$\frac{ds(t)}{dt} = v(t) - \lambda s(t). \quad (6.39)$$

If the production rate v is constant, the concentration s relaxes from an initial value $s(0) = s_0$ to the steady-state value $s^{\text{st}} = v/\lambda$ as

$$s(t) = s^{\text{st}} + (s_0 - s^{\text{st}}) e^{-\lambda t}. \quad (6.40)$$

Again, the response time $\tau = 1/\lambda$ is the time after which the initial deviation $\Delta s(t) = s(t) - s^{\text{st}}$ has decreased by a factor 1/e, and response half time $\tau_{(1/2)} = (\ln 2)/\lambda$ is defined accordingly. A more general response time for single reactions is defined as [57]:

$$\tau_j = \left(\sum_i \frac{\partial v_j}{\partial s_i} n_{ij} \right)^{-1}. \quad (6.41)$$

This definition also applies to reactions with several substrates or products and with nonlinear rate expressions. Time constants for metabolite concentrations can be defined in different ways. Reich and Sel'kov [58] defined the *turnover time* as the time that is necessary to convert a metabolite pool once:

$$\tau_i^{\text{turn}} = \frac{s_i}{\sum_{j=1}^r (n_{ij}^- v_j^+ + n_{ij}^+ v_j^-)}. \quad (6.42)$$

Every reaction is split into partial forward (v_j^+) and reverse (v_j^-) reactions. Accordingly, stoichiometric coefficients are assigned to individual reaction directions (n_{ij}^+, n_{ij}^-). To define a transition time for entire pathways, Easterby [59] considered a pathway in which enzymes,

but no metabolites, are initially present. The transition time after addition of substrate describes how fast intermediate pools reach their steady-state concentrations. For each intermediate, $\tau_i = s_i^{\text{st}}/J$, where s_i^{st} and J denote concentration and flux, respectively, in the final steady state. The transition time of the pathway is the sum $\tau = \sum_{i=1}^n \tau_i$ of all these transition times. Another time constant, as defined by Heinrich and Rapoport, measures the time necessary to return to a steady state after a small perturbation [60]. Let $\Delta s(t) = s(t) - s^{\text{st}}$ be the deviation from steady-state concentrations s^{st} . Then the transition time is defined as

$$\tau = \frac{\int_0^\infty t \Delta s(t) dt}{\int_0^\infty \Delta s(t) dt}. \quad (6.43)$$

This definition applies only if $\Delta s(t)$ vanishes asymptotically for large t . Lloréns *et al.* [61] have generalized this definition. Let y denote an output quantity, for example, a flux or concentration, that is analyzed after a perturbation. Using the absolute derivative $|dy/dx|$ as a probability weight, a characteristic time can be calculated as

$$T = \frac{\int_0^\infty t |dy/dx| dt}{\int_0^\infty |dy/dx| dt}. \quad (6.44)$$

This definition applies even to damped oscillations after a perturbation.

6.3.3 Quasi-Equilibrium and Quasi-Steady State

In systems with several different time scales, various kinds of simplifications are possible: (i) In molecular dynamics, we may focus on slow changes of a thermodynamic ensemble: Then, fast molecular movements (e.g., fast jittering atom movements versus slow conformation changes in proteins) average out. (ii) If there is fast diffusion, leading to homogeneous concentrations, spatial structures can be neglected. (iii) In the quasi-equilibrium approximation, we assume that the substrates and products of a reaction are practically equilibrated; then their concentration ratio is given, in every moment, by the equilibrium constant. A quasi-equilibrium requires that the equilibrium reaction be reversible and much faster than all other processes considered. An example is the permanent protonation and deprotonation of acids: Since this process is much faster than the usual enzymatic reactions, the protonation states of a substance need not be treated separately, but can be effectively captured by a quasi-species. Another example is transcription factor binding (see Section 10.3), which can be treated as an equilibrium process to derive effective gene regulation functions. (iv) In the quasi-steady-state

Example 6.8 Quasi-Steady-State and Quasi-Equilibrium Approximations

Let us illustrate both approximations, quasi-steady state and quasi-equilibrium, with a simple model of upper glycolysis (see Figure 6.14 and Section 4.2.4). Glucose (GLC) is imported into the cell at a rate of v_0 and subsequently converted into glucose-6-phosphate (G6P), fructose-6-phosphate (F6P), and fructose-1,6-bisphosphate (FBP), which then feeds into lower glycolysis. In our model, the cofactors ATP and ADP have fixed concentrations. With mass-action kinetics and a reversible reaction between G6P and F6P, the rate equations read as follows:

$$\frac{ds_1}{dt} = v_0 - k_1 s_A s_1. \quad (6.45)$$

$$\frac{ds_2}{dt} = k_1 s_A s_1 - k_{+2} s_2 - k_{-2} s_3. \quad (6.46)$$

$$\frac{ds_3}{dt} = k_{+2} s_2 - k_{-2} s_3 - k_3 s_A s_3. \quad (6.47)$$

$$\frac{ds_4}{dt} = k_3 s_A s_3 - k_4 s_4. \quad (6.48)$$

The indices refer to metabolites and reactions in the scheme and s_A denotes the constant ATP concentration. We first assume that all reactions take place on a similar time scale, setting $k_{\pm 2} = 2$ and all other rate constants and the ATP concentration to values of 1 (arbitrary units). Figure 6.14a shows simulated concentration curves of GLC, G6P, F6P, and FBP; the initial concentrations are chosen to be zero. Until time $t = 5$, the influx has a value of $v_0 = 2$ and the intermediate levels rise one after the other. Then, the influx drops to $v_0 = 1$ and the levels decrease again. How would the system behave if the first or the second reaction were very fast? The two scenarios can be approximated, respectively, by a quasi-steady state for glucose or by a quasi-equilibrium between G6P and F6P.

Quasi-steady-state approximation

If k_1 is increased to a value of 5 (Figure 6.14c), glucose is rapidly consumed and its steady-state level remains low; moreover, due to the fast turnover, it adapts almost instantaneously to changes of the input flux. This behavior can be approximated by the quasi-steady-state approximation: We replace the glucose concentration at each time point by the steady-state value $s_1^{\text{st}}(t) = v_0(t)/(k_1 s_A)$ obtained from the current value of $v_0(t)$. This algebraic equation replaces the differential equation (Eq. (6.45)) for s_1 ; formally, we could also obtain it by setting the left-hand side of the differential equation to zero.

Quasi-equilibrium approximation

Next, we consider a rapid and reversible conversion between the hexose phosphates G6P and F6P. To do so, we increase both rate constants by the same factor ($k_{+2} = 10$ and $k_{-2} = 5$ in Figure 6.14d), thus keeping their ratio $K_{\text{eq}} = k_{+2}/k_{-2}$ fixed. In the simulation, the ratio of F6P to G6P levels rapidly approaches the equilibrium constant $[F6P]/[G6P] = s_3/s_2 = K_{\text{eq}}$. In a quasi-equilibrium approximation, we force them to show this ratio in every moment. By adding Eqs. (6.46) and (6.47), we obtain the following equation:

$$\frac{ds_{2+3}}{dt} = \frac{d(s_2 + s_3)}{dt} = k_1 s_1 s_1 - k_2 s_1 s_3. \quad (6.49)$$

Given s_{2+3} and K_{eq} , we can now substitute $s_3 = s_{2+3}K_{\text{eq}}/(1 + K_{\text{eq}})$ in Eq. (6.48) and obtain a simplified differential equation system in which the fast reaction has disappeared. The two differential equations for s_2 or s_3 are replaced by the single differential equation (Eq. (6.49)) (for the sum of the two variables) and an algebraic equation $s_3/s_2 = K_{\text{eq}}$ for the concentration ratio.

approximation, we assume that production and consumption of a substance are balanced. If the consuming reactions are irreversible, the substance concentration will be directly determined by the production rate. Quasi-equilibrium and quasi-steady-state approximations can be used, for instance, to derive the Michaelis–Menten rate law from an enzyme mechanism with elementary mass-action steps (see Section 4.1).

6.3.4 Global Model Reduction

If a system is constrained to a submanifold in state space (as shown in Figure 6.13a), its movement on this manifold can be described by a smaller set of variables. Constraints can arise, for instance, from linear conservation relations between metabolite concentrations (see Section 3.1.4): If rows of the stoichiometric matrix \mathbf{N} are linearly dependent, the vector of metabolite concentrations is confined to a linear subspace and the system state can be described by independent metabolite concentrations from which all other concentrations can be directly computed. Other constraints can arise from fast processes, like in the quasi-steady-state and quasi-equilibrium approximation. Unlike the original concentrations, effective variables need not refer to individual substances: A linear manifold can be spanned by linear combinations of all substance concentrations, representing global modes of the system's dynamics. Such modes can appear, for instance, in metabolic systems linearized around a steady state: Each mode represents a specific pattern of metabolite levels (actually, their deviations from steady state) with a specific temporal dynamics (e.g., exponential relaxation).

6.3.4.1 Linearization of Biochemical Models

Linear control theory is concerned with dynamic models of the general form:

$$\begin{aligned} \frac{dx}{dt} &= Ax + Bu, \\ y(t) &= Cx + Du, \end{aligned} \quad (6.50)$$

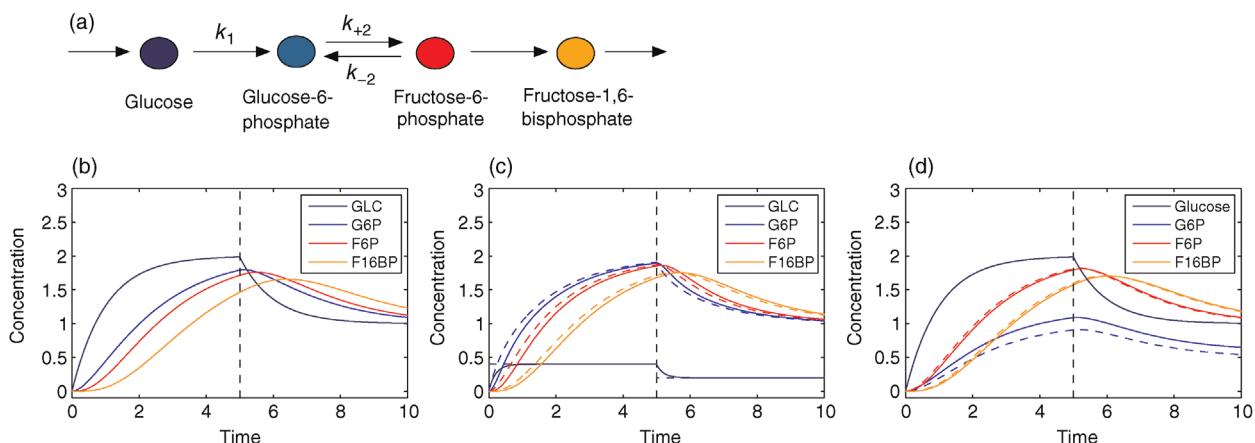


Figure 6.14 Simulation results for a model of upper glycolysis. (a) Metabolites and reactions in the model. (b) Results from the original model, showing levels of GLC, G6P, F6P, and FBP (time and concentrations measured in arbitrary units). (c) Results from the model with fast glucose turnover $k_1 = 5$ (solid lines) and the quasi-steady-state approximation (broken lines). (d) Results from the model with fast reversible conversion $G6P \leftrightarrow F6P$ (solid lines), parameters $k_{+2} = 10$, $k_{-2} = 5$, and the quasi-equilibrium approximation (broken lines).

with vectors of input variables $\mathbf{u}(t)$, internal variables \mathbf{x} , and observable outputs $\mathbf{y}(t)$ (see Section 15.5). In practice, the output variables in \mathbf{y} represent variables that can be observed, that affect other systems, or that determine the objective in some optimal control problem. Biochemical models are usually nonlinear, but many of them can be linearized around a steady state. Consider a kinetic model:

$$\frac{ds}{dt} = N\mathbf{v}(\mathbf{s}, \mathbf{p}), \quad (6.51)$$

with concentration vector \mathbf{s} , stoichiometric matrix N , reaction rate vector \mathbf{v} , and parameter vector \mathbf{p} . We assume that for given parameter sets \mathbf{p} , the system shows a stable steady state $\mathbf{s}^{st}(\mathbf{p})$. To linearize Eq. (6.51), we determine the steady state $\mathbf{s}_0^{st} = \mathbf{s}^{st}(\mathbf{p}_0)$ at a reference parameter vector \mathbf{p}_0 and compute the elasticity matrices $\tilde{\epsilon} = \partial\mathbf{v}/\partial\mathbf{s}$ and $\tilde{\pi} = \partial\mathbf{v}/\partial\mathbf{p}$ (see Section 4.2). For small deviations of concentrations $\mathbf{x}(t) = \mathbf{s}(t) - \mathbf{s}_0^{st}$ and parameters $\mathbf{u}(t) = \mathbf{p}(t) - \mathbf{p}_0$, linearizing Eq. (6.51) leads to

$$\frac{dx(t)}{dt} = A\mathbf{x}(t) + B\mathbf{u}(t), \quad (6.52)$$

with the Jacobian matrix $A = N\tilde{\epsilon}$ and the matrix $B = N\tilde{\pi}$. In general, the approximation (6.52) is valid only near the expansion point, and its accuracy decreases for larger deviations \mathbf{u} or \mathbf{x} . Moreover, linearized models may not be able to reproduce certain kinds of dynamic behavior, for example, stable limit cycles.

Closely related to this linear approximation, we can consider the input–output relation $\mathbf{y}(\mathbf{u})$ between static parameter deviations \mathbf{u} and steady output variables \mathbf{y} such as fluxes or concentrations. For instance, consider a stable system under small parameter perturbations: If the perturbations are slow enough, the entire system will follow them in a quasi-steady state. Therefore, the system's input–output relation – even for complicated systems – can be described by a function and approximated by simple functions with fitted parameters. After linearization, this becomes relatively simple: By setting the time derivative in Eq. (6.52) to zero, assuming a small static perturbation \mathbf{u} , solving for \mathbf{x} , and computing \mathbf{y} , we obtain $\mathbf{y} = (-CA^{-1}\mathbf{B} + \mathbf{D})\mathbf{u}$, which is nothing else but the linear response:

$$\mathbf{y} \approx \tilde{\mathbf{R}}_{\mathbf{p}}^{\mathbf{y}} \Delta \mathbf{u}, \quad (6.53)$$

with response matrix $\tilde{\mathbf{R}}_{\mathbf{p}}^{\mathbf{y}}$, well known from metabolic control analysis. The effects of oscillating parameter perturbations [62,63], as well as the effects of stationary perturbations on transient behavior [64], can be treated accordingly (see Sections 10.1.5 and 15.5).

6.3.4.2 Linear Relaxation Modes

In the linearized system (Eq. (6.52)), the general model behavior can be represented by a superposition of global modes z_j , each corresponding to an eigenvector of A (see Section 15.2). For constant system parameters ($\mathbf{u} = 0$), a small deviation $\mathbf{x} = \mathbf{s} - \mathbf{s}_0^{st}$ will follow

$$\frac{dx}{dt} = Ax. \quad (6.54)$$

Now we assume that the Jacobian is diagonalizable, $A = Q\Lambda Q^{-1}$ with a diagonal matrix $\Lambda = Dg(\lambda_i)$ and a transformation matrix $Q = \{q_{ji}\}$, and that all eigenvalues λ_i have negative (or, at least, vanishing) real parts. We introduce the transformed vector $\mathbf{z} = Q^{-1}\mathbf{x}$, which follows the equation

$$\frac{dz}{dt} = \Lambda z. \quad (6.55)$$

Thus, whenever A is diagonalizable, we obtain an equation

$$\frac{d}{dt} z_j = \lambda_j z_j \quad (6.56)$$

for each global mode z_j . The behavior of the original variables x_i can be written as

$$x_i(t) = \sum_j q_{ij} z_j(t) = \sum_j q_{ij} z_j(0) e^{-\lambda_i t}, \quad (6.57)$$

with initial value $\mathbf{z}(0) = Q^{-1}\mathbf{x}(0)$. Each mode is characterized by a response time similar to Eq. (6.40). If the eigenvalue λ_i is a real number, z_j relaxes exponentially to a value of 0, with a response time $\tau_j = 1/\lambda_j$. A pair of complex conjugate eigenvalues, in contrast, leads to a pair of oscillatory modes with time constant $\tau_i = 1/\text{Re}(\lambda_i)$. An eigenvalue $\lambda_i = 0$ (corresponding to infinitely slow modes) can arise, for instance, from linear conservation relations. Modes in biochemical dynamics are comparable to the harmonics of a guitar string. Each of the harmonics displays a characteristic spatial pattern and, as a simple temporal behavior, a sine wave modulation of its amplitude. All possible movements of the string, at least for small amplitudes that justify our linear approximation, can be obtained by linear superposition of these modes.

6.3.4.3 Model Reduction

In model reduction, we assume that fast modes (with small τ_j) in the sum (6.57) relax immediately. If we neglect these modes, the accuracy in describing rapid changes is reduced, but no metabolite or reaction is omitted from the model. Like in Figure 6.13, the system state is projected to a space of slow modes. Even in cases where A cannot be diagonalized, the state space can be

split into subspaces related to fast and slow dynamics, and by neglecting the fast subspace, the number of variables can be reduced. This method can be applied adaptively during computer simulations [65]. Powerful methods for linear model reduction, such as balanced truncation [66,55], have been developed in control engineering. Using such methods, we can mimic the external behavior of a complex biochemical model, that is, its responses to perturbations, by linearized, dynamic black-box models with similar input–output relations (see Section 16.5.5). However, these methods operate on the system as a whole and are not directly related to the quasi-steady-state and quasi-equilibrium approximations for single network elements. A model reduction for selected network elements can be achieved by the computational *singular perturbation method* that identifies such elements automatically during a simulation and can be used in the software tool COPASI [67].

6.4 Coupled Systems and Emergent Behavior

Summary

All biological systems, from organisms to molecules, interact with larger environments. A coupling between systems can lead to qualitatively new behavior, which can be studied by models. Kinetic models have a local and causal structure (e.g., locally interacting reactions and metabolite species), and numerically solving them is a way to trace influences across the system and to infer their global dynamics. When building models, we can either start from small elements and combine them to larger units or start from a simplified global picture and refine it progressively. In both cases, interactions between subsystems are central: Whether systems are described in isolation (with fixed or controlled environments) or coupled to dynamic environments can make a big difference for their dynamics. A similar distinction holds for experiments, where conditions like pH or cofactor concentrations can be experimentally controlled *in vitro*, whereas these conditions are dynamic and often difficult to quantitate in living cells.

To understand cells, we need to consider their parts, the interactions between these parts, and the global dynamics that emerges from the interactions. Molecules and biochemical processes have been studied in great detail, and systems biology builds on a wealth of knowledge from biochemistry, molecular biology, and molecular genetics. A basic credo of systems biology says that

analysis (taking apart) and synthesis (putting together) must be combined.

There are two contrary approaches, bottom-up and top-down modeling. Both proceed from simplicity to complexity, but in opposite ways. In *bottom-up* modeling [68,69], one considers elementary processes and aggregates them to larger models. Since there is no guarantee that models will remain valid when combined with others, parameters may have to be refitted after model construction. In *top-down* modeling, a model is based on a coarse-grained picture of the entire system, which can then be iteratively refined. If a model structure is biologically plausible, such models, even though lacking many details, may yield accurate predictions. Bottom-up and top-down approaches pursue different goals: A bottom-up model is constructed to explore how high-level behaviors emerge from the interaction of many molecular components; a top-down model, on the other hand, is designed to accurately predict high-level behavior regardless of molecular correctness. Generally, it is difficult to construct models that are both molecularly detailed and globally correct.

Recently, researchers developed the first whole-cell model [70] of *Mycoplasma genitalium*, a pathogenic bacterium known for its small genome size (525 genes). This constituted a major step forward in modular cell modeling. In this model, a variety of cellular processes – including genome-wide transcription and translation, DNA repair, DNA replication, metabolism, cell growth, and cell division – were represented by a total of 28 coupled submodels. Accomplishing this task involved overcoming several challenges. First, a huge number of cell components were modeled in detail. For example, DNA-binding proteins were described by their precise position on the chromosome, and the specific function of every annotated gene product was represented. Second, the submodels were expressed using multiple mathematical formalisms, including differential equations, stochastic particle simulations, and dynamic flux balance analysis (FBA). These submodels were coupled together using an Euler-type integration scheme in which submodels are repeatedly simulated independently for 1 s intervals and coupled by updating a set of global state variables (see Figure 6.15). To allow processes in different submodels to consume the same shared resources (e.g., ATP, which is involved in many processes described), the available resources are partitioned before each integration step. At each iteration, each submodel obtains a fraction of each resource for potential consumption. The *M. genitalium* model was fitted and validated with various types of biological data and can now be used to simulate numerous biological behaviors, including the movement and collisions of DNA-binding proteins, the global allocation of

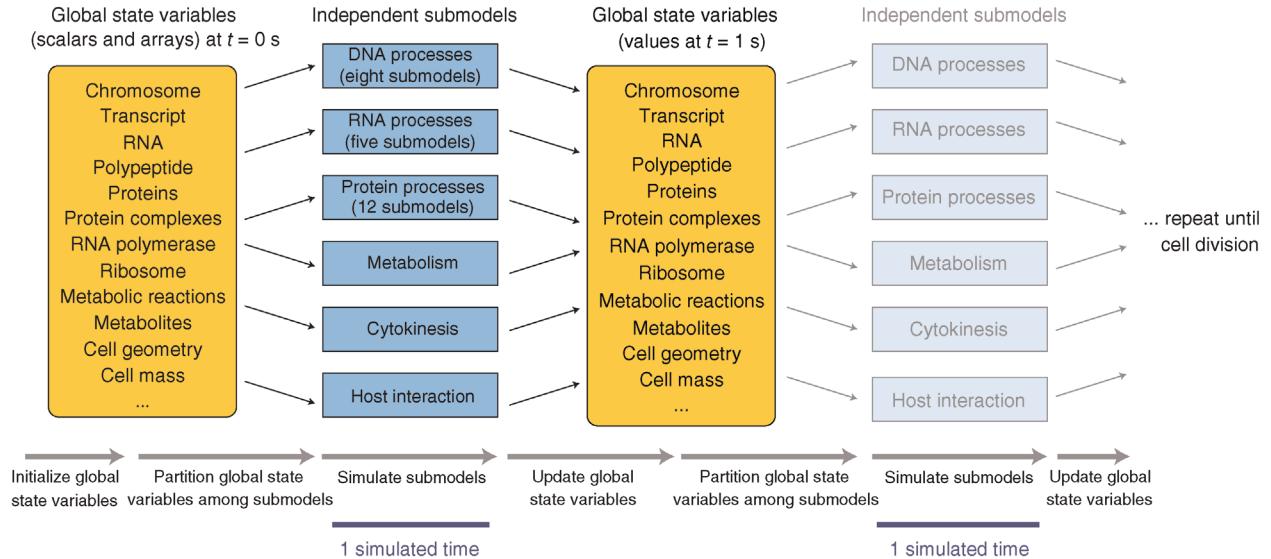


Figure 6.15 Modular whole-cell model of *M. genitalium* [70]. The model consists of 28 submodels (blue boxes), each of which represents a distinct cellular process. Submodels are coupled via global state variables (yellow box), which represent the instantaneous configuration of a single cell. To simulate the life cycle of a single cell, the state variables are first initialized corresponding to the beginning of the cell cycle. Second, the submodels are simulated for 1 s of real (i.e., the cell's) time. Next, the global variables are updated. The submodels are repeatedly simulated and the global variables are repeatedly updated until the *in silico* cell cycle has been completed. The modular model structure allows the model to simultaneously employ multiple mathematical modeling formalisms, including differential equations, particle-based stochastic models, and dynamic FBA.

energy to different functional subsystems (see Section 8.3), and the detailed effects of single-gene disruptions on the growth rate and macromolecule synthesis.

6.4.1 Modeling of Coupled Systems

6.4.1.1 Modeling the System Boundary

When building a model, we separate a *system of interest* (which is explicitly modeled) from its *environment*, which is either not modeled or described very roughly. This distinction is artificial, but hard to avoid. The model is bounded by quantities like external substance levels or fluxes whose values are set as model assumptions. If the boundary variables are fixed, the system is modeled *as if* it were isolated. To justify this assumption at least approximately, one should carefully consider experimental details and pay attention to the choice of boundary variables. The system boundary should be chosen such that interactions between system and environment are weak, constant in time, or average out (because they are fast or random). If boundary variables (concentrations or fluxes) change dynamically, their time courses may be predefined based on experimental data or computed from a separate environment model. If the environment responds dynamically to the system, it needs to be described within the model: either by effective algebraic

relationships [71] or by simplified dynamic black-box models [72].

6.4.1.2 Coupling of Submodels

If a model is composed of submodels, the submodels are connected through variables such as shared metabolite concentrations. As shown in Figure 6.16, a submodel can take metabolite concentrations as inputs and yield reaction rates as an output. These reaction rates in turn contribute to the rate equations of the boundary metabolites. Submodels affect each other only through these interfaces, and if time curves of the communicating variables were fixed and given, each module could be simulated independently. Thus, communicating variables connect submodules and separate them at the same time. One advantage of this modular structure is that submodels can be based on different mathematical formalisms or be simulated in parallel on separate computers. If the influences between modules form an acyclic graph, we can simulate their dynamics sequentially. We begin with the upstream modules, compute their outputs, and use them as inputs for the downstream modules. In contrast, if the coupling involves loops, all modules need to be simulated together, with the communicating variables as shared variables and all other variables as “private” variables of the modules. If global variables are updated in intervals – as in the *Mycoplasma* model in Figure 6.15,

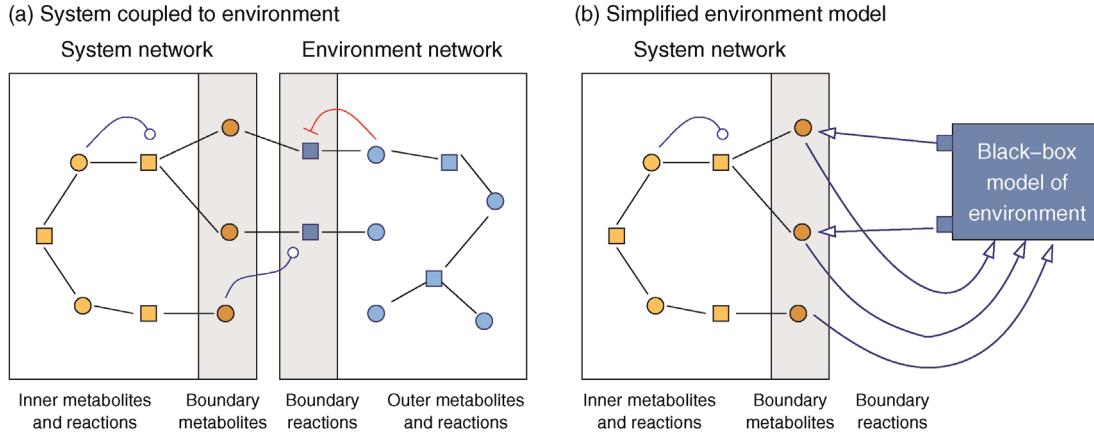


Figure 6.16 Model reduction applied to a schematic biochemical network model. (a) A metabolic system is split into a pathway of interest (left half) and its environment (right half). Both subsystems interact via communicating metabolites (belonging to the pathway of interest) and reactions (belonging to the environment). (b) To simplify numerical simulations, the environment model has been replaced by a reduced black-box model (see Ref. [72]).

then the dynamics of the shared global variables should be slow in relation to the updating intervals.

6.4.1.3 Supply–Demand Analysis

Metabolic pathways need to match input and output fluxes, and thus supply and demand [73,74]. In a metabolic pathway, the steady-state flux depends on the initial substrate concentration. However, if rate laws are reversible or if enzymes are allosterically regulated, the end product will also exert control on the flux. Supply–demand analysis [73] splits metabolism into blocks that are coupled by matching their supply and demand variables at the boundaries. The elasticities of supply and demand reactions, which can be measured for the individual blocks, are then used to describe the behavior, control, and regulation of metabolism.

6.4.1.4 Hierarchical Regulation Analysis

An important question, one that concerns the interplay of metabolic systems and the underlying transcriptional or posttranslational regulation of enzymes, is whether flux changes are mainly caused by changes in metabolite levels or by enzymatic changes; and, in turn, whether these enzymatic changes are mainly caused by expression changes or by posttranslational modification. This varies from case to case and has consequences for time scales (posttranslational modification being much faster than expression changes) and cellular economics (different underlying changes will be associated with different costs and benefits). In metabolic control analysis, one would address such questions by considering an initial perturbation, for example, an expression change, tracing its effects in the system, and predicting the resulting flux change. This, however, would require a comprehensive model of

the system. A much simpler procedure is provided by hierarchical regulation analysis [75]. Here one starts from the effect – namely, the flux changes – and traces back to its possible causes. For the reaction in question, we assume a rate law of the following form:

$$\nu = f(E)g(\mathbf{s}, \mathbf{p}, \mathbf{x}), \quad (6.58)$$

with rate ν , enzyme level E , substrate levels \mathbf{s} , product levels \mathbf{p} , and effector levels \mathbf{x} . The maximal velocity f is typically given by $f(E) = k_{\text{cat}}E$. Now we consider a change in conditions, where E , \mathbf{s} , \mathbf{p} , and \mathbf{x} may change at the same time. The resulting change in logarithmic reaction rate (where a positive flux is assumed without loss of generality) reads

$$\Delta \log \nu = \Delta \log f(E) + \Delta \log g(\mathbf{s}, \mathbf{p}, \mathbf{x}), \quad (6.59)$$

which implies

$$1 = \frac{\Delta \log f(E)}{\Delta \log \nu} + \frac{\Delta \log g(\mathbf{s}, \mathbf{p}, \mathbf{x})}{\Delta \log \nu} = \rho_h + \rho_m. \quad (6.60)$$

The hierarchical regulation coefficient ρ_h describes which fraction of the log-rate change is caused by changes in maximal velocity, and thus by changes in the gene expression cascade. Given a hierarchical model of expression (including transcription, translation, and post-translational modification), the coefficient can further be split into contributions of these individual processes. The metabolic regulation coefficient ρ_m describes the fraction of log-rate change caused by changes in metabolite levels, capturing the enzyme's interaction with the rest of metabolism. Regulation coefficients can be directly determined from flux and expression data and can yield direct information about regulation without the need for complex quantitative models.

6.4.2

Combining Rate Laws into Models

Once models for several reactions, pathways, or cellular subsystems have been built, they can be combined to form more complex models. One may even start from individual reaction kinetics measured *in vitro*: Pioneering this modeling approach, Teusink *et al.* [68] built a model of yeast glycolysis from *in vitro* rate laws. In general, *in vitro* measurements allow for an exact characterization and manipulation of enzyme parameters, but they will not reflect the precise biochemical conditions – such as pH or ion concentrations – existing in living cells. Nevertheless, the yeast glycolysis model, without further adaptation, yielded a plausible steady state.

In this way, metabolic network models can be constructed by inserting collected rate laws into known network structures. In theory, a network with correct rate laws should yield a consistent model; in reality, however, such models, for given external metabolite levels and enzyme levels, may show unrealistic states or no steady state at all because, for instance, the parameters obtained from different experiments simply do not fit. In particular, thermodynamic correctness may not be ensured because equilibrium constants do not fit together. This can be tested by simulating the model with all metabolites and cofactors treated as internal. In a thermodynamically correct model, this simulation should lead to an equilibrium state with vanishing fluxes. However, if no attention has been paid to consistent equilibrium constants, the simulation may instead result in steady fluxes, contradicting the laws of thermodynamics and representing a *perpetuum mobile*.

To avoid these issues, Stanford *et al.* [76] proposed a systematic way to construct models with reversible rate laws, using the network structure as a scaffold. The model variables are determined step-by-step: (i) Choose the intended flux distribution (which must be thermodynamically feasible, but need not be stationary). (ii) Choose equilibrium constants satisfying the Wegscheider conditions. (iii) Choose metabolite concentrations such that the thermodynamic forces (i.e., negative reaction Gibbs free energies) follow the flux directions (see Section 16.6). (iv) Choose reversible rate laws (e.g., modular rate laws) with rate constants satisfying the Haldane relationships (e.g., by parameter balancing). (v) Set all enzyme levels to 1 and compute the resulting reaction rates; by construction, the rates will have the same signs as the predefined fluxes. (vi) Adjust the enzyme levels (or generally V^{\max} values) such that reaction rates match the previously defined fluxes. An open problem in such models is how to choose suitable rate laws for biomass production; a possibility would be to use rate laws for

polymerization reactions [77], treating biomass – which actually comprises protein, polynucleotides, lipids, and many other substances – as a single hypothetical polymer, produced after a template sequence.

6.4.3

Modular Response Analysis

When dynamic systems are coupled, be they biological systems or mathematical models describing them, mutual interactions can drastically change the dynamic behavior. Metabolic control analysis allows us to study the global behavior emerging from many coupled reactions. Modular response analysis [78,79], one of its variants, addresses the coupling of larger modules, for example, interacting signaling pathways, and predicts emergent global behavior from the effective input–output behavior of single modules. In modular response analysis (see Figure 6.17a), a communicating variable is an output of one module that acts as a parameter in other modules. All other system parameters are collected in the external parameter vector \mathbf{p} , and modules are not connected by fluxes.

To calculate response coefficients for the entire system, we first consider the single modules and compute their local response coefficients, assuming that the communicating variables are clamped and the modules are isolated from each other. The dynamics of a module μ depends, in general, on the external parameter vector \mathbf{p} and on the output vectors $\mathbf{x}_a, \mathbf{x}_b, \dots$ of the other modules. Its steady-state output \mathbf{s}_μ can be written as a function $\mathbf{s}_\mu(\mathbf{p}, \mathbf{x}_a, \mathbf{x}_b, \dots)$, where the argument list contains all module outputs except for \mathbf{x}_μ itself (see Figure 6.17b). Next, we consider the modules coupled to one another. We assume that the coupled system reaches a stable steady state in which the values \mathbf{y} of the output variables satisfy

$$\begin{aligned} \mathbf{y}_a(\mathbf{p}) &= \mathbf{s}_a(\mathbf{p}, \mathbf{y}_b(\mathbf{p}), \mathbf{y}_c(\mathbf{p}), \dots), \\ \mathbf{y}_b(\mathbf{p}) &= \mathbf{s}_b(\mathbf{p}, \mathbf{y}_a(\mathbf{p}), \mathbf{y}_c(\mathbf{p}), \dots), \end{aligned} \quad (6.61)$$

and so on. The modules' *local response coefficients* in this steady state are defined by

$$\tilde{\mathbf{R}}_p^{S_\mu} = \frac{\partial \mathbf{s}_\mu}{\partial \mathbf{p}}, \quad \tilde{\mathbf{R}}_{S_\nu}^{S_\mu} = \frac{\partial \mathbf{s}_\mu}{\partial \mathbf{x}_\nu} \Big|_{\mathbf{x}=\mathbf{y}}, \quad \tilde{\mathbf{R}}_{S_\mu}^{S_\mu} = 0. \quad (6.62)$$

The *global response coefficients* – the sensitivities of the coupled system to small external parameter changes – are defined as

$$\tilde{\mathbf{R}}_p^{Y_\mu} = \frac{\partial \mathbf{y}_\mu}{\partial \mathbf{p}}. \quad (6.63)$$

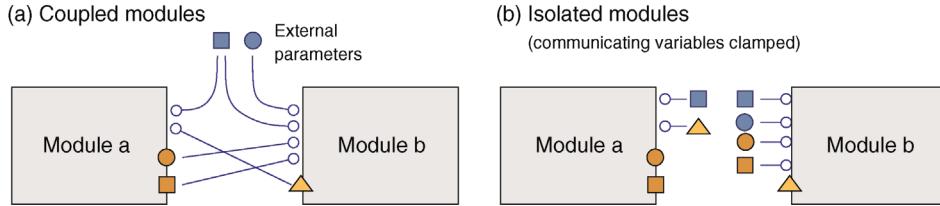


Figure 6.17 Modular response analysis. (a) Two network modules, described by kinetic models, interact through communicating variables. (b) To study the behavior of isolated modules, the communicating variables are treated as controllable parameters. A sensitivity analysis yields the local response coefficients. These coefficients, together with the wiring scheme of the communicating variables in part (a), allow us to compute the global response coefficients for scenario (a).

If we collect the response matrices in large block matrices:

$$\tilde{\mathbf{R}}_S^S = \begin{pmatrix} \tilde{\mathbf{R}}_{S_a}^{S_a} & \tilde{\mathbf{R}}_{S_a}^{S_b} & \vdots \\ \tilde{\mathbf{R}}_{S_b}^{S_a} & \tilde{\mathbf{R}}_{S_b}^{S_b} & \vdots \\ \dots & \dots & \dots \end{pmatrix},$$

$$\tilde{\mathbf{R}}_p^S = \begin{pmatrix} \tilde{\mathbf{R}}_p^{S_a} \\ \tilde{\mathbf{R}}_p^{S_b} \\ \dots \end{pmatrix}, \quad \tilde{\mathbf{R}}_p^Y = \begin{pmatrix} \tilde{\mathbf{R}}_p^{Y_a} \\ \tilde{\mathbf{R}}_p^{Y_b} \\ \dots \end{pmatrix}, \quad (6.64)$$

the global response coefficients can be computed from the local ones by

$$\tilde{\mathbf{R}}_p^Y = -(\tilde{\mathbf{R}}_S^S - I)^{-1} \tilde{\mathbf{R}}_p^S. \quad (6.65)$$

Thus, to compute global steady-state responses to small perturbations, the internal details of the modules need not be known. We only need to know the local response coefficients, that is, the modules' effective input–output behavior for the communicating variables. Modular response analysis only applies to steady-state perturbations. To treat dynamic small-amplitude perturbations instead, the submodels could be approximated by dynamical black-box models obtained by linear model reduction (see Figure 6.16) [72].

6.4.4

Emergent Behavior in Coupled Systems

In Section 5.4, we saw how biochemical models can be formally combined and which technical and conceptual problems may arise on the way. However, if we couple two models, how will this affect their dynamic behavior? If models influence each other sequentially, their overall dynamics can be simulated step-by-step. However, if models interact in both ways or in a circle, this may change their qualitative behavior. The following examples illustrate how new behavior can emerge in coupled systems.

The two possible descriptions – treating systems as isolated, or treating them as dynamically coupled – are characteristic of two general perspectives on complex

systems. In *reductionist* approaches, one studies the parts of a system in isolation and in great detail. This view is dominant in molecular biology and biochemistry. A *holistic* perspective, in contrast, focuses on global dynamic behavior that emerges in the coupled system. Instead of tracing causal chains across the network, it emphasizes how the system dynamics, as a whole, responds to changes of external conditions. In the examples above, we saw that fundamental notions for describing biochemical dynamics (bistable system; steady-state flux) rely on holistic explanations.

Whether behavior is caused locally or globally can be hard to decide: Yeast cells, for instance, communicate by exchanging chemicals, and this interaction can lead to synchronized glycolytic oscillations. Spontaneous oscillations of this sort are observed in experiments and

Example 6.9 Bistable Switch

Two genes X and Y that mutually inhibit each other can form a genetic switch (Figure 6.18). We describe their levels x and y by a differential equation model:

$$\frac{dx}{dt} = f(x, y),$$

$$\frac{dy}{dt} = g(x, y). \quad (6.66)$$

By setting the second equation to zero and solving for y , we obtain the steady-state value of y as a function of x . The curve $y^{st}(x)$ in Figure 6.18a is called the *nullcline* of y . Likewise, we obtain the nullcline $x^{st}(y)$ from the first equation. The nullclines represent response curves for the individual systems. When both systems are coupled, both steady-state requirements $y^{st} = f(x^{st})$ and $x^{st} = g(y^{st})$ must be satisfied at the same time. We obtain three fixed points, two of which are stable as indicated by the slopes of the nullclines. Due to the positive feedback, a bistable switch emerges. Bistability is not a property of the individual genes X and Y: It is a systemic property that only arises from their coupling (for an biological example, see Figure 2.2).

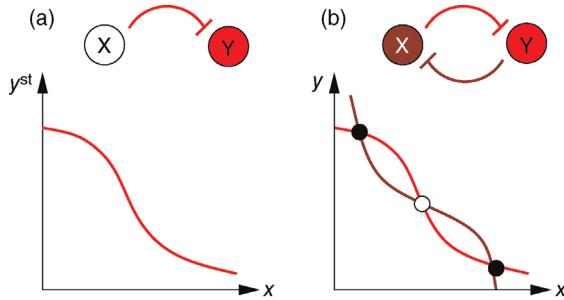


Figure 6.18 Mutual inhibition can lead to bistability. (a) A gene (level y) is inhibited by another gene (level x), which acts as a regulatory input. The steady-state level y^{st} (black) depends on the given value of x . (b) If both genes inhibit each other, the coupled system becomes bistable: There are two stable fixed points (black dots) and one unstable fixed point (white dot) at the intersection of the two nullclines (for a biological example, see Figure 2.2).

models [80,81,82]. Experiments with desynchronized populations suggest that individual yeast cells oscillate by themselves, and that the main effect of their coupling is to synchronize preexisting oscillations, which would otherwise go unnoticed in most experiments. Thus, the overall oscillations rely on a local dynamics (namely, the oscillations in individual cells), but only the coupling between cells enables them to emerge as a global phenomenon.

6.4.5 Causal Interactions and Global Behavior

What are the roles of reductionism and holism in biochemical modeling? We can see this by looking at how kinetic models are built and solved in practice. Obviously, models rely on strong simplifications on the level of structure (by omitting elements from the model), dynamics (by deciding which interactions exist in a model), and function (by assigning specific functions to molecules,

Example 6.10 Reaction Velocity and Steady-State Flux

In metabolic steady states, the single reactions are tightly coupled by their common fluxes. Figure 6.19 shows this for a simple two-reaction chain. To study the first reaction in isolation, we fix the concentrations of substrate X and product Y . The reaction rate is given by the rate law $v_1(X, Y, E_1)$ and the response to a small increase of enzyme activity is described by the elasticity coefficient $\tilde{\pi}_{E_1}^{v_1} = \partial v_1 / \partial E_1$. By increasing the enzyme activity at fixed reactant levels, we can increase the reaction rate to arbitrarily high values. Now we couple both reactions and study a steady flux in which the levels of X and Z are fixed and the level of Y determined by the steady-state condition. The rate of the first reaction equals the steady-state flux $j(x, z, E_1, E_2)$, and the effect of an increased enzyme activity is given by a response coefficient $\tilde{R}_{E_1}^j = \partial j / \partial E_1$. Now the first enzyme has only a limited effect on the reaction rate: As its activity increases, its flux control goes down and the flux is mostly controlled by the second enzyme.

pathways, or organs). In particular, we assume that our biological system can be described in isolation, and that it can be subdivided into simple interacting components. We thus choose a number of substances and reactions to be modeled and postulate *direct* causal interactions between them. These causalities define our network.

Starting from this causal model, we study how local interactions translate into global behavior. When integrating the model, we trace the local interactions step-by-step. This becomes visible in the Euler method for integration of differential equations: In each time step, we consider direct influences between neighboring elements; then, however, perturbations propagate step-by-step throughout the network. A steady state, even though

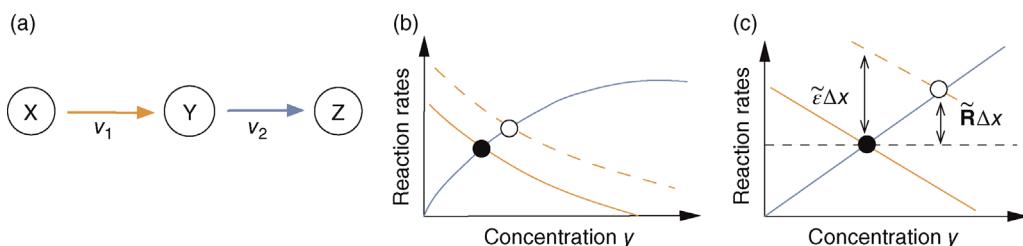


Figure 6.19 Immediate and long-term responses in metabolic systems are respectively described by elasticities and response coefficients. (a) Chain of two reactions with external metabolites X and Z and a reaction intermediate Y . (b) The reaction rates v_1 and v_2 depend on the internal level y . In steady state, both rates must be identical (black dot). If v_1 is increased – for example, by an increase of the external substrate X (broken line), steady-state flux and intermediate concentration are shifted (white dot). (c) In the magnified scheme, we can see the direct increase of the reaction rate $\Delta v = \tilde{\epsilon} \Delta x$ (depending on the reaction elasticity $\tilde{\epsilon}_{v_1}^{v_1}$) and the long-term increase $\Delta j = \tilde{R}_x \Delta x$ of the steady-state flux (depending on the response coefficient \tilde{R}).

it may look simple, can be difficult to understand from a causal point of view: It is a state in which elements are tightly interacting, but with a zero net effect in terms of dynamic changes. In metabolic analysis, the transition from direct influences (described by differential equations) to systemic behavior (described by control coefficients) is achieved by a matrix inversion (in the calculation of control coefficients). Also in this case, we

move from local interactions to a global, systemic behavior, which can then be compared with cell behavior in experiments or with phenomenological cell models. Finally, a network can be subdivided into pathways for convenience, for instance, in modular response analysis; since there is no “objective truth” about these modules, they may simplify our understanding, but should not change the result of the calculations.

Exercises

Section 6.1

- 1) *Linear regression.* A data set $\{(t_1, y_1), (t_2, y_2), \dots\}$ has been generated by a linear model:

$$y(t) = \theta_1 t + \theta_2 + \xi_t,$$

with random errors ξ_t . (a) Given the vectors $\mathbf{t}=(t_1, t_2, \dots)^T$ and $\mathbf{y}=(y_1, y_2, \dots)^T$, estimate the model parameters θ_1 and θ_2 by maximizing the likelihood. Assume that the errors ξ_t are independent Gaussian random variables with mean 0 and variance σ^2 .

- 2) *Bootstrapping procedure for the empirical mean.* The expected value of a random number X can be estimated by the empirical mean value $\bar{x} = 1/n \sum_{m=1}^n x^{(m)}$ of n realizations $x^{(1)}, \dots, x^{(n)}$. (a) Compute the mean and variance of the estimator \bar{x} . (b) Choose a distribution of X and approximate the mean and the variance of \bar{x} numerically by repeatedly drawing samples $(x^{(1)}, \dots, x^{(n)})$. (c) Implement a bootstrapping procedure and assess the distribution of the estimate \bar{x} based on a single sample $(x^{(1)}, \dots, x^{(n)})$. (d) Explain why these three results differ.
- 3) *One-norm and two-norm.* The method of least squares can be derived from the maximum likelihood estimator under the assumption of independent standard Gaussian errors. (a) Assume that measurement errors ξ are not Gaussian, but follow an exponential distribution with density $p(\xi) \sim \exp(-|\xi|/a)$. Find the minimization principle that replaces the method of least squares in this case. (b) Assume that a model is fitted to the same data set (i) by the principle of least squares or (ii) by the minimization principle derived in (a). How will the two fitting results differ qualitatively?
- 4) *Local and global optimization.* (a) Why is it important in parameter estimation to find a global optimum rather than a suboptimal local one? Do local optimum points also have a relevance?

Section 6.2

- 5) *Experimental design.* A kinetic model has been fitted to an experimental concentration time series. An additional data point can be measured, and you can choose the time point at which the measurement will take place. How would you choose the best time point for the measurement, and what circumstances would influence your choice?

- 6) *Selection criteria.* Three models A, B, and C have been fitted to experimental data ($n=10$ data points) by a maximum likelihood fit. The numbers k of free parameters and the optimized likelihood values are given below. (a) Calculate the selection criteria AIC, AICc, and BIC, and use these results to choose between the models. (b) Assume that the models are nested, that is, A is a submodel of B, and B is a submodel of C. Decide for one of the models by using the likelihood ratio test.

Model	A	B	C
k	2	3	4
In L	10.0	5.0	2.0

- 7) *Model selection.* Models A and B are supposed to explain a given experimental time series. Model A contains more free parameters than B and fits the data better. Discuss reasons for choosing model A or model B. What would you do to choose between them in practice, and how could you verify your choice?
- 8) *Bayesian parameter estimation.* A measured quantity $x^{\text{exp}} = x + \xi$ consists of a true value x and an additive error ξ . Compute the posterior distribution of x given a measured value x^{exp} ; assume that the error ξ is Gaussian distributed with mean value 0 and standard deviation σ_ξ and that the prior distribution of x is also Gaussian, with mean value μ_{prior} and standard deviation σ_{prior} .

Section 6.3

- 9) *Fast buffering.* A metabolite appears in a kinetic model in two forms, either free or bound to proteins; there is a fast conversion between both forms, and only the free form participates in chemical reactions. Explain how the model could be modified to describe the metabolite only by its total concentration.
- 10) *Complete cell models.* Does the notion of a complete cell model make any sense? (a) Speculate about possible definitions of “complete models.” (b) Estimate roughly the number of variables and parameters in models of living cells. Consider the following types of model: (i) Kinetic model of whole-cell metabolism without spatial structure. (ii) Compartment model including organelles. (iii) Particle-based model describing single molecules and their complexes in different conformation states. (iv) Model with atomic resolution.
- 11) *All models are wrong.* Discuss the phrase by George Box “Essentially, all models are wrong, but some are useful.” What do you think of it? Does it give any helpful advice for modeling?
- 12) *Quasi-steady state.* Consider a metabolic pathway $A \rightarrow B \rightarrow C \rightarrow$ with irreversible mass-action kinetics. The concentrations of B and C are described by the differential equation system:

$$\begin{aligned} \frac{db}{dt} &= k_1a - k_2b, \\ \frac{dc}{dt} &= k_2b - k_3c. \end{aligned}$$

Assume that the second reaction is much faster than the other reactions, $k_2 \gg k_1, k_3$. Use the

quasi-steady-state approximation to replace the first differential equation by an algebraic equation.

- 13) *Quasi-equilibrium.* Consider the same pathway, with the second reaction being fast and reversible. The differential equation system reads

$$\begin{aligned} \frac{db}{dt} &= k_1a - k_{+2}b + k_{-2}c, \\ \frac{dc}{dt} &= k_{+2}b - k_{-2}c - k_3c. \end{aligned}$$

The conversion between B and C is much faster than the other reactions, $k_{\pm 2} \gg k_1, k_3$; use the quasi-equilibrium approximation to express b by an algebraic equation.

Section 6.4

- 14) *Holism and reductionism.* (a) Discuss Aristotle’s proposition “The whole is more than the sum of its parts.” in the context of biochemical systems and mathematical models describing them. (b) Speculate about the advantages and disadvantages of reductionist and holistic approaches in systems biology.
- 15) *Oscillations by interactions.* Consider a pair of dynamical systems with the following properties: Each system, operating in isolation, shows a stable steady state; when coupled, both systems show stable oscillations. Describe how such oscillations can arise, find real-world examples, and formulate a biochemical model with these properties.
- 16) *Modular response analysis.* Derive Eq. (6.65) from Eqs. (6.64).

References

- 1 Vanrolleghem, P.A. and Heijnen, J.J. (1998) A structured approach for selection among candidate metabolic network models and estimation of unknown stoichiometric coefficients. *Biotechnol. Bioeng.*, 58 (2–3), 133–138.
- 2 Wiechert, W. (2004) Validation of metabolic models: concepts, tools, and problems, in *Metabolic Engineering in the Post Genomic Era* (eds. B.N. Kholodenko and H.V. Westerhoff), Horizon Bioscience, Taylor & Francis.
- 3 Takors, R., Wiechert, W., and Weuster-Botz, D. (1997) Experimental design for the identification of macrokinetic models and model discrimination. *Biotechnol. Bioeng.*, 56, 564–567.
- 4 Seber, G.A.F. and Wild, C.J. (2005) *Nonlinear Regression*, Wiley-Interscience.
- 5 Hoops, S., Sahle, S., Gauges, R., Lee, C., Pahle, J., Simus, N., Singhal, M., Xu, L., Mendes, P., and Kummer, U. (2006) COPASI: a complex pathway simulator. *Bioinformatics*, 22 (24), 3067–3074.
- 6 Sontag, E.D. (2002) For differential equations with r parameters, $2r+1$ experiments are enough for identification. *J. Non Linear Sci.*, 12 (6), 553–583.
- 7 Efron, B. and Tibshirani, R. (1993) *An Introduction to the Bootstrap*, Chapman & Hall/CRC.
- 8 Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. (1997) *Bayesian Data Analysis*, Chapman & Hall, New York.
- 9 Liebermeister, W. and Klipp, E. (2006) Bringing metabolic networks to life: integration of kinetic, metabolic, and proteomic data. *Theor. Biol. Med. Model.*, 3, 42.
- 10 Heckerman, D. (1998) A tutorial on learning with Bayesian networks. *Learning in Graphical Models*, Kluwer Academic Publishers.
- 11 Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, Inc.

- 12** Jensen, F.V. (2001) *Bayesian Networks and Decision Graphs*, Springer, New York.
- 13** Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000) Using Bayesian networks to analyze expression data. *J. Comput. Biol.*, 7 (3–4), 601–620.
- 14** Perrin, B.E., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., and d'Alché Buc, F. (2003) Gene networks inference using dynamic Bayesian networks. *Bioinformatics*, 19 (ii), 138–148.
- 15** Schomburg, I., Chang, A., Ebeling, C., Gremse, M., Heldt, C., Huhn, G., and Schomburg, D. (2004) Brenda, the enzyme database: updates and major new developments. *Nucleic Acids Res.*, 32 (Database Issue), D431–D433.
- 16** Wittig, U., Golebiewski, M., Kania, R., Krebs, O., Mir, S., Weidemann, A., Anstein, S., Saric, J., and Rojas, I. (2006) SABIO-RK: integration and curation of reaction kinetics data, in *Data Integration in the Life Sciences: Proceedings of the 3rd International Workshop on (DILS'06) Hinxton, UK* (eds. U. Leser, F. Naumann, and B. Ekman), Lecture Notes in Computer Science, vol. 4075, pp. 94–103.
- 17** Jaynes, E.T. (1957) Information theory and statistical mechanics. *Phys. Rev. A*, 106, 620–630.
- 18** Goldberg, R.N. (1999) Thermodynamics of enzyme-catalyzed reactions: Part 6–1999 update. *J. Phys. Chem. Ref. Data*, 28, 931.
- 19** Liebermeister, W. (2005) Predicting physiological concentrations of metabolites from their molecular structure. *J. Comp. Biol.*, 12 (10), 1307–1315.
- 20** Bar-Even, A., Noor, E., Savir, Y., Liebermeister, W., Davidi, D., Tawfik, D.S., and Milo, R. (2011) The moderately efficient enzyme: evolutionary and physicochemical trends shaping enzyme parameters. *Biochemistry*, 21, 4402–4410.
- 21** Liebermeister, W. and Klipp, E. (2005) Biochemical networks with uncertain parameters. *IEE Proc. Syst. Biol.*, 152 (3), 97–107.
- 22** Ederer, M. and Gilles, E.D. (2007) Thermodynamically feasible kinetic models of reaction networks. *Biophys. J.*, 92, 1846–1857.
- 23** Liebermeister, W. and Klipp, E. (2006) Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. *Theor. Biol. Med. Model.*, 3, 41.
- 24** Liebermeister, W., Uhendorf, J., and Klipp, E. (2010) Modular rate laws for enzymatic reactions: thermodynamics, elasticities, and implementation. *Bioinformatics*, 26 (12), 1528–1534.
- 25** Lubitz, T., Schulz, M., Klipp, E., and Liebermeister, W. (2010) Parameter balancing for kinetic models of cell metabolism. *J. Phys. Chem. B*, 114 (49), 16298–16303.
- 26** Stanford, N.J., Lubitz, T., Smallbone, K., Klipp, E., Mendes, P., and Liebermeister, W. (2013) Systematic construction of kinetic models from genome-scale metabolic networks. *PLoS One*, 8 (11), e79195.
- 27** Mendes, P. and Kell, D.B. (1998) Non-linear optimization of biochemical pathways: application to metabolic engineering and parameter estimation. *Bioinformatics*, 14 (10), 869–883.
- 28** Moles, C.G., Mendes, P., and Banga, J.R. (2003) Parameter estimation in biochemical pathways: a comparison of global optimization methods. *Genome Res.*, 13 (11), 2467–2474.
- 29** Rodriguez-Fernandez, M., Mendes, P., and Banga, J.R. (2006) A hybrid approach for efficient and robust parameter estimation in biochemical pathways. *Biosystems*, 83 (2–3), 248–265.
- 30** Rodriguez-Fernandez, M., Egea, J.A., and Banga, J.R. (2006) Novel metaheuristic for parameter estimation in nonlinear dynamic biological systems. *BMC Bioinformatics*, 7, 483.
- 31** Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., and Teller, E. (1953) Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21 (6), 1087–1092.
- 32** Hastings, W.K. (1970) Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57 (1), 97–109.
- 33** Kirkpatrick, S., Gelatt, C.D., and Vecchi, M.P. (1983) Optimization by simulated annealing. *Science*, 220 (4598), 671–680.
- 34** Storn, R. and Price, K. (1997) Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.*, 11, 341–359.
- 35** Runarsson, T.P. and Yao, X. (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput.*, 4 (3), 284–294.
- 36** Wiechert, W. (2004) Validation of metabolic models: concepts, tools, and problems, in *Metabolic Engineering in the Post Genomic Era* (eds. B.N. Kholodenko and H.V. Westerhoff), Horizon Bioscience, Taylor & Francis.
- 37** Haunschmid, M., Freisleben, B., Takors, R., and Wiechert, W. (2005) Investigating the dynamic behaviour of biochemical networks using model families. *Bioinformatics*, 21, 1617–1625.
- 38** Borges, J.L. (1999) Suarez Miranda, Viajes de varones prudentes, Libro IV, Cap. XLV, Lerida, 1658, in *Jorge Luis Borges: Collected Fictions*, Penguin.
- 39** Box, G.E.P. and Draper, N.R. (1987) *Empirical Model-Building and Response Surfaces*, John Wiley & Sons, Inc., New York.
- 40** Dyson, F. (2004) A meeting with Enrico Fermi. *Nature*, 427, 297.
- 41** Freedman, D.A. (1983) A note on screening regression equations. *Am. Stat.*, 37 (2), 152–155.
- 42** Atkinson, A.C. (1981) Likelihood ratios, posterior odds and information criteria. *J. Econom.*, 16, 15–20.
- 43** Ghosh, J.K. and Samanta, T. (2001) Model selection: an overview. *Curr. Sci.*, 80 (9), 1135.
- 44** Hansen, M.H. and Yu, B. (2001) Model selection and the principle of minimum description length. *J. Am. Stat. Assoc.*, 96, 746–774.
- 45** Johnson, J.B., and Omland, K.S. (2004) Model selection in ecology and evolution. *Trends Ecol. Evol.*, 19 (2), 101–108.
- 46** Vuong, Q.H. (1989) Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica*, 57 (2), 307–333.
- 47** Akaike, H. (1974) A new look at the statistical model identification. *IEEE Trans. Automat. Contr.*, 19 (6), 716–723.
- 48** Hurvich, C.M. and Tsai, C.-L. (1989) Regression and time series model selection in small samples. *Biometrika*, 76, 297–307.
- 49** Schwarz, G. (1978) Estimating the dimension of a model. *Ann. Stat.*, 6 (2), 461–464.
- 50** Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. (1997) *Bayesian Data Analysis*, Chapman & Hall, New York.
- 51** Stewart, W.E., and Henson, T.L. (1996) Model discrimination and criticism with single-response data. *AIChE J.*, 42 (11), 3055.
- 52** Okino, M.S. and Mavrovouniotis, M.L. (1998) Simplification of mathematical models of chemical reaction systems. *Chem. Rev.*, 98 (2), 391–408.
- 53** Visser, D., Schmid, J.W., Mauch, K., Reuss, M., and Heijnen, J.J. (2004) Optimal re-design of primary metabolism in *Escherichia coli* using linlog kinetics. *Metab. Eng.*, 6 (4), 378–390.
- 54** Degenring, D., Fromel, C., Dikta, G., and Takors, R. (2004) Sensitivity analysis for the reduction of complex metabolism models. *J. Process Control*, 14, 729–745.
- 55** Liebermeister, W., Baur, U., and Klipp, E. (2005) Biochemical network models simplified by balanced truncation. *FEBS J.*, 272 (16), 4034–4043.
- 56** Roussel, M.R. and Fraser, S.J. (2001) Invariant manifold methods for metabolic model reduction. *Chaos*, 11 (1), 196–206.
- 57** Higgins, J.J. (1965) Dynamics and control in cellular systems, in *Control of Energy Metabolism* (eds. B. Chance, R.W. Estabrook, and J.R. Williamson), Academic Press, New York, p. 13.
- 58** Reich, J.G. and Sel'kov, E.E. (1975) Time hierarchy, equilibrium and non-equilibrium in metabolic systems. *BioSystems*, 7, 39–50.

- 59** Easterby, J.S. (1981) A generalized theory of the transition time for sequential enzyme reactions. *Biochem. J.*, 199, 155–161.
- 60** Heinrich, R. and Rapoport, T.A. (1975) Mathematical analysis of multienzyme systems: II. Steady state and transient control. *BioSystems*, 7, 130–136.
- 61** Lloréns, M., Nuno, J.C., Rodríguez, Y., Meléndez-Hevia, E., and Montero, F. (1999) Generalization of the theory of transition times in metabolic pathways: a geometrical approach. *Biophys. J.*, 77, 23–36.
- 62** Ingalls, B.P. (2004) A frequency domain approach to sensitivity analysis of biochemical systems. *J. Phys. Chem. B*, 108, 1143–1152.
- 63** Liebermeister, W. (2005) Response to temporal parameter fluctuations in biochemical networks. *J. Theor. Biol.*, 234 (3), 423–438.
- 64** Ingalls, B.P. and Sauro, H.M. (2003) Sensitivity analysis of stoichiometric networks: an extension of metabolic control analysis to non-steady state trajectories. *J. Theor. Biol.*, 222 (1), 23–36.
- 65** Zobeley, J., Lebiedz, D., Ishmurzin, A., and Kummer, U. (2005) A new time-dependent complexity reduction method for biochemical systems, in *Transactions on Computational Systems Biology* (ed. C. Prami *et al.*), Lecture Notes in Computer Science, vol. 3380, Springer, pp. 90–110.
- 66** Moore, B.C. (1981) Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Trans. Automat. Contr.*, 26, 17–32.
- 67** Surovtsova, I., Simus, N., Hübner, K., Sahle, S., and Kummer, U. (2012) Simplification of biochemical models: a general approach based on the analysis of the impact of individual species and reactions on the systems dynamics. *BMC Syst. Biol.*, 6, 14.
- 68** Teusink, B., Passarge, J., Reijenga, C.A., Esgalhado, E., van der Weijden, C.C., Schepper, M., Walsh, M.C., Bakker, B.M., van Dam, K., Westerhoff, H.V., and Snoep, J.L. (2000) Can yeast glycolysis be understood in terms of *in vitro* kinetics of the constituent enzymes? Testing biochemistry. *Eur. J. Biochem.*, 267, 5313–5329.
- 69** Chassagnole, C., Raïs, B., Quentin, E., Fell, D.A., and Mazat, J. (2001) An integrated study of threonine-pathway enzyme kinetics in *Escherichia coli*. *Biochem. J.*, 356, 415–423.
- 70** Karr, J.R., Sanghvi, J.C., Macklin, D.N., Gutschow, M.V., Jacobs, J.M., Bolival, B., Assad-Garcia, N., Glass, J.I., and Covert, M.W. (2012) A whole-cell computational model predicts phenotype from genotype. *Cell*, 150 (2), 389–401.
- 71** Petersen, S., Lierer, E.v., de Graaf, A.A., Sahm, H., and Wiechert, W. (2004) A multi-scale approach for the predictive modeling of metabolic regulation, in *Metabolic Engineering in the Post Genomic Era* (eds. B.N. Kholodenko and H.V. Westerhoff), Horizon Bioscience, Taylor & Francis.
- 72** Liebermeister, W., Baur, U., and Klipp, E. (2005) Biochemical network models simplified by balanced truncation. *FEBS J.*, 272 (16), 4034–4043.
- 73** Hofmeyr, J.-H.S. and Cornish-Bowden, A. (2000) Regulating the cellular economy of supply and demand. *FEBS Lett.*, 476 (1–2), 47–51.
- 74** He, F., Fromion, V., and Westerhoff, H.V. (2013) (Im)Perfect robustness and adaptation of metabolic networks subject to metabolic and gene-expression regulation: marrying control engineering with metabolic control analysis. *BMC Syst. Biol.*, 7, 131.
- 75** van Eunen, K., Rossell, S., Bouwman, J., Westerhoff, H.V., and Bakker, B.M. (2011) Quantitative analysis of flux regulation through hierarchical regulation analysis. *Methods Enzymol.*, 500, 571–595.
- 76** Stanford, N.J., Lubitz, T., Smallbone, K., Klipp, E., Mendes, P., and Liebermeister, W. (2013) Systematic construction of kinetic models from genome-scale metabolic networks. *PLoS One*, 8 (11), e79195.
- 77** Hofmeyr, J.S., Gqwaka, O.P.C., and Rohwer, J.M. (2013) A generic rate equation for catalysed, template-directed polymerisation. *FEBS Lett.*, 587, 2868–2875.
- 78** Schuster, S., Kahn, D., and Westerhoff, H.V. (1993) Modular analysis of the control of complex metabolic pathways. *Biophys. Chem.*, 48, 1–17.
- 79** Bruggeman, F., Westerhoff, H.V., Hoek, J.B., and Kholodenko, B. (2002) Modular response analysis of cellular regulatory networks. *J. Theor. Biol.*, 218, 507–520.
- 80** Wolf, J. and Heinrich, R. (2000) Effect of cellular interaction on glycolytic oscillations in yeast: a theoretical investigation. *Biochem. J.*, 345, 312–334.
- 81** Wolf, J., Sohn, H.-Y., Heinrich, R., and Kuriyama, H. (2001) Mathematical analysis of a mechanism for autonomous metabolic oscillations in continuous culture of *Saccharomyces cerevisiae*. *FEBS Lett.*, 499, 230–234.
- 82** du Preez, F.B., van Niekerk, D.D., Kooi, B., Rohwer, J.M., and Snoep, J.L. (2012) From steady-state to synchronized yeast glycolytic oscillations I: model construction. *FEBS J.*, 279, 2810–2822.

Further Reading

Modular Response Analysis: Bruggeman, F., Westerhoff, H.V., Hoek, J.B., and Kholodenko, B. (2002) Modular response analysis of cellular regulatory networks. *J. Theor. Biol.*, 218, 507–520.

Bottom-Up Modeling (Threonine Pathway Model): Chassagnole, C., Raïs, B., Quentin, E., Fell, D.A., and Mazat, J. (2001) An integrated study of threonine-pathway enzyme kinetics in *Escherichia coli*. *Biochem. J.*, 356, 415–423.

Bootstrap: Efron, B. and Tibshirani, R. (1993) *An Introduction to the Bootstrap*, Chapman & Hall/CRC.

Bayesian Data Analysis: Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. (1997) *Bayesian Data Analysis*, Chapman & Hall, New York.

Metabolic Response to Periodic Perturbations: Ingalls, B.P. (2004) A frequency domain approach to sensitivity analysis of biochemical systems. *J. Phys. Chem. B*, 108, 1143–1152.

Metabolic Response for Time Series: Ingalls, B.P. and Sauro, H.M. (2003) Sensitivity analysis of stoichiometric networks: an extension of metabolic control analysis to non-steady state trajectories. *J. Theor. Biol.*, 222 (1), 23–36.

Modular Whole-Cell Model of *Mycoplasma genitalium*: Karr, J.R., Sanghvi, J.C., Macklin, D.N., Gutschow, M.V., Jacobs, J.M., Bolival, B., Jr., Assad-Garcia, N., Glass, J.I., and Covert, M.W. (2012) A whole-cell computational model predicts phenotype from genotype. *Cell*, 150 (2), 389–401.

Flux-Based Model Construction: Stanford, N.J., Lubitz, T., Smallbone, K., Klipp, E., Mendes, P., and Liebermeister, W. (2013) Systematic construction of kinetic models from genome-scale metabolic networks. *PLoS One*, 8 (11), e79195.

Bottom-Up Modeling (Glycolysis Model): Teusink, B., Passarge, J., Reijenga, C.A., Esgalhado, E., van der Weijden, C.C., Schepper, M., Walsh, M.C., Bakker, B.M., van Dam, K., Westerhoff, H.V., and Snoep, J.L. (2000) Can yeast glycolysis be understood in terms of *in vitro* kinetics of the constituent enzymes? Testing biochemistry. *Eur. J. Biochem.*, 267, 5313–5329.

Hierarchical Regulation Analysis: van Eunen, K., Rossell, S., Bouwman, J., Westerhoff, H.V., and Bakker, B.M. (2011) Quantitative analysis of flux regulation through hierarchical regulation analysis. *Methods Enzymol.*, 500, 571–595.

Discrete, Stochastic, and Spatial Models

7

Kinetic and stoichiometric models provide a useful level of description for metabolic and signaling pathways, provided that molecule numbers are large and that spatial cell structure can be captured by the assumption of well-mixed compartments. But sometimes, these model formalisms are not applicable. On the one hand, kinetic models may be too complicated: for models of gene expression networks, approximate expression levels may be the only data available. If these data solely allow us to distinguish between active and inactive genes, we should reflect this knowledge by a discrete model: in a Boolean network model, genes can switch between two states (“on” or “off”) and, in doing so, affect the states of other genes.

On the other hand, kinetic models may be too simple because they ignore important microscopic details, for example, the movement and reactions of single molecules. Kinetic models provide a good and numerically simple approximation whenever random fluctuations can be neglected. However, whenever molecule numbers are low – as is typically the case in gene expression – or in models with nonlinear and unstable dynamics, random fluctuations can become important. In such cases, random models can provide a much more detailed description. Moreover, many biological processes rely on spatial structure, for example, the spatial distribution of pathogens that control the emergence of body shapes in embryonic development. To describe such pattern formation processes, spatial reaction–diffusion models may be much more suitable than simple kinetic compartment models.

The three modeling approaches introduced in this chapter – discrete, stochastic, and spatial models – exist in many variants and will appear again in later chapters,

7.1 Discrete Models

- Boolean Networks
- Petri Nets

7.2 Stochastic Modeling of Biochemical Reactions

- Chance in Biochemical Reaction Systems
- The Chemical Master Equation
- Stochastic Simulation
- Chemical Langevin Equation and Chemical Noise
- Dynamic Fluctuations
- From Stochastic to Deterministic Modeling

7.3 Spatial Models

- Types of Spatial Models
- Compartment Models
- Reaction–Diffusion Systems
- Robust Pattern Formation in Embryonic Development
- Spontaneous Pattern Formation
- Linear Stability Analysis of the Activator–Inhibitor Model

Exercises

References

Further Reading

where specific biological systems are modeled. Compared with simple kinetic models, spatial and stochastic models open some new perspectives. First, they explain randomness that emerges on a molecular scale, but also affects the macroscopic behavior of cells. Second, they account for processes in space and time, for example, the movements of vesicles in cells, or the growth of blood vessels within tissues. Third, they can capture the behavior of individual agents, for instance, viruses infecting a cell or cancer cells conquering a tissue, and thus come closer and closer to our mental pictures of cell and body physiology.

7.1 Discrete Models

7.1.1 Boolean Networks

Summary

Boolean models are, perhaps, the simplest representation of dynamic biological networks. However, they have proven to be very helpful to understand complex regulatory networks for which we have not much more information at hand than their nodes (e.g., the genes) and whether they interact. By assigning simple rules, we can analyze the potential network behavior in time. Since – at least in the basic version – the numbers of states and state transitions are finite, we can try to investigate the full state space. Boolean networks are discrete, both in state and in time.

The most prominent application of Boolean networks has been in the analysis of gene regulatory networks.

7.1.1.1 Basic Principles of Boolean Networks

Boolean models are based on proposition logic founded by George Boole, 1815–1864. This type of logic entails the principle of bivalence: Any statement is either true or false. A third possibility or contradictions are excluded. Statements can be combined using the operators such as “and,” “or,” or “not” and combinations thereof. The truth value of combined statements depends only on the truth value of the individual statements and on the operation between them.

Boolean logic has been applied to biological processes such as regulation of gene expression in the framework of Kauffman's *NK* Boolean networks [1–4]. Genes are the elements of the network. Levels of gene expression are approximated by only two states: each gene is either expressed (is assigned the value "1") or not expressed ("0"). The network has N elements or *nodes*. Each element has K inputs (regulatory interactions) and one output, that is, its state. Inputs and

Table 7.1 Boolean rules (truth table) for systems with one input.

Input	Output		
A	0	A	Not A
0	0	0	1
1	0	1	0
Rule	0	1	2
			3

output have binary values (1 or 0). Since every node can be in one of the two different states, a network of N genes can assume 2^N different states. An N -dimensional vector of elements can describe the state at time t . The values are updated in discrete time steps; that is, the value of each element at time $t+1$ depends on the values of its inputs at time t . Boolean networks always have a finite (although possibly large) number of possible states and hence only a finite number of possible state changes. The state changes of an individual element are specified by the Boolean rules that relate the output to the inputs. There are 2^{2^K} possible Boolean rules for a node with K inputs. The rules can be enumerated according to the respective binary numbers of output or rules can be associated with their meaning in normal life (*and*, *or*) (see Tables 7.1 and 7.2).

The dynamics of Boolean networks can be further characterized with the following notions. The sequence of states given by the Boolean transitions represents the *trajectory* of the system. Since the number of states in the state space is finite, the number of possible transitions is also finite. Therefore, each trajectory will lead either to a steady state (as in Figure 7.2c) or to a steady cycle (as in Figure 7.2a). The *cycle length* is the number of states on the cycle. A steady state is a cycle with length 1. These states or state sequences are also called *attractors*.

Transient states are those states that do not belong to an attractor. All states that lead to the same attractor constitute its *basin of attraction* (or confluent) such as all

Table 7.2 Boolean rules (truth table) for systems with two inputs.

Input		Output															
A	B	0	And	A	B	Xor	Or	Nor	Not B	Not A	Nand	1					
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1			
0	1	0	0	0	1	1	1	1	0	0	0	1	1	1			
1	0	0	0	1	1	0	0	1	0	0	1	0	0	1			
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0			
Rule		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Example 7.1 The network

$$A \rightarrow B \rightarrow C \rightarrow D \quad (7.1)$$

has the connectivity $K = 1$. Let

$$A = \text{constant},$$

$$B = f_B(A) = \text{not } A,$$

$$C = f_C(B) = \text{not } B,$$

$$D = f_D(C) = C$$

with the initial state $(A, B, C, D)(t_0) = (1, 0, 0, 0)$. The following states at successive time points t_1, t_2 , and so on are

$$(A, B, C, D)(t_1) = (1, 0, 1, 0),$$

$$(A, B, C, D)(t_2) = (1, 0, 1, 1),$$

:

$$(A, B, C, D)(t_i) = (1, 0, 1, 1) \text{ for } i = 2, \dots, \infty.$$

After two steps, the system has attained a fix point (Figure 7.1).

	A_1	B_1	C_1	D_1
t_0	Black	White	White	White
t_1	White	Black	Black	White
t_2	White	White	Black	Black
t_3	Black	Black	Black	Black

Figure 7.1 Temporal behavior of the network depicted in Example 7.1. Black denotes ON (1) and white denotes OFF (0). The indices of the nodes indicate the number of the rules according to Table 7.1.

states leading to $(1, 1, 1)$ or all states leading to $(0, 0, 0)$ in Figure 7.3c. The *path length* (or run-in length or transient length) is the number of states between initial states and those entering the attractor. Further important characteristics of a Boolean network are *average path length* and the *number of attractors*, which can lie between 1 and N .

There are different ways of measuring the distance between states. One way is to enumerate the number of steps on a trajectory to get from one state to the other state. Another way is to count the number of different elements in each state vector. If $N = 5$, for example, the difference between the states $(1, 0, 1, 0, 1)$ and $(1, 1, 1, 0, 0)$ would be 2 (for the second and fifth entries). This is also called the *Hamming distance*.

Figure 7.4 presents an example for a larger network with one periodic attractor and its full basin of attraction.

Boolean networks can be used to study network perturbations. A potential perturbation is the change of a node state (from 0 to 1 or 1 to 0). Then one can follow the dynamics of the system, which moves either to the same attractor as before or to another attractor, if this perturbation shifted the system into another basin of attraction. Further types of perturbations are to change the rules of individual nodes (e.g., from AND to OR) or to modify the network wiring (i.e., to choose different input nodes for a selected node). Such perturbations can completely change number and types of attractors or their basins of attraction. Studying the effect of network perturbations is useful, if the network structure is not fully understood from the available data and one wants to test alternative hypotheses or if one wants to test the effect of biologically meaningful perturbations such as knockout mutations.

7.1.1.2 Advanced Types of Boolean Networks

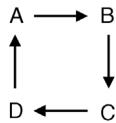
As mentioned before, Boolean networks consist of N nodes. Each node can have only one of the two possible values (0 or 1) and the full system has 2^N possible states.

In Section 7.1.1.1, we assumed a concerted (synchronous) update of all states. In *asynchronous* Boolean networks, a random node is selected at each time point and updated [6]. Repeated simulation of the same network with identical starting conditions can provide an average behavior of the network. The approach of asynchronous updates reflects the experience that in biological networks not all nodes change their states necessarily at the same time.

Random Boolean networks are a generalization of Boolean networks. They also consist of N nodes and their K connections. Their major feature is that update rules are chosen randomly during construction. They remain constant over the time course.

Boolean networks have been used to explore general and global properties of large gene expression networks. Studying random networks (i.e., each gene has K inputs, and each gene is controlled by a randomly assigned Boolean function), Kauffman [3,7] has shown that systems exhibit highly ordered dynamics for small K and certain choices of rules. Both the median number of attractors and the length of attractors are on the order of \sqrt{N} . Kauffman suggested the interpretation of the number of possible attractors as the number of possible cell types arising from the same genome.

Probabilistic Boolean networks [8] assign with a certain probability update rules to nodes at each time step. They have also been used to analyze potential cell fates. An example is the maintenance of pluripotency or transition

Example 7.2

(7.2)

Another typical structure of a Boolean network with $K = 1$ is the closed loop, in which the input of the first element is the output of the last element.

Periodic behavior is possible if all elements obey rules 1 or 2 (see Table 7.1). Assume, for example, that all elements follow rule 1 and the initial state is $(ABCD) = (1000)$. The states for the following time steps will be (0100) , (0010) , (0001) , and again (1000) , which closes the cycle. Rule 0 or 3 breaks the periodic behavior, since the output of the respective element is no longer dependent on the input and leads to constant behavior. Figure 7.2 shows different types of behavior for different rules.

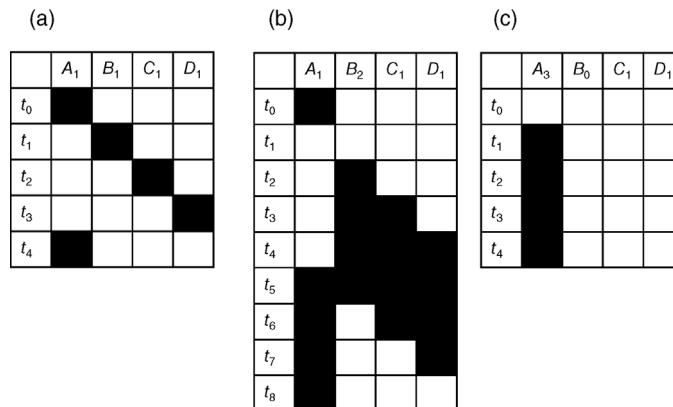


Figure 7.2 Temporal behavior of the closed loop with four nodes and different initial conditions at time t_0 . The indices of the nodes indicate the number of the rules according to Table 7.1.

to differentiation during somatic cell reprogramming steered by the NANOG–OCT4–SOX2 regulatory network and its interplay with chromatin remodeling and DNA methylation [9]. Despite starting from the same initial state, the system could approach different classes of induced pluripotency, differentiation, or cell death with certain probability.

7.1.2 Petri Nets

An alternative to ordinary differential equations (ODEs) for the simulation of time-dependent processes are Petri nets. A Petri net is a graphical and mathematical modeling tool for discrete and parallel systems. The mathematical concept was developed in the early 1960s by Carl Adam Petri. The basic elements of a Petri net are two

types of nodes (called places), transitions, and arcs that connect places and transitions, or vice versa. When represented graphically, places are shown as circles and transitions as rectangles; thus, we speak about bipartite graphs. Places represent objects (e.g., molecules, cars, or machine parts) and transitions describe whether and how individual objects are interconverted. Places can contain zero or more tokens, indicating the number of objects that currently exist. Whether a transition can take place (can fire) or not depends on the places that are connected to the transition by incoming arcs, to contain enough tokens. If this condition is fulfilled, the transition fires and changes the state of the system by removing tokens from the input places and adding tokens to the output places. The number of tokens that are removed and added depends on the weights of the arcs. A simple example is shown in Figure 7.5.

Example 7.3

The following network resembles the basic regulation structure of the NANOG–OCT4–SOX2 network that regulates stemness or differentiation of pluripotent cells (see Chapter 2). All three components exhibit self-activation supported by one of the other components.



The network has $N = 3$ elements and may assume $2^N = 8$ different states. Let the rules be

$$\begin{aligned} A(t+1) &= A(t) \text{ and } B(t), \\ B(t+1) &= A(t) \text{ and } B(t), \\ C(t+1) &= A(t) \text{ or } (B(t) \text{ and } C(t)). \end{aligned}$$

Table 7.3 lists the possible states and the respective following states for (ABC) . Table 7.3 shows that the states (000) and (111) are fix points, since the next state is identical to the current state. See also Figure 7.3.

Table 7.3 Successive states for the Boolean network (ABC) depicted in (7.3).

Current state	000	001	010	011	100	101	110	111
Next state	000	000	000	001	001	001	111	111

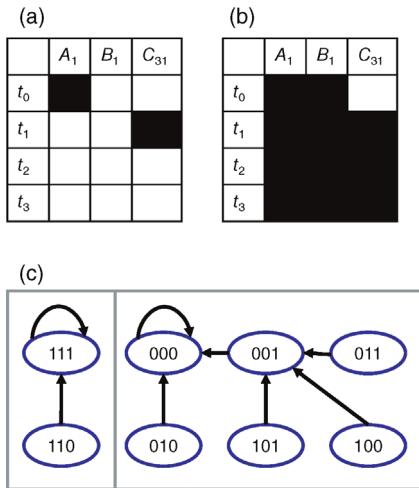


Figure 7.3 Dynamics of the small pluripotency network for different initial conditions. In the bottom, we see all possible states and trajectories. Two states end in (111) , that is, a fully active network ensuring pluripotency; six states evolve toward $(0, 0, 0)$, that is, a silent network that may allow differentiation.

Petri nets not only are an optically pleasing representation of a system, but can also be described mathematically in terms of integer arithmetic (Figure 7.6). To this end, a Petri net is a 5-tuple $\mathbf{G} = (\mathbf{P}, \mathbf{T}, \mathbf{F}, \mathbf{W}, \mathbf{M}_0)$, where $\mathbf{P} = \{P_1, P_2, \dots, P_k\}$ is a finite set of places, $\mathbf{T} = \{T_1, T_2, \dots, T_l\}$ is a finite set of transitions, $\mathbf{A} \subseteq (\mathbf{P} \times \mathbf{T}) \cup (\mathbf{T} \times \mathbf{P})$ is a finite set of arcs representing flow relations, $\mathbf{W} : \mathbf{A} \rightarrow \{1, 2, 3, \dots\}$ is a weight function, and $\mathbf{M}_0 : \mathbf{P} \rightarrow \{0, 1, 2, \dots\}$ is the initial marking.

Note that places and transitions are different types of nodes. The intersection of \mathbf{P} and \mathbf{T} is the empty set. In other words, Petri nets are a type of bipartite network. Through the assignment of tokens and the firing of transitions, the network is dynamic. If multiple transitions are possible with a given marking, any of them may fire. Thus, Petri nets are not deterministic.

The network can also be described by the *incidence matrix* \mathbf{C} , a $(k \times l)$ -matrix reflecting the arc weights (represented by appropriate signs). The matrix entry C_{ij} provides the change of tokens at place P_i upon firing of transition T_j . For the network depicted in Figure 7.5 with four places and one transition, the incidence matrix is $\mathbf{C} = (-2 \ -1 \ 1 \ 2)^T$. For biochemical reaction networks and, especially, for metabolic networks, the incidence matrix is similar to the stoichiometric matrix introduced in Chapter 2. A similar type of mathematical analysis is possible. A *T-invariant* of the incidence matrix is a nonzero vector $\mathbf{x} \in \mathbf{M}_0$ fulfilling $\mathbf{C} \cdot \mathbf{x} = \mathbf{0}$. It represents a set of transitions that together have no effect on the actual marking. For biochemical networks, the *T-invariants* are comparable to steady-state fluxes represented by the kernel \mathbf{K} of the stoichiometric matrix \mathbf{N} with $\mathbf{N} \cdot \mathbf{K} = \mathbf{0}$. The conservation relations of biochemical networks expressed by $\mathbf{G} \cdot \mathbf{N} = \mathbf{0}$ (with $\mathbf{G} \cdot \mathbf{S} = \text{constant}$, \mathbf{S} being the vector of compound concentrations) have their counterparts in the Petri net formalism in the *P-invariants*. *P-invariants* are nonzero vectors $\mathbf{y} \in \mathbf{M}_0$ satisfying $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$. They portray a token conservation rule for a set of places independent of the firing.

A number of typical motifs for Petri nets have been described. *Autocatalysis* denotes a closed circle from a place via a transition back to that place. A siphon is a place for which the set of outgoing arcs includes the set of incoming arcs. If it is marked once, it will remain so. A *deadlock* is a situation where no more transitions are possible (e.g., because there are insufficiently many tokens in the input places to allow firing of the transition such as in the final state of the network depicted in Figure 7.5). Deadlock-free is a net if a transition is enabled for any possible marking. *Liveness* describes the potential of all the transitions to fire. Different levels of liveness can be distinguished, from weak liveness where

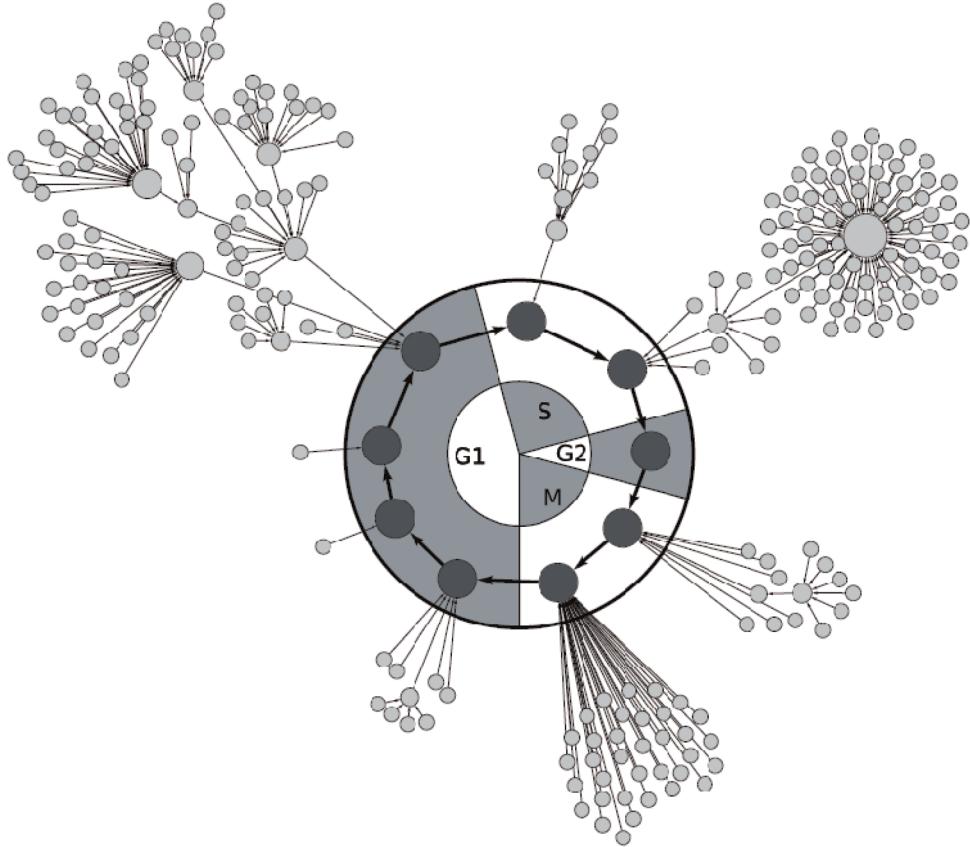


Figure 7.4 Network describing cell cycle dynamics of the yeast *Saccharomyces cerevisiae* with 8 nodes (not shown), 256 states, and 1 cyclic attractor consisting of 9 subsequent states [5].

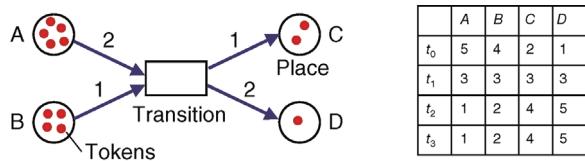


Figure 7.5 A Petri net consists of places and transitions. A transition can fire when the places contain enough tokens as indicated by arc weights. The table shows the number of tokens at each place for successive time steps starting with the present marking.

the transitions may fire to liveness where they may fire arbitrarily or infinitively often.

For simple types of Petri nets, certain properties can be calculated analytically, but often the net has to be simulated to study the long-term system properties. Over the years, many extensions to the basic Petri net model have been developed for the different simulation purposes [10].

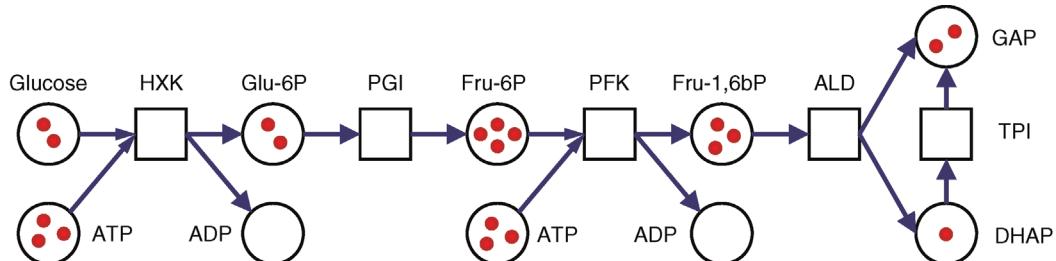


Figure 7.6 The upper glycolysis represented as Petri net. The places are the compounds such as glucose, ATP, or GAP. The transitions are the metabolic reactions transforming compounds into each other. In this example, all arc weights are 1 (not shown) referring to stoichiometric coefficients of 1 in this pathway. Only a few tokens per compound are indicated since typical molecule numbers are much too large to display (on the order of ten thousand to millions).

- 1) Hybrid Petri nets that add the possibility to have places that contain a continuous token number instead of discrete values.
- 2) Timed Petri nets that extend transitions to allow for a specific time delay between the moment when a transition is enabled and the actual firing.
- 3) Stochastic Petri nets that go one step further and allow a random time delay drawn from a probability distribution.
- 4) Hierarchical Petri nets, in which modularity is introduced by representing whole nets as a single place or transition of a larger net.
- 5) Colored Petri nets that introduce different types (colors) of tokens and more complicated firing rules for transitions.

With these extensions, Petri nets are powerful enough to be used for models in systems biology. Biochemical pathways can be modeled with places representing metabolites, transitions representing reactions, and stoichiometric coefficients are encoded as different weights of input and output arcs. Consequently, Petri nets have been used to model metabolic networks [11,12] and signal transduction pathways [13]. Many free and commercial tools are available to explore the behavior of Petri nets. The Petri Nets World webpage (<http://www.informatik.uni-hamburg.de/TGI/PetriNets/>) is an excellent starting point for this purpose.

Macromolecules, although often described as “molecular machines,” differ a lot from the machines we see in our everyday life: on a microscopic level, molecules are constantly formed and destroyed by chemical reactions. Proteins and other molecules tumble back and forth, diffuse, change their conformations, and assemble and disassemble in permanent thermal movement. Thermal motion and the reactions of single molecules can be described mathematically by random processes: reactions occur unpredictably, and each sequence of random events leads to a possible history of the system. Stochastic models allow us to compute mean values, fluctuations, and temporal correlations of system states [14], and individual realizations of random processes can be obtained by stochastic simulation. On larger space and time scales, the microscopic processes give rise to effective macroscopic behavior, as described by kinetic models. How the two perspectives can be linked will be a topic in this chapter.

Stochastic models apply to all biochemical systems, including metabolic and gene expression networks. They can refer to individual stochastic reaction events, randomly varying count numbers of reactions within discrete time intervals, or randomly fluctuating reaction velocities in continuous time. Finally, for well-mixed systems with large particle numbers, the random dynamics lead to deterministic laws as a limiting case [15]. The same biochemical system can be described in all four frameworks, each being an approximation of the previous one. Mathematical details of stochastic processes are given in Section 15.4.

7.2 Stochastic Modeling of Biochemical Reactions

Summary

On a molecular level, chemical reactions are random events and cause molecule numbers to fluctuate. To study the microscopic dynamics in biochemical systems, we can describe chemical reaction systems as stochastic processes, compute the resulting fluctuations of substance amounts, and trace them across cellular networks. Mathematical random processes used in such models can describe individual reaction events (calculation by the chemical master equation or direct simulation), their frequencies in time (calculation by the τ -leaping method), or randomly drifting substance concentrations (chemical Langevin equation), and they entail deterministic models as a limiting case. The temporal fluctuations in substance amounts can be characterized by autocorrelations and spectral densities.

7.2.1 Chance in Biochemical Reaction Systems

Molecule numbers in cells fluctuate in time. As a simple case, consider a molecule species described by integer particle numbers $x(t)$. We assume that molecules are produced with a constant propensity (probability per time unit) and degraded with a constant stochastic rate per molecule. This birth–death process describes a random movement in the state space of molecule numbers

$$0 \longleftrightarrow 1 \longleftrightarrow 2 \longleftrightarrow 3 \longleftrightarrow 4 \longleftrightarrow \dots$$

This process may serve as a simple model for mRNA dynamics (with constitutive transcription and linear degradation), but it can also describe how molecules enter or leave a spatial volume by diffusion. Mathematically, we describe it as a Markov process (see Section 15.4) with continuous time and transitions

Event	Transition	Propensity	(7.4)
Production	$x \rightarrow x + 1$	$a_+(x) = w_+$	
Degradation	$x \rightarrow x - 1$	$a_-(x) = w_- x$	

If the system is in a state $x(t)$ at time point t , the next reaction event can in principle occur at any moment. To quantify the probabilities for different possible events, we consider a short time interval $[t, t + \Delta t]$. In the limit $\Delta t \rightarrow 0$, each possible event $j = +$ (production) or $j = -$ (degradation) can occur with probability $p_j \approx a_j \Delta t$ and the chances for double events to occur in the interval can be neglected.

In Eq. (7.4), we distinguish between two kinds of rates: the *propensities* $a_{\pm}(t)$ refer to the absolute rate of reaction events (for all molecules together), while the degradation rate w_- refers to single molecules. The propensity $a_-(x)$ is proportional to x because each of the existing x molecules can be degraded with a rate w_- ; it vanishes if no molecule is present. Production events, on the other hand, occur with a constant propensity $a_+ = w_+$ (in units of s^{-1}). While the rates w_{\pm} are constant, the propensity a_- changes in time because it depends on the system state $x(t)$. The waiting time between subsequent reaction events follows an exponential distribution with mean value $(a_+(x) + a_-(x))^{-1}$. Statistical properties of such random processes can be computed with the help of generating functions (see Section 15.4).

The birth–death process (7.4) gives rise to a time-dependent probability distribution $p_x(t)$ in the state space, assigning a probability to each possible molecule number x . For a large ensemble of cells, this probability corresponds to the percentage of cells with molecule number x at time t . The distribution $p_x(t)$ changes in time according to the chemical master equation

$$\frac{dp_x}{dt} = a_+(x-1)p_{x-1} + a_-(x+1)p_{x+1} - a_+(x)p_x - a_-(x)p_x \quad (7.7)$$

The terms on the right-hand side correspond to the four possible events that can change the probability of state x : the system can jump to state x from the state $x - 1$ (production) or $x + 1$ (degradation), and it can leave state x by the production or degradation of a molecule.

To solve Eq. (7.7), we need to specify the initial condition: if the system starts with precisely one molecule at $t = 0$, we set $p_1(0) = 1$ and $p_j(0) = 0$ for all $j \neq 1$. The stationary distribution of Eq. (7.7) is a Poisson distribution $p(x, \lambda) = \lambda^x e^{-\lambda} / x!$ with average value and variance

$$\lambda = \langle x \rangle = \text{var}(x) = \frac{w_+}{w_-}. \quad (7.8)$$

Example 7.4 Degradation of a Single Particle

As a simple example, we consider the degradation of a single particle. Our simple two-state system contains a state x_1 , in which the particle is present, and a state x_0 , in which it has disappeared. The transition $x_1 \rightarrow x_0$ occurs with fixed propensity a . The time-dependent probabilities for both states follow the rate equations (called “master equation”)

$$\begin{aligned} \frac{dp(0,t)}{dt} &= a p(1,t), \\ \frac{dp(1,t)}{dt} &= -a p(1,t). \end{aligned} \quad (7.5)$$

If the particle is present at time $t = 0$, we can use the initial condition $p(0,t) = 0, p(1,t) = 1$, and Eq. (7.5) has the solution

$$\begin{aligned} p(0,t) &= 1 - e^{-at}, \\ p(1,t) &= e^{-at}. \end{aligned} \quad (7.6)$$

In each realization of the process, the particle will decay at a certain time t_0 . For different realizations, that is, many independent particles, these times t_0 follow an exponential distribution $p(t_0) = a e^{-at_0}$ with time constant $\tau = 1/a$. Such exponential waiting time distributions also occur in more complicated Markov processes. Considering a large ensemble of independent particles, we could describe the average amount of nondegraded particles by deterministic concentration curves as shown in Figure 4.1.

If the mean molecule number is large, the numbers in different realizations of the random process will differ only little and the random process (7.4) can be approximated by a deterministic kinetic model

$$\frac{ds}{dt} = v_+ - v_- = \alpha - \beta s \quad (7.9)$$

for the ensemble-average concentration s (particle number per volume V). The stoichiometric matrix $N = (1, -1)$ and the velocity vector $v = (\alpha, \beta s)^T$ correspond to the events and rates in model (7.4), and the parameters of the two models are related by $V\alpha = w_+$ and $\beta = w_-$. The kinetic model (7.9) yields a steady-state concentration $s^{st} = \alpha/\beta$ corresponding to a molecule number $x^{st} = V\alpha/\beta$. In linear models such as this, the result of the kinetic model represents the ensemble average value (7.8) of the stochastic process. Moreover, the relative spread $\sqrt{\text{var}(x)/\langle x \rangle}$ of molecule numbers in the stochastic ensemble decreases if we increase the volume $V \rightarrow \infty$ at a fixed mean concentration. In the *thermodynamic limit* (infinite volume), it goes to zero. Therefore, the

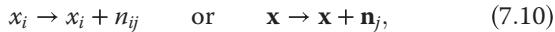
kinetic model can be seen as a limiting case of the stochastic model (7.4) for infinite volumes V .

7.2.2

The Chemical Master Equation

Our birth–death process for a single molecule species can be extended to more complicated reaction systems [16]. If reactions are reversible, they must be split into irreversible forward and backward reactions. To translate a kinetic model into a stochastic process, we define the system states by a particle number vector \mathbf{x} , specify the propensities $a_j(\mathbf{x})$ for state transitions, and determine the chemical master equation. In a well-mixed system, substrate molecules hit each other with rates proportional to the substrate abundances; this leads to a mass action law for the propensities. We can describe complex reactions by breaking them down into elementary steps following mass action rate laws or, approximately, by computing the propensities by effective (e.g., Michaelis–Menten) rate laws [17].

Let us now consider a biochemical reaction system with m substance species S_i , irreversible reactions R_j , and the stoichiometric matrix N . The system state, a vector $\mathbf{x} \in \mathbb{N}^m$, contains the molecule numbers of all substances, and a reaction R_j changes the molecule numbers by



where \mathbf{n}_j is the j th column of N . As above, we assume that reactions occur randomly with propensities a_j ; to compute them, we consider a well-stirred mixture of molecules in a volume V . Formulas for the propensities follow from kinetic theory: each reaction rate depends on how many substrate molecules are available, how often the substrates of a reaction come into close vicinity, and how often such a contact leads to a successful reaction. A unimolecular reaction R_j with a single substrate S_i occurs with a constant rate w_j per substrate molecule, leading to a propensity $a_j(\mathbf{x}) = w_j x_i$. The rate w_j (measured in units of s^{-1}) is identical to the mass action rate constant k_j in the corresponding kinetic model, irrespective of the system volume. For other reaction stoichiometries, k_j has to be rescaled with the system volume in order to obtain w_j . Propensities for different types of reactions and their relations to rate constants are listed in Table 7.4. If a reaction R_j is catalyzed by a substance S_i , the formula for the propensity $a_j(\mathbf{x})$ has to be modified: the molecule number x_i appears as a prefactor in a_j , and the scaling with V changes accordingly.

Like in the process (7.4), the time-dependent distribution $p(x_1, \dots, x_m, t)$ of system states is governed by a

Table 7.4 Rates for reactions with different stoichiometries (no substrate, one substrate, two substrates, and two substrate molecules of the same type).

Reaction	Formula	Propensity	Scaling
No substrate	$\rightarrow \dots$	$a_j = w_j$	$w_j = k_j V$
Unimolecular	$A \rightarrow \dots$	$a_j = w_j x_A$	$w_j = k_j$
Bimolecular	$A + B \rightarrow \dots$	$a_j = w_j x_A x_B$	$w_j = k_j / V$
Bimolecular	$A + A \rightarrow \dots \dots$	$a_j = w_j x_A (x_A - 1)$	$w_j = 2k_j / V$

chemical master equation:

$$\frac{dp(\mathbf{x}, t)}{dt} = \sum_j a_j(\mathbf{x} - \mathbf{n}_j) p(\mathbf{x} - \mathbf{n}_j, t) - \sum_j a_j(\mathbf{x}) p(\mathbf{x}, t). \quad (7.11)$$

In this notation, the states \mathbf{x} appear symbolically as a function argument; alternatively, they could also be written as discrete subscripts (e.g., $p_x(t)$). Each state \mathbf{x} gives rise to one differential equation (7.11), and the entire equation system is called the chemical master equation. If we consider process (7.4) as a simple biochemical reaction system, we can easily recognize Eq. (7.7) as a special case of Eq. (7.11). The positive term in Eq. (7.11) enumerates all realizations that start in state $\mathbf{x} - \mathbf{n}_j$ and lead to state \mathbf{x} via a reaction R_j ; the negative term collects all realizations that exit state \mathbf{x} . The propensities for impossible states $\mathbf{x} - \mathbf{n}_j$ with negative molecule numbers are defined as zero. For analytical solutions of the chemical master equation, see Ref. [18].

7.2.3

Stochastic Simulation

A stochastic simulation (or “Monte Carlo simulation”) yields individual realizations of a random process. Just like a random number generator, which produces realizations of a random variable, stochastic simulations draw possible histories of a system according to their probabilities (or probability densities) in the random process. Stochastic simulation of biochemical systems can be run within software tools (see Chapter 17). D. Gillespie has shown how different simulation methods can be applied to chemical random models with continuous time and discrete particle numbers [19,20]. The methods, however, are older and apply to Markov jump processes in general (see Section 15.4).

7.2.3.1 Direct Method

The *direct method* [19] simulates a series of state transitions (e.g., chemical reaction events in continuous time).

First, we choose an initial state $x(t = 0)$. Then, individual transitions are simulated as follows:

- 1) Determine all possible transitions R_j starting in x and compute their propensities $a_j(x)$.
- 2) Decide when the next transition will happen: the waiting time Δt is drawn from an exponential random distribution with characteristic time $\tau = 1/\sum_j a_j(x)$. To see why the waiting time should be exponentially distributed, we can compare our transition with a degradation process as in Example 7.4. Since we are only interested in *when* the system leaves its current state, and not *how*, the rates for all possible transitions can simply be added.
- 3) Decide which transition will happen: one of the possible transitions is chosen randomly; the propensities $a_j(x)$ are used as probability weights, that is, probabilities for the transitions are calculated as $p_j = a_j(x)/\left(\sum_j a_j(x)\right)$, where the sum runs over all possible transitions starting in x .
- 4) Update the system state x and the time t , and go back to step 1 to simulate the next transition.

This procedure is iterated until the end time T has been reached. By running many simulations with different random seeds, we can obtain a large set of realizations; from a statistics over the realizations, we can then estimate statistical properties of the process such as average behavior, time correlations, or probabilities for specific qualitative behavior. The direct simulation method is numerically expensive if certain transitions occur very frequently while others are rare. In this case, it would take many steps until these rare reactions have occurred in considerable numbers. This resembles the problems caused by stiff differential equations (see time scale separation in Section 6.3) and can be avoided by using the explicit τ -leaping method or hybrid methods in which fast reactions are effectively described by ordinary differential equations [21].

7.2.3.2 Explicit τ -Leaping Method

In the explicit τ -leaping method [22,23], we do not simulate individual events, but consider time intervals of fixed length τ and determine the number of reaction events in each time interval. As an approximation, we need to assume that all propensities remain constant within each time interval. Under this assumption, the numbers of reaction events will follow independent Poisson distributions and we can simulate them by Poisson-distributed random numbers. To compute an entire history of our system, we choose an initial state; then we iteratively compute the reaction propensities, choose the numbers of reaction events for the next interval (by drawing them from a Poisson distribution), and update the system state.

By iterating this procedure, we obtain a complete stochastic simulation.

The efficiency and the approximation error of the explicit τ -leaping method depend on the interval size of τ : if time intervals are large, many reaction events occur in an interval, so the τ -leaping method becomes much faster than the direct method. However, the assumption of constant propensities within intervals becomes less justified, and so the approximation error increases. The τ -leaping method works best for large particle numbers: then, even large numbers of reaction events will not considerably change the propensities.

7.2.3.3 Stochastic Simulation and Spatial Models

To justify the use of random models for chemical reactions, we assumed a well-stirred mixture in which molecules move freely and diffusion is much faster than chemical reactions. In reality, compartment structure, special geometries, and heterogeneous distribution of substances can affect the overall reaction rates. Stochastic methods can be used to model spatially heterogeneous systems: we just split the cell volume into many spatial volume elements and describe the cell state by the molecule numbers in each element. Diffusion on a larger spatial scale is modeled explicitly by transitions between volume elements; within the elements, we assume fast mixing. The state space of such spatial models can be enormous: to describe m molecular species by their numbers in n volume elements, we need a state vector of length $m \cdot n$. Instead of describing molecules by particle numbers, we may also track the fate of individual molecules. If the number of molecules is small, Gillespie's algorithm, with a state vector describing their detailed positions and conformations, can be used to simulate their dynamics.

7.2.4

Chemical Langevin Equation and Chemical Noise

In the stochastic model given by Eq. (7.10), molecules are described by particle numbers. If these numbers are large, one can approximate them by real-valued variables x_i that follow a Brownian motion-like dynamics in state space [16] as described by the *chemical Langevin equation* (see Section 15.4)

$$\frac{dx_i(t)}{dt} = \sum_j n_{ij} a_j(x(t)) + \sum_j n_{ij} \sqrt{a_j(x(t))} \xi_j(t). \quad (7.12)$$

The term $\xi_j(t)$ denotes Gaussian white noise. Mathematically, white noise is a random process with mean $\langle \xi_j \rangle = 0$

and covariance function $\langle \xi_j(t_1) \xi_j^T(t_2) \rangle = \delta_{jl} \delta(t_1 - t_2)$ with Kronecker's $\delta_{jl} = 1$ and Dirac's $\delta(t)$ distribution. The time integral of the white noise process yields Gaussian-distributed random numbers, which makes Eq. (7.12) easy to simulate.

Equation (7.12) can be derived by a series of three approximations: (i) Like in the τ -leaping method, reaction events are counted within small time intervals τ . The rates a_j are assumed to be constant within each time interval, so the numbers of reaction events follow a Poisson distribution. (ii) If the numbers of reactions in a time interval are large, this Poisson distribution can be further approximated by a Gaussian distribution. In a time interval of length τ , the (random) number of reaction events of type j now reads

$$R_j = \bar{r}_j + \sqrt{\bar{r}_j} \eta_j = a_j \tau + \sqrt{a_j \tau} \eta_j, \quad (7.13)$$

where \bar{r}_j is the average number of reaction events, a_j is the propensity, and η_j is a standard Gaussian-distributed random number. Within the time interval, the particle numbers will change by

$$x_i(t + \tau) - x_i(t) = \sum_j n_{ij} R_j = \sum_j n_{ij} [a_j \tau + \sqrt{a_j \tau} \eta_j]. \quad (7.14)$$

Dividing this by the time interval itself, we obtain

$$\frac{x_i(t + \tau) - x_i(t)}{\tau} = \sum_j n_{ij} a_j + \sum_j n_{ij} \sqrt{a_j} \frac{\eta_j}{\sqrt{\tau}}. \quad (7.15)$$

(iii) Finally, the stochastic process (7.15), with discrete time steps and Gaussian-distributed jumps, can be replaced by the continuous Langevin equation (7.12) with additive white noise (see Section 15.4). Thus, we return to a process in continuous time, but with continuous levels instead of discrete molecule numbers. To justify the approximations (i) and (ii), the molecule numbers must be high: in this case, many reaction events can happen within an interval τ without too much changing the propensities.

To arrive at a macroscopic description, we describe the substances by concentrations $s_i = x_i/V$ (in units of molecules per volume, not moles per volume) and reactions by velocities $v_j(\mathbf{s}(\mathbf{x})) = a_j(\mathbf{x})/V$. After dividing Eq. (7.12) by the system volume V , we obtain the chemical Langevin equation for concentrations:

$$\frac{ds_i(t)}{dt} = \sum_j n_{ij} v_j(s(t)) + \sum_j n_{ij} \sqrt{v_j(s(t))/V} \xi_j(t). \quad (7.16)$$

It resembles a kinetic model, but each reaction rate contains an additive noise term. The noise amplitude scales

with the square root of the mean reaction rate and inversely with the square root of the volume. If the system volume increases while the mean concentrations are kept fixed, the noise term contributes less and less, and in the limit of very large systems, it becomes arbitrarily small.

Thus, if a kinetic model defines a deterministic trajectory in state space, the noise term in the chemical Langevin equation will lead to deviations from this trajectory. If the dynamics along the trajectory is stable (as indicated by the eigenvalues of the Jacobian matrix) and if particle numbers are high, the system will show small fluctuations around the mean trajectory. For a macroscopic steady state, these fluctuations can be computed using a linear approximation: given the reaction velocities $v_j = a_j/V$, we linearize Eq. (7.12) for molecule numbers and obtain

$$\frac{d\mathbf{x}}{dt} \approx \mathbf{A}\mathbf{x} + V\mathbf{B}\xi \quad (7.17)$$

with the Jacobian $\mathbf{A} = \mathbf{N}\tilde{\boldsymbol{\epsilon}}$ (see Section 15.2) and the matrix $\mathbf{B} = V^{-1/2}\mathbf{N}Dg(\mathbf{v})^{1/2}$. Likewise, Eq. (7.16) for concentrations yields

$$\frac{d\mathbf{s}}{dt} \approx \mathbf{A}\mathbf{s} + \mathbf{B}\xi. \quad (7.18)$$

The same elasticity matrix links the concentrations to reaction rates ($\tilde{\boldsymbol{\epsilon}} = \partial\mathbf{v}/\partial\mathbf{c}$) and the molecule numbers to propensities ($\tilde{\boldsymbol{\epsilon}} = \partial\mathbf{a}/\partial\mathbf{x}$). Given the matrices \mathbf{A} and \mathbf{B} , we can compute the covariance matrix $\mathbf{Q} = \text{cov}(\mathbf{x})$ for molecule number fluctuations from the Lyapunov equation (see Section 15.5)

$$\mathbf{A}\mathbf{Q} + \mathbf{Q}\mathbf{A}^T + \mathbf{B}\mathbf{B}^T = 0. \quad (7.19)$$

From the covariance matrix \mathbf{Q} , we obtain individual noise levels for all substances (diagonal elements of \mathbf{Q}) and linear correlations between pairs of substances. This method to derive mean values and covariances is called the *linear noise approximation* [24–27].

Equation (7.19) shows that upon a rescaling $\mathbf{A} \rightarrow \lambda\mathbf{A}$, the covariance matrix scales as $\mathbf{Q} \rightarrow 1/\lambda\mathbf{Q}$; hence, if the steady state is very stable (large negative eigenvalues of the Jacobian), the fluctuations will be small. For large volumes V , the noise matrix \mathbf{B} becomes small, and the error of the linear approximation diminishes. For smaller volumes, in contrast, the lower particle numbers lead to larger fluctuations, which can, for example enable bistable systems to escape from their current steady state. In this case, the above approximations break down [27,28]. Thus, due to fluctuations bistable system can switch randomly between their stable states (see Section 10.3.5).

7.2.5

Dynamic Fluctuations

The chemical Langevin equation (7.12) describes the intrinsic noise caused by stochastic reaction events. Langevin equations can also be used to describe extrinsic noise, that is, perturbations caused by the system's environment. If the parameter vector $\mathbf{k}(t)$ in a kinetic model

$$\frac{ds(t)}{dt} = \mathbf{N}\mathbf{v}(s(t), \mathbf{k}(t)) \quad (7.20)$$

follows a random process, the substance levels $s(t)$ will be affected and will behave randomly. To compute their statistical properties, we assume that the equation system (7.20), with constant parameters \mathbf{k}_0 , has a stable steady state with concentrations $s^{st}(\mathbf{k})$. We linearize Eq. (7.20) around this reference state, consider fluctuating deviations $\mathbf{u}(t) = \mathbf{k}(t) - \mathbf{k}_0$ and $\mathbf{x}(t) = s(t) - s^{st}(\mathbf{k}_0)$, and obtain the approximation

$$\frac{dx(t)}{dt} = \mathbf{N}\tilde{\mathbf{e}}\mathbf{x}(t) + \mathbf{N}\tilde{\pi}\mathbf{u}(t) \quad (7.21)$$

with substrate elasticity matrix $\tilde{\mathbf{e}} = \partial\mathbf{v}/\partial s$ and parameter elasticity matrix $\tilde{\pi} = \partial\mathbf{v}/\partial p$. By collecting all concentrations and fluxes in an output vector

$$\mathbf{y} = \begin{pmatrix} s(t) - s^{st} \\ v(t) - v^{st} \end{pmatrix} = \begin{pmatrix} \mathbf{I} \\ \tilde{\mathbf{e}} \end{pmatrix} \mathbf{x} + \begin{pmatrix} 0 \\ \tilde{\pi} \end{pmatrix} \mathbf{u}, \quad (7.22)$$

and abbreviating the matrices in Eqs. (7.21) and (7.22) by \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} , we can rewrite our model in the standard form

$$\begin{aligned} \frac{dx(t)}{dt} &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y} &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t) \end{aligned} \quad (7.23)$$

used in control theory (see Section 15.5). The propagation of noise in linear, time-invariant systems of this form is described by the frequency response function (see Section 15.5)

$$\mathbf{H}(i\omega) = -\mathbf{C}(\mathbf{A} - i\omega\mathbf{I})^{-1}\mathbf{B} + \mathbf{D}. \quad (7.24)$$

If the system output \mathbf{y} represents metabolic concentrations or fluxes, the frequency responses are given, respectively, by the spectral response matrices [29,30]

$$\begin{aligned} \tilde{\mathbf{R}}^S(\omega) &= -(\mathbf{N}\tilde{\mathbf{e}} - i\omega\mathbf{I})^{-1}\mathbf{N}\tilde{\pi}, \\ \tilde{\mathbf{R}}^J(\omega) &= (\mathbf{I} - \tilde{\mathbf{e}}(\mathbf{N}\tilde{\mathbf{e}} - i\omega\mathbf{I})^{-1}\mathbf{N})\tilde{\pi}. \end{aligned} \quad (7.25)$$

Like the response matrices for steady states, the matrices (7.25) characterize the response to small perturbations, but at finite frequencies ω . Their complex-valued elements describe the amplitudes and phases of output variables relative to parameter perturbations (compare with Section 15.5).

Example 7.5 Minimal Biochemical System with Hopf Bifurcation

Figure 7.7 shows the minimal biochemical reaction system with a Hopf bifurcation. In Hopf bifurcations, a stable steady state becomes unstable and oscillations arise [31] (see Section 15.2). The rate equations for metabolite concentrations a , b , and c in the model read

$$\begin{aligned} \frac{da}{dt} &= (k_1x - k_4)a - k_2ac, \\ \frac{db}{dt} &= k_4a - k_5b, \\ \frac{dc}{dt} &= k_5b - k_3c. \end{aligned} \quad (7.27)$$

The external substrate concentration x acts as a bifurcation parameter: as long as x is small, the system has a stable steady state, but when the critical value x_{crit} is reached, the state becomes unstable and the metabolite levels start oscillating at a frequency ω_0 . This change from a stable to an unstable focus is known as a *supercritical Hopf bifurcation*.

However, the system reveals its tendency to oscillate already before the bifurcation is reached: when approaching the bifurcation point, it becomes more and more susceptible to periodic perturbations and starts to amplify parameter oscillations and noise at frequencies around ω_0 [30]. Hence, the spectral densities of metabolite fluctuations caused by intrinsic chemical noise show a resonance peak near the frequency ω_0 (see Figure 7.8a).

The bifurcation and the associated resonance can be explained by eigenvalues of the Jacobian. Figure 7.8b shows the eigenvalues as points in the complex plane: in a stable steady state, all eigenvalues have negative real parts and are therefore located in the left half-plane. Close to the Hopf bifurcation – where x is just below its critical value – a

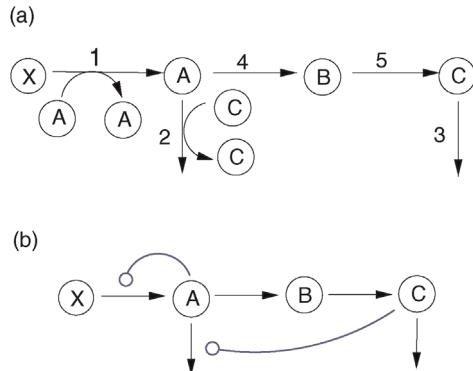


Figure 7.7 The minimal biochemical system with Hopf bifurcation [31]. (a) Network scheme. If the external substrate X exceeds its critical level, the levels of A , B , and C show sustained oscillations. (b) A simplified representation shows two feedback loops, which are responsible for the oscillation: A activates its own production and, with a delay, its own degradation.

pair of complex eigenvalues comes close to the imaginary axis. These eigenvalues are responsible for the resonance behavior: according to Eq. (7.25), the spectral response coefficients contain the term $\mathbf{A} - i\omega \mathbf{I}$, in which the eigenvalues of \mathbf{A} are shifted down by $i\omega \mathbf{I}$. At a frequency $\omega \approx \omega_0$, the upper eigenvalue comes close to the origin, giving rise to a very large eigenvalue of the inverse matrix $(\mathbf{A} - i\omega \mathbf{I})^{-1}$ (see Figure 7.8c). This amplification leads to a resonance peak in the spectral response matrix and, as a consequence, in the spectral densities.

If the noisy parameters follow a stationary Gauss–Markov process with mean values \mathbf{k}_0 and small fluctuation amplitudes (described by a spectral density matrix $\Phi_u(\omega)$), the spectral density matrix Φ_y of the output \mathbf{y} can be computed from the frequency response function as

$$\Phi_y(\omega) = \mathbf{R}(\omega)\Phi_u(\omega)\mathbf{R}(\omega)^\dagger, \quad (7.26)$$

where the symbol \dagger denotes the adjoint (conjugate transpose) matrix (see Section 15.5). In particular, if parameter fluctuations follow a white noise process (i.e., with spectral density $\Phi_u(\omega) = \mathbf{I}$), the spectral density matrix of substance concentrations reads $\Phi_y(\omega) = \mathbf{R}(\omega)\mathbf{R}(\omega)^\dagger$.

Figure 7.8b also illustrates another general feature of biochemical systems. If a system is driven by periodic perturbations, then at high frequencies (above the highest resonance frequency), all eigenvalues of $(\mathbf{A} - i\omega \mathbf{I})^{-1}$ will decrease. Therefore, linearized biochemical models act as low-pass filters for periodic perturbations or noise.

7.2.6 From Stochastic to Deterministic Modeling

Biochemical reaction systems can be described by stochastic or deterministic models. Since stochastic simulations

are usually more expensive to perform, they should only be used when random fluctuations matter. For well-mixed systems of infinite volume (i.e., infinite particle numbers), the fluctuations vanish and the random process is well approximated by a macroscopic deterministic model. At finite particle numbers, random fluctuations may affect the microscopic dynamics. If the dynamics is linear, the macroscopic model represents ensemble averages of the random process. In nonlinear systems, fluctuations may lead to macroscopic effects, for instance, enable bistable systems to jump randomly between different steady states.

As a last example, Figure 7.9 shows simulation results from deterministic and stochastic simulations of a simple gene expression model. In the deterministic macroscopic model, mRNA concentration x and protein concentration y follow the rate equations

$$\begin{aligned} \frac{dx}{dt} &= k_{+x} - k_{-x}x, \\ \frac{dy}{dt} &= k_{+y}x - k_{-y}y. \end{aligned} \quad (7.28)$$

For simplicity, we assume a volume $V = 1$, so concentrations in the deterministic model (measured as molecules per volume) correspond to molecule numbers. This model will be studied again in Section 9.4 on stochastic gene expression.

7.3 Spatial Models

Summary

Spatial structure is essential for the functioning of cells and organisms. Spatiotemporal dynamics can be modeled in a number of mathematical frameworks, including compartment models, reaction–diffusion equations, and stochastic agent-based models. Biochemical systems can establish, maintain, control, and adapt spatial

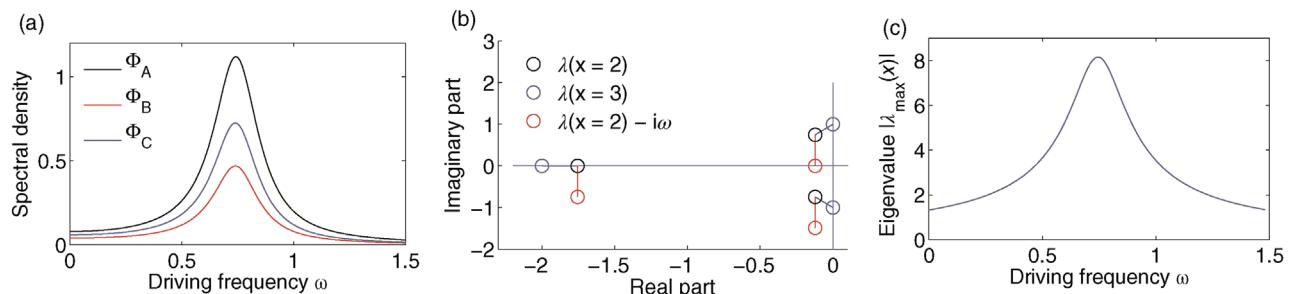


Figure 7.8 Resonance in the minimal biochemical system with Hopf bifurcation (Figure 7.7). (a) Spectral densities of metabolite levels in the presence of chemical noise. (b) Eigenvalues of the Jacobian matrix in the complex plane. Black and blue circles show, respectively, eigenvalues for a bifurcation parameter below and at the bifurcation point. The term $-i\omega$ in the spectral response coefficient shifts the upper-right eigenvalue λ^* (black) toward the origin (red). (c) Resonance curve of $|(\lambda^* - i\omega)^{-1}|$. All quantities in arbitrary units. (From Ref. [30].)

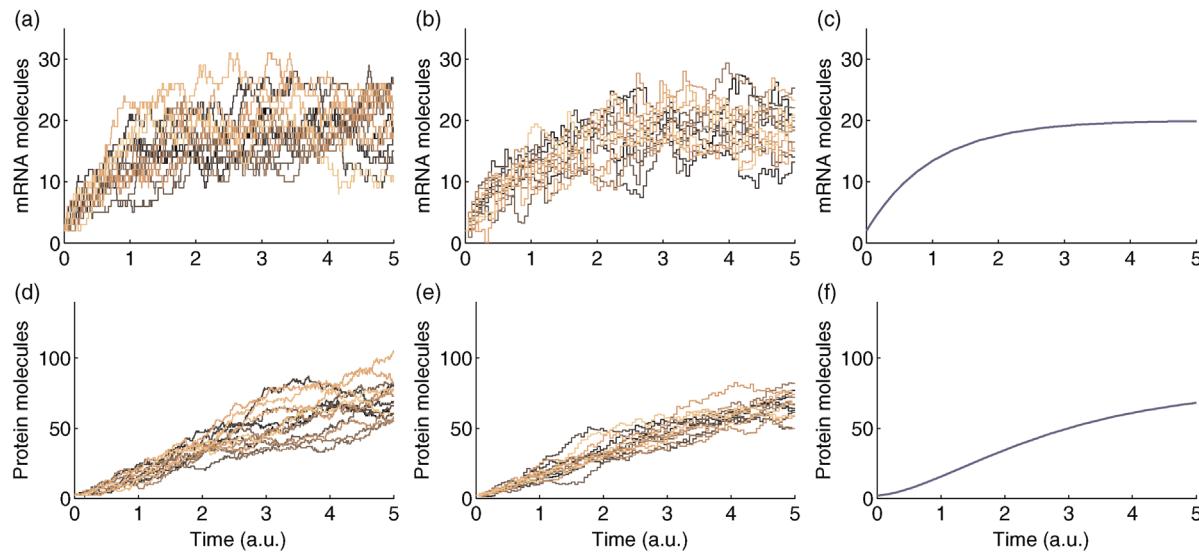


Figure 7.9 Simulations of molecule numbers for an mRNA species (a–c) and the corresponding protein (d–f). Time series were computed with Gillespie’s direct method (a, d), chemical Langevin equation (b, e), and a kinetic model (c, f). For the stochastic models, 10 realizations are shown in different colors. Parameters: $k_x^+ = 20$, $k_x^- = 1$, $k_y^+ = 2$, and $k_y^- = 5$. Initial molecule numbers $x = 2$, $y = 2$, and volume $V = 1$.

structure in a self-organized and robust manner. The body plan of animals, for instance, is shaped during embryonic development in a self-organized manner by the spatiotemporal dynamics of morphogen levels. In reaction–diffusion systems, dynamic instabilities in a homogeneous state can give rise to spontaneous pattern formation. Realistic spatial simulation is numerically demanding, but relatively simple two-substance models can suffice to study characteristic behavior such as waves or pattern formation.

Cells, tissues, and organisms show complex spatial structures, which are essential to the processes of life. Membranes allow cells to maintain different biochemical environments and to establish gradients of concentrations or chemical potential, for example, proton gradients that serve as an energy storage in bacteria and mitochondria. Cell organelles provide suitable environments for different biochemical processes (e.g., presence of hydrolases and low pH in lysosomes). Vesicles, moving along the cytoskeleton, can transport substances between organelles. Eukaryotic cells can become polarized, that is, develop special structures on one side of the cell (e.g., the shmoo tips in budding yeast) in response to external stimuli [32]. Molecule localization also plays a role in signaling: in the Jak–Stat pathway, for instance, active Stat proteins accumulate in the nucleus to induce transcriptional changes. On the molecular scale, scaffold proteins hold together protein complexes to coordinate their enzymatic activity; channeling, in which metabolites are

directly passed from enzyme to enzyme, can increase enzymatic efficiency.

Many biological processes involve spatiotemporal dynamics. Prominent examples are calcium waves within cells, action potentials in the brain, patterning during embryonic development, gradients of secreted signaling molecules [33], or the invasion of tissues by cancer cells. Such spatiotemporal processes can be modeled in mathematical frameworks such as compartment models and reaction–diffusion models [34], which are treated in this section.

A recurrent question in spatial modeling concerns self-organization: how can complex structures emerge and be maintained through local interactions between molecules, that is, without being centrally coordinated? For instance, how can organelles develop and maintain different identities, marked by protein composition [35,36], and how can cells become spontaneously polarized? Models of reaction–diffusion systems show how patterns arise spontaneously, how they can be controlled via biochemical molecule parameters, and what mechanism makes patterning robust – which is essential, for instance, in embryonic development.

7.3.1

Types of Spatial Models

How can spatial substance distributions be described by models? If molecules move freely and independently and if diffusion is much faster than chemical reactions,

inhomogeneities will rapidly fade away, so substances can be described by their average cellular concentrations. If diffusion is slow or hindered by membranes, molecules will be inhomogeneously distributed and their localization within cell structure needs to be modeled.

- 1) *Compartment models.* In compartment models, substances are described by homogeneous concentrations within compartments, for example, organelles of a cell. In pharmacokinetics, compartments represent organs between which drugs can be distributed [37,38]. Compartment models are based on the assumption of fast diffusion within each compartment; if biological compartments resemble each other (e.g., the mitochondria in a cell) or if there is a rapid mixing between them, they can be treated as a single effective compartment.
- 2) *Reaction-diffusion systems.* If diffusion is slow, it will not wear away spatial concentration inhomogeneities even within compartments: on the contrary, a coupling between diffusion and chemical reactions may generate patterns (e.g., gradients or waves). The dynamics of such patterns can be described by *reaction-diffusion systems*, partial differential equations that describe substance concentrations in continuous space and time. The equations can be solved numerically by splitting space into *finite elements*, small volume elements that are treated like compartments, but are chosen on the basis of numerical requirements.
- 3) *Stochastic models.* Partial differential equation models assume that concentrations are smooth functions in space. This, however, holds only on a scale much larger than the average distance between molecules. If substance concentrations are small, the thermal movement and chemical reactions of individual molecules can be simulated by stochastic models. A stochastic simulation may track individual molecules, describing their diffusion by random walks. If two molecules meet, they can react and be converted into product molecules. The numerical effort of particle simulations can be high, especially if many particles are involved. An alternative to tracking individual particles is to split the cell into subvolumes and to simulate particle numbers within subvolumes by random processes describing reaction and diffusion (see Section 7.2.3).
- 4) *Cellular automata.* Cellular automata simulate temporal or spatiotemporal processes in discrete space and time [39]. A cellular automaton consists of a regular lattice of components called cells, which can assume a finite number of states. Cell states switch

synchronously in discrete time steps and according to deterministic or stochastic rules. The rules are usually local, that is, accounting only for the state of a cell and its neighbors. Cellular automata models were invented in the late 1940s by von Neumann and Ulam. A prominent example is Conway's *game of life* [40]. In this simple model, cells on a square lattice can either be "dead" (or 0) or "alive" (or 1). The states are updated synchronously according to the following rules: a living cell with two or three living neighbors remains alive; a living cell with less than two or more than three living neighbors dies; if a dead cell has exactly three living neighbors, it comes to life, otherwise it remains dead. These rules give rise to a surprisingly rich dynamic behavior. Cellular automata can, for instance, be used to simulate the proliferation of cells and organisms in spatial settings.

7.3.2 Compartment Models

In compartment models, substance concentrations can differ between compartments, but are homogeneous within them. Transport between compartments, for example, diffusion across membranes, is modeled by transport reactions, while passive exchange through membranes or pores may be described as diffusion with a rate

$$\nu^*(s_1, s_2) = A P(s_1 - s_2). \quad (7.29)$$

The permeability P (in m s^{-1}) depends on the physicochemical properties of membrane, channels, and the diffusing molecules. A denotes the membrane area, and indices 1 and 2 refer to compartments. Active transport by transporter proteins may be modeled by saturable rate laws, for instance, irreversible Michaelis–Menten kinetics. Importantly, transport rates are measured as *amounts per time* (in mol s^{-1}), but depend on compound *concentrations* in mM (e.g., the difference $s_1 - s_2$ in Eq. (7.29)).

Compartment models resemble usual kinetic models, but some attention needs to be paid to the conversion between substance amounts, concentrations, and compartment volumes. Moreover, if we account for thermodynamics, pH values and electric potentials may differ between compartments, implying different standard chemical potentials for the same substance [41]. This will affect the reaction Gibbs free energies and, thereby, the kinetics of transport reactions.

Compartment sizes can affect the biochemical dynamics via dilution effects. To obtain rate equations accounting for dilution, we first formulate them in terms of amounts a_i (where the subscript i indicates a substance located in a compartment). With the reaction velocities

v^* (in mol s $^{-1}$), we obtain the rate equation

$$\frac{da_i}{dt} = \sum_j n_{ij} v_j^*(s), \quad (7.30)$$

where n_{ij} are the stoichiometric coefficients. Each substance amount a_i is defined in a compartment with index $k(i)$ and a volume $V_{k(i)}$ (in m 3). After introducing concentrations $s_i = a_i/V_{k(i)}$, we can rewrite the left-hand side of Eq. (7.30) as

$$\frac{da_i}{dt} = \frac{d}{dt}(V_{k(i)}s_i) = V_{k(i)}\frac{ds_i}{dt} + \frac{dV_{k(i)}}{dt}s_i. \quad (7.31)$$

By combining Eqs. (7.30) and (7.31), we obtain a rate equation for the concentrations

$$\frac{ds_i}{dt} = \sum_j \frac{n_{ij}}{V_{k(i)}} v_j^*(s) - \frac{dV_{k(i)}/dt}{V_{k(i)}} s_i. \quad (7.32)$$

The terms on the right-hand side show that concentration changes can arise in two ways: by chemical reactions or transport (first term) and by temporal volume changes (second term). If all compartments in Eq. (7.32) have identical, time-independent volumes, we can replace $v_j^*/V_{k(i)}$ by the usual reaction velocity v_l in mM s $^{-1}$, and since the second term vanishes, we obtain the usual form of kinetic models.

What if compartments have different sizes? If their volumes are constant in time, the second term still vanishes and we obtain, from the transport reaction alone,

$$V_1 \frac{ds_1}{dt} = -V_2 \frac{ds_2}{dt}. \quad (7.33)$$

The minus sign stems from the stoichiometric coefficient. Dilution effects play an important role in transport between cells and the external medium: intra- and extracellular concentration changes are converted to each other by the volume ratio $V_{\text{cell}}/V_{\text{ext}}$, where V_{cell} is the volume of a single cell and V_{ext} is the extracellular volume divided by the number of cells.

Finally, the effects of temporal volume changes are described by the second term in Eq. (7.32): substances in growing cells are diluted, so their concentration will decrease even if they are not degraded by chemical reactions. If a cell population grows at a rate $\kappa(t)$, the total cell volume V increases as

$$\frac{dV}{dt} = \kappa(t)V, \quad (7.34)$$

so the fraction in the second term in Eq. (7.32) is exactly given by the growth rate $\kappa(t)$. Thus, dilution of molecules in growing cells resembles linear degradation, with the cell growth rate κ as an effective degradation constant.

7.3.3

Reaction–Diffusion Systems

7.3.3.1 Diffusion Equation

The diffusion of substances in homogeneous media can be modeled by a partial differential equation, called *diffusion equation*, describing their space- and time-dependent concentrations $s(\mathbf{r}, t)$. Space coordinates are represented by three-dimensional vectors $\mathbf{r} = (x, y, z)^T$, but we often consider a single space dimension only, for instance, if we can assume homogeneity along the other two space dimensions. The flow of a substance is described by a vectorial flow field $\mathbf{j}(\mathbf{r}, t)$; we can regard the flow as a product $\mathbf{j}(\mathbf{r}, t) = s(\mathbf{r}, t)\mathbf{w}(\mathbf{r}, t)$, where $\mathbf{w}(\mathbf{r}, t)$ is the local average particle velocity. If a substance is neither produced nor degraded, its concentration obeys the continuity equation

$$\frac{\partial s(\mathbf{r}, t)}{\partial t} = -\nabla \cdot \mathbf{j}(\mathbf{r}, t). \quad (7.35)$$

According to Fick's law, a small concentration gradient in a homogeneous isotropic medium will evoke a flow

$$\mathbf{j}(\mathbf{r}, t) = -D\nabla s(\mathbf{r}, t) \quad (7.36)$$

with a *diffusion constant* D . By inserting Eq. (7.36) into the continuity equation (7.35), we obtain the diffusion equation

$$\frac{\partial s(\mathbf{r}, t)}{\partial t} = D\nabla^2 s(\mathbf{r}, t) \quad (7.37)$$

with the *Laplace operator* (or diffusion operator)

$$\nabla^2 s = \frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} + \frac{\partial^2 s}{\partial z^2}. \quad (7.38)$$

In one space dimension, the Laplace operator reads $\nabla^2 s(r, t) = \partial^2 s / \partial r^2$. The diffusion equation (7.37) for substance concentrations corresponds to the Fokker–Planck equation for the Brownian motion of individual particles (see Section 15.4).

To solve the diffusion equation in a region in space, we need to specify initial conditions (a concentration field $s(\mathbf{r}, 0)$ at time $t = 0$) and boundary conditions for all points \mathbf{r}_0 on the boundary. It is common to either fix concentrations $s(r_0, t)$ on the boundary (a *Dirichlet boundary condition*) or assume that the boundary is impermeable, that is, there is no flow component orthogonal to the boundary. As the flow is parallel to the concentration gradient, this is an example of a *von Neumann boundary condition*, which predefines the values of $\nabla s(\mathbf{r}_0) \cdot \mathbf{n}(\mathbf{r}_0)$, where \mathbf{n} is a unit vector orthogonal to the surface. In one space dimension, an impermeable boundary implies that $\partial s(r, t)/\partial r|_{r=r_0} = 0$, that is, the concentration gradient on the boundary vanishes. The diffusion

equation, together with proper initial and boundary conditions, determines a time-dependent concentration field $s(\mathbf{r}, t)$ at times $t \geq 0$.

7.3.3.2 Solutions of the Diffusion Equation

Diffusion tends to remove spatial heterogeneity: local concentration maxima (with negative curvature $\nabla^2 s$) will shrink and local minima will be filled. This can be seen from simple solutions of the diffusion equation in one space dimension.

- 1) *Stationary profile.* A stationary concentration profile in one space dimension has no curvature. In a region $0 \leq r \leq L$ with impermeable boundaries, that is, $\partial s / \partial r = 0$ at both $r = 0$ and $r = L$, the stationary solution $s^{st}(r) = \text{constant}$ is homogeneous, corresponding to a thermodynamic equilibrium. If concentrations at $r = 0$ and $r = L$ are fixed by boundary conditions, we obtain a linear stationary profile $s^{st}(r)$, in which substance flows down the concentration gradient in a steady state.
- 2) *Cosine profile.* We now consider the same region and choose a cosine pattern as initial condition. To keep concentrations positive, a baseline concentration is added. The cosine pattern is an eigenmode of the diffusion operator. Under diffusion, it keeps its shape, but its amplitude decreases exponentially (Figure 7.10b):

$$s(r, t) = s_0 e^{-\lambda(k)t} \cos(kr). \quad (7.39)$$

The time constant λ is given by the dispersion relation $\lambda(k) = -Dk^2$, so finer cosine patterns (with large wave numbers k) become smoothed out faster than broader patterns. Due to the boundary conditions, the possible wave numbers are restricted to values $k = n\pi/L$ with integer n .

- 3) *Gaussian profile.* Now we consider an infinite interval $-\infty < r < \infty$ and a substance amount a that is initially concentrated in the point $r = 0$. Diffusion leads to a Gaussian-shaped concentration profile (see Figure 7.10a)

$$s(r, t) = \frac{a}{\sqrt{2\pi(2Dt)}} e^{-r^2/(2Dt)}, \quad (7.40)$$

with a width $\sqrt{2Dt}$ proportional to the square root of diffusion constant D and time t . The variance increases proportionally in time, and the microscopic analogy to this diffusion process is Brownian motion as described by a Wiener process (see Section 15.4).

Since the diffusion equation is linear, general solutions can be obtained by linear combinations of the profiles (7.40) or (7.39).

7.3.3.3 Reaction–Diffusion Equation

Reaction–diffusion systems describe diffusing substances that are also engaged in chemical reactions. By combining the diffusion equation (7.37) with a kinetic model, we obtain a reaction–diffusion model

$$\frac{\partial s_i(\mathbf{r}, t)}{\partial t} = \sum_l n_{il} v_l(s(\mathbf{r}, t)) + D_i \nabla^2 s_i(\mathbf{r}, t) \quad (7.41)$$

for concentrations s_i . The first term represents local chemical reactions, while the second term describes diffusion with substance-specific diffusion constants D_i . The rate laws $v_l(s)$ are usually nonlinear, and most reaction–diffusion models can only be solved numerically, for example, by finite-element methods. Reaction–diffusion equations can show a wide range of dynamic behavior, including pattern formation, traveling and spiraling waves, or chaos. Similar behavior is also observed in reality: for instance, traveling waves in simple reaction–diffusion

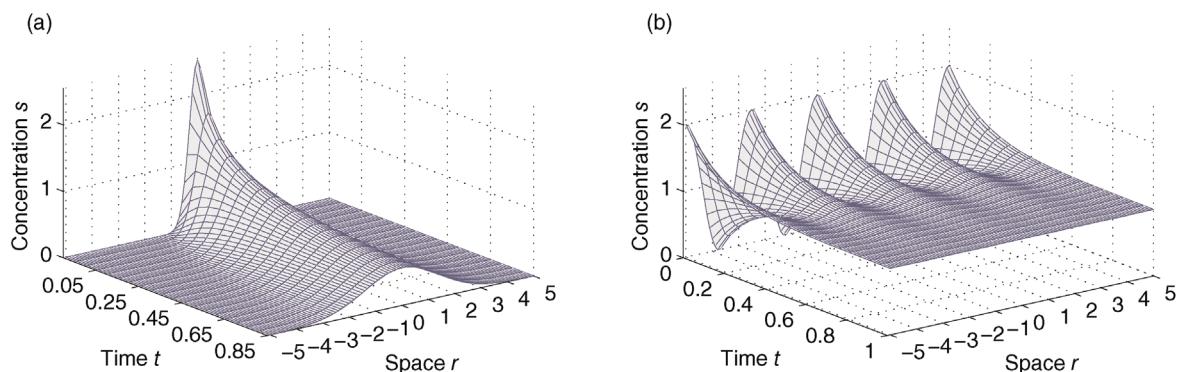


Figure 7.10 Diffusion blurs spatial concentration patterns. (a) A substance, initially localized at $r = 0$, forms a Gaussian-shaped cloud of increasing width. Parameters $a = 2$ and $D = 1$. Time, space, and concentration in arbitrary units. (b) An initial cosine pattern retains its shape, but its amplitude decreases exponentially in time. Parameters $s_0 = 1$, $D = 1$, $k = 2\pi/(5/2)$, and baseline concentration = 1.

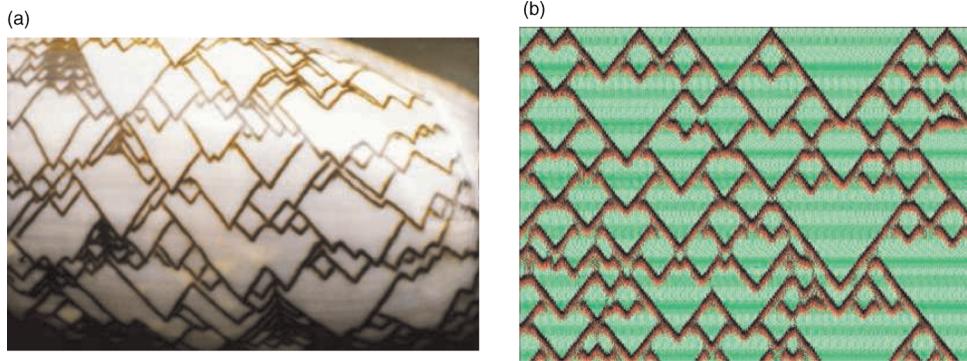


Figure 7.11 Patterns on seashells can be explained by a reaction–diffusion system. (a) Color patterns on the shell of *Oliva porphyria* are formed as material is added to the growing edge of the shell. The vertical direction in the picture can be seen as a time axis. The patterns are preformed by a chemical reaction–diffusion system: traveling waves lead to diagonal lines. (b) The pattern can be simulated by an activator–inhibitor system (black and red). Waves annihilate where they meet, and an additional global signaling substance (green) is needed to trigger the emergence of new waves [42,43]. The vertical axis represents time from top to bottom. (Courtesy of H. Meinhardt.)

systems yield a plausible explanation for the patterns on seashells [42,43] (Figure 7.11).

7.3.4 Robust Pattern Formation in Embryonic Development

The body plan of animals is established during embryonic development by the coordinated growth and differentiation of cells. The symmetry of our bodies and the similarity between twins show that very specific shapes can be generated robustly. Many developmental processes are organized by spatial patterns of *morphogens*, which establish a coordinate system in the developing tissue. If a morphogen shows a gradient along the embryo's anterior–posterior axis, cells can sense their positions on this axis and differentiate accordingly [44]. The spatio-temporal dynamics of morphogen fields arises from how cells sense and secrete morphogens, and it is influenced by the shape of the growing tissue and by other physiological processes. For instance, the process of ossification in embryonic development, responsible for bone formation, is triggered by stress fields caused by the embryo's movements [45]. In this way, bones grow adaptively where local stresses are highest and, thus, where existing shapes are strengthened most effectively.

Basic forms of collective dynamics based on cell communication are already found in bacteria (e.g., in quorum sensing) or in colonies of the social amoeba *Dictyostelium discoideum*, which can temporarily form a multicellular organism. A striking example of self-organization is provided by the hydra, a small multicellular animal that can not only regrow its main organs, foot and head, after it is cut into halves, but even reassociate from individual cells.

The spontaneous development of a new body axis in the hydra as a case of biochemical pattern formation has been simulated by mathematical models [46,47].

7.3.4.1 Bicoid Gradient in the Fly Embryo

Embryonic development has been studied extensively in the fly *Drosophila melanogaster*. The embryo's anterior–posterior body axis is established in early oocyte stage by the gradient of a morphogen called Bicoid. Bicoid is produced from mRNA attached to microtubules at the anterior end of the unfertilized egg; its gradient marks the anterior part of the embryo and serves as a coordinate for further patterning processes (see Figure 7.12a). In a model of the stationary Bicoid profile, Eldar *et al.* [48] assumed a steady-state balance of production, diffusion, and degradation. The concentration $s(r, t)$ along the anterior–posterior axis can be described by the reaction–diffusion equation

$$\frac{\partial s(r, t)}{\partial t} = D \nabla^2 s(r, t) - \kappa s(r, t) \quad (7.42)$$

with diffusion constant D and degradation constant κ . The stationary profile $s^{st}(r)$ satisfies the steady-state condition

$$0 = D \nabla^2 s^{st}(r) - \kappa s^{st}(r), \quad (7.43)$$

which can be solved by a sum of exponential profiles

$$s^{st}(r) = a_1 e^{-r/L_0} + a_2 e^{r/L_0} \quad (7.44)$$

with characteristic degradation length $L_0 = \sqrt{D/\kappa}$. This length is determined by biochemical constants and does not depend on the actual length of the embryo. The coefficients a_1 and a_2 need to be chosen to match the boundary conditions. To describe Bicoid production, we fix a

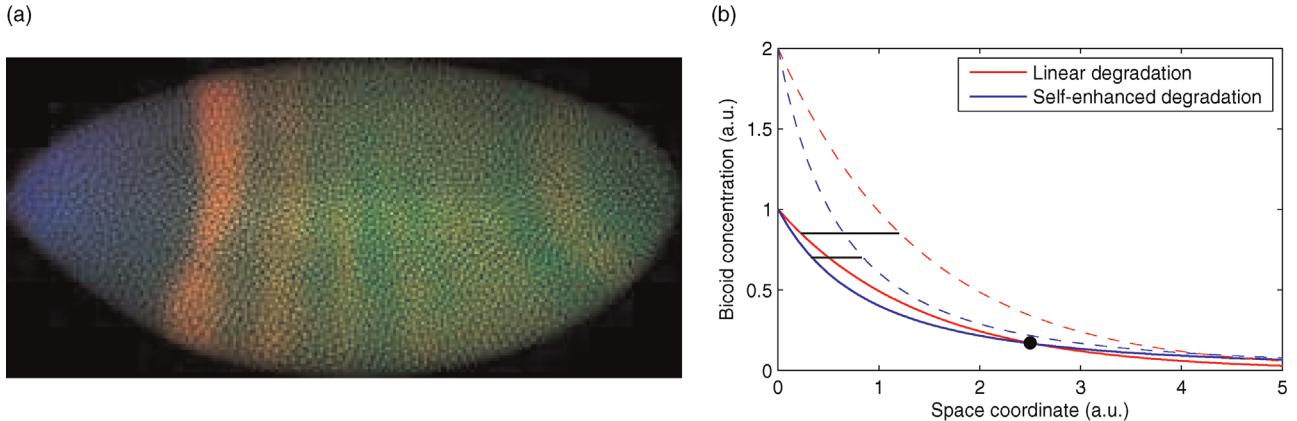


Figure 7.12 The morphogen Bicoid forms a gradient in the *Drosophila* embryo. (a) A microscope image [49] shows expression patterns of the genes bicoid (blue), even-skipped (red), and caudal (green). (From the FlyEx Database [50].) (b) Simulated gradient of the Bicoid protein. Solid curves show simulations from models with linear degradation (red) or self-enhanced degradation (blue) and with parameters $D = 0.01$, $\kappa = 0.02$, and $s_0 = 1$. The Bicoid concentrations of both models were made to coincide at $r = 0$ and $r \approx 2.5$ (dot). An increased concentration $s_0 = 2$ at the anterior end shifts both profiles (broken curves), but by different amounts (shifts marked by black lines).

constant concentration $s(r = 0) = s_0$ at the anterior boundary; s_0 is proportional to the protein production rate and thus to the mRNA amount. On the posterior end, Bicoid cannot leave the cell, so we set $ds^{\text{st}}/dr|_{r=L} = 0$. With these boundary conditions, the coefficients in Eq. (7.44) read

$$\alpha_1 = \frac{\beta^2 s_0}{1 + \beta^2}, \quad \alpha_2 = \frac{s_0}{1 + \beta^2} \quad (7.45)$$

with $\beta = \exp(L/L_0)$. If the characteristic length L_0 is much shorter than the length L of the embryo, we can approximately set $s^{\text{st}}(r) = 0$ at the posterior end, the second term in Eq. (7.44) vanishes, and we obtain the exponential profile

$$s^{\text{st}}(r) = s_0 e^{-r/L_0}. \quad (7.46)$$

This profile can serve as a coordinate system for other developmental processes. However, if the mRNA amount, and thus morphogen production, varies from embryo to embryo, this will shift the emerging pattern (see Figure 7.12b), which makes it an unreliable clue for other developmental processes. The size of these shifts is inversely proportional to the morphogen slope near the source. Eldar *et al.* showed how more reliable gradients can be obtained by a slight change in the model [48]. In the new model, Bicoid is assumed to catalyze its own degradation with a degradation rate $\kappa s^2(r, t)$ instead of $\kappa s(r, t)$. With boundary conditions as above, the steady-state profile reads

$$s^{\text{st}}(r) = \frac{6D/\kappa}{(r + \sqrt{6D/(\kappa s_0)})^2}. \quad (7.47)$$

This pattern combines two favorable properties for reliable gradients: a steep decrease near the source (which makes the gradient robust against variations in Bicoid production) and a relatively large Bicoid level along the embryo. With an exponential profile (7.46), there would be a trade-off between the two properties: a steeper decrease would lead to smaller Bicoid levels along the embryo, which are more easily distorted by noise. The profile arising from Eq. (7.47) combines both advantages. In Figure 7.12b, the two profiles are compared at equal Bicoid levels at the center of the embryo: model (7.47) shows a much smaller shift after overexpression of Bicoid. We will see more examples of robustness in Section 10.2.

7.3.5 Spontaneous Pattern Formation

The color patterns on zebra and leopard furs are believed to arise from a process called self-organized pattern formation. Simple reaction–diffusion models, which simulate this process, can reproduce the typical geometries of these patterns and their dependence on body shapes [51]. While a single morphogen can form gradients, two interacting morphogens can produce more complicated patterns whose properties depend on the rate laws and diffusion constants. A number of pattern formation processes (including hair follicle spacing, hydra regeneration, and lung branching), for instance, have been attributed to the activator–inhibitor pair Wnt and Dkk [52]. Figure 7.13 shows an example of pattern formation, the emergence of spots

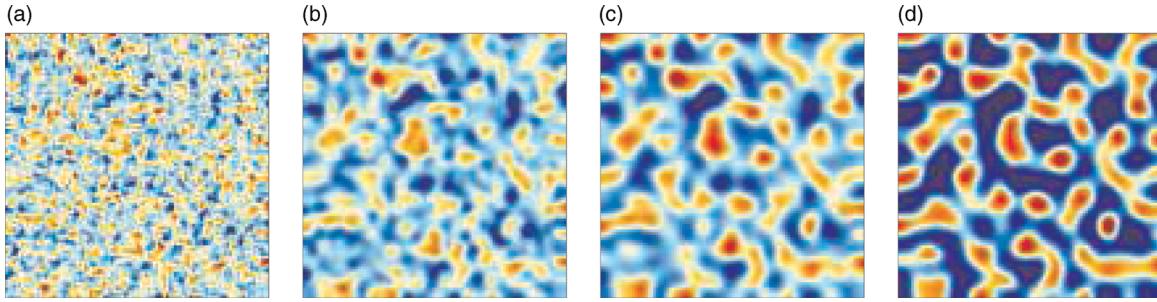


Figure 7.13 Spontaneous pattern formation in the Gierer–Meinhardt model. The pictures show simulation snapshots (activator concentration) at time points $t = 50$ (a), $t = 200$ (b), $t = 50$ (c), and $t = 200$ (d). A stripe pattern emerges spontaneously from a noisy initial concentration profile (uniform random values from the interval $[0.1, 1]$). Parameters $D_a = 0.002$, $D_b = 0.2$, $\rho = 1$, $\mu_a = 0.01$, $\mu_b = 0.015$, $\kappa = 0.1$, and discretization $\Delta x = 0.2$, $\Delta t = 1$ (arbitrary units).

in the Gierer–Meinhardt model

$$\begin{aligned}\frac{\partial a}{\partial t} &= \frac{\rho a^2}{b(1 + \kappa a^2)} - \mu_a a + D_a \nabla^2 a, \\ \frac{\partial b}{\partial t} &= \rho a^2 - \mu_b b + D_b \nabla^2 b\end{aligned}\quad (7.48)$$

with parameters ρ and κ for production, μ_a and μ_b for degradation, and D_a and D_b for diffusion. The concentrations a and b correspond, respectively, to an activator (which increases the production of both substances) and an inhibitor (which inhibits the activator). Depending on parameters, the model can lead to spots, stripes, or gradients. The pattern in Figure 7.13 has a typical length scale (distance between spots), which is predetermined by reaction and diffusion parameters and does not depend on tissue size. The exact shape depends on random fluctuations in the initial homogeneous state.

In reaction–diffusion systems, patterns can spontaneously emerge from almost homogeneous morphogen distributions – a paramount example of spontaneous symmetry breaking. Similar patterns arise in clouds or sand ripples. Although the physical systems are completely different, their ability to form patterns relies on a common mechanism called *Turing instability* [53]. To allow for spontaneous pattern formation, a system must have some basic properties: first, it must possess a homogeneous steady state that is stable against homogeneous concentration changes, but unstable against spatial variation. Second, the dynamics must amplify fluctuations of some finite wavelength more strongly than fluctuations of larger or smaller wavelengths. These conditions can be satisfied by simple systems with two morphogens, performing short-range activation and long-range inhibition [46] (see Figure 7.14a). If the inhibitor diffuses faster than the activator, the diffusion will not remove patterns, as usually, but on the contrary create patterns. If the homogeneous state is unstable against local fluctuations,

small fluctuations in this state will be amplified, leading to a stable pattern with spots or stripes of high or low activator levels. For propagating waves to occur (as shown in Figure 7.11), the inhibitor must have a longer lifetime and diffuse more slowly than the activator.

7.3.6

Linear Stability Analysis of the Activator–Inhibitor Model

Nonlinear reaction–diffusion systems can show complicated dynamic behavior. However, general conclusions about pattern formation can be derived from a linear stability analysis, which concerns the very beginning of

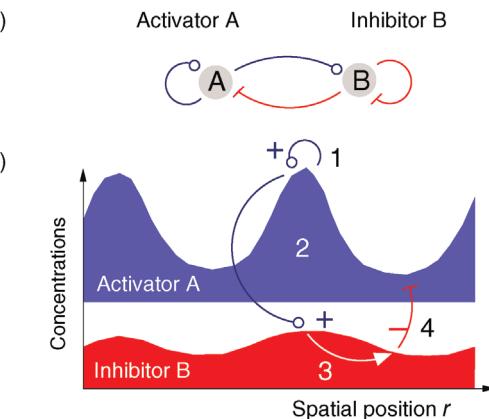


Figure 7.14 Pattern formation in an activator–inhibitor system. (a) Activator A increases the production of both substances, while inhibitor B decreases the production. In a nonspatial model, the system shows a stable steady state. (b) In a reaction–diffusion system, local inhomogeneities are amplified: an elevation of A catalyzes its own further increase (1); it increases the level of B (2), which diffuses faster than A (3) and represses the level of A in a distance (4). By the same mechanism, the resulting valley then strengthens the original elevation and other concentration peaks nearby.

pattern formation, the emergence of small inhomogeneities from a spatially homogeneous state. Focusing again on activator–inhibitor systems, we consider the reaction–diffusion system

$$\begin{aligned}\frac{da}{dt} &= f(a, b) + D_A \nabla^2 a, \\ \frac{db}{dt} &= g(a, b) + D_B \nabla^2 b\end{aligned}\quad (7.49)$$

describing an activator A and an inhibitor B. If the production functions f and g are nonlinear, the system can show complicated dynamic behavior. However, a main condition for pattern formation, the amplification of small initial fluctuations of specific sizes, can be studied by linear stability analysis. In homogeneous, nonpatterned states, the diffusion terms vanish and space dependence can be neglected. For small deviations $u(t) = a(t) - a^{st}$ and $v(t) = b(t) - b^{st}$, we can linearize the system and obtain

$$\frac{d}{dt} \begin{pmatrix} u \\ v \end{pmatrix} \approx \begin{pmatrix} f_A & f_B \\ g_A & g_B \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{A} \begin{pmatrix} u \\ v \end{pmatrix}. \quad (7.50)$$

As a stability criterion (considering a small homogeneous variation of a and b), the eigenvalues $\lambda_{1,2}$ of \mathbf{A} must have negative real parts, implying that

$$\begin{aligned}\text{Tr}(\mathbf{A}) &= f_A + g_B < 0, \\ \text{Det}(\mathbf{A}) &= f_A g_B - f_B g_A > 0.\end{aligned}\quad (7.51)$$

Next, we consider small space-dependent deviations $u(r, t) = a(r, t) - a^{st}$ and $v(r, t) = b(r, t) - b^{st}$. In a linear approximation, we obtain the equation

$$\frac{d}{dt} \begin{pmatrix} u \\ v \end{pmatrix} = \mathbf{A} \begin{pmatrix} u \\ v \end{pmatrix} + \mathbf{D} \begin{pmatrix} \nabla^2 u \\ \nabla^2 v \end{pmatrix} \quad (7.52)$$

with the diffusion matrix $\mathbf{D} = \begin{pmatrix} D_A & 0 \\ 0 & D_B \end{pmatrix}$. To solve Eq. (7.52) in one space dimension, we use the ansatz

$$\begin{pmatrix} u(r, t) \\ v(r, t) \end{pmatrix} = \mathbf{w}(r, t) = \mathbf{w}_0 e^{\lambda t} \cos(k, r), \quad (7.53)$$

where λ is real-valued. The function describes a cosine pattern whose amplitude grows or decreases exponentially in time; the vector \mathbf{w}_0 contains the initial amplitudes of u and v . If $\lambda > 0$, the amplitude of a pattern will be amplified. If we consider a spatial region $0 \leq r \leq L$ with impermeable boundaries, the boundary conditions ($\partial a / \partial r = \partial b / \partial r = 0$ at $r = 0$ and $r = L$) can only be satisfied by wave numbers k of the form $k = n\pi/L$ with integer n . By taking the derivatives

$$\begin{aligned}\frac{d\mathbf{w}(r, t)}{dt} &= \lambda \mathbf{w}_0 e^{\lambda t} \cos(k, r), \\ \nabla^2 \mathbf{w}(r, t) &= -k^2 \mathbf{w}_0 e^{\lambda t} \cos(k, r)\end{aligned}\quad (7.54)$$

and inserting them into Eq. (7.52), we obtain the condition

$$(\mathbf{A} - k^2 \mathbf{D} - \lambda I) \mathbf{w}_0 = 0. \quad (7.55)$$

It shows that λ and k cannot be chosen independently; instead, the growth rate λ in our ansatz (7.53) must be a function of the wave number k . Specifically, λ must be an eigenvalue of the matrix $\mathbf{A} - k^2 \mathbf{D}$, which leads to a dispersion relation $\lambda = \lambda(k^2)$.

We now try to find an instability, that is, a range of positive wave numbers k for which initial patterns will grow. As a condition for such wave numbers k , $\mathbf{A} - k^2 \mathbf{D}$ must have a positive eigenvalue, that is, one of the conditions

$$\text{Tr}(\mathbf{A} - k^2 \mathbf{D}) = \text{Tr}(\mathbf{A}) - k^2(D_A + D_B) < 0, \quad (7.56)$$

$$\begin{aligned}\text{Det}(\mathbf{A} - k^2 \mathbf{D}) &= \text{Det}(\mathbf{A}) + k^4 D_A D_B \\ &\quad - k^2(f_A D_B + g_B D_A) > 0\end{aligned}\quad (7.57)$$

must be violated. Since Eq. (7.51) still holds, condition (7.56) cannot be violated; instead, condition (7.57) must be violated, so $\text{Det}(\mathbf{A} - k^2 \mathbf{D}) \leq 0$. The critical case $\text{Det}(\mathbf{A} - k^2 \mathbf{D}) = 0$ leads to a biquadratic equation for k , with the solutions

$$\begin{aligned}k_{1,2}^2 &= \frac{f_A D_B + g_B D_A}{2 D_A D_B} \\ &\pm \sqrt{\left(\frac{f_A D_B + g_B D_A}{2 D_A D_B} \right)^2 - \frac{\text{Det}(\mathbf{A})}{D_A D_B}}.\end{aligned}\quad (7.58)$$

To obtain a window of positive wave numbers with positive λ , both solutions $k_{1,2}^2$ of Eq. (7.58) must be positive, so $k_{1,2}$ must be real-valued. This requires that

$$\begin{aligned}f_A D_B + g_B D_A &> 0, \\ (f_A D_B + g_B D_A)^2 &> 4 D_A D_B \text{Det}(\mathbf{A}).\end{aligned}\quad (7.59)$$

A comparison to condition (7.51) shows that f_A and g_B must have opposite signs. By definition, the activator A has a self-activating effect $f_A > 0$, so the inequality (7.51) requires that $D_B > D_A$: the inhibitor must diffuse faster than the activator. The growing pattern is dominated by the sine wave pattern that grows most rapidly: its squared wave number k^2 is given by the mean value of the solutions of Eq. (7.57) and reads

$$k^2 = \frac{f_A D_B + g_B D_A}{2 D_A D_B}. \quad (7.60)$$

If the nonlinearities in Eq. (7.49) have weak saturating effects, the initial pattern will approximately pre-determine the final stationary pattern.

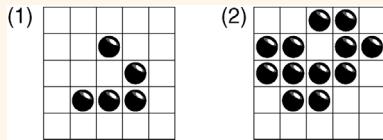
Exercises

Section 7.2

- 1) *Fluctuations of a single substance.* Consider a biochemical system in a macroscopic steady state with average molecule numbers \bar{x}_i , reaction velocities v_b , stoichiometric coefficients n_{jb} and elasticities \tilde{e}_i^l . Compute the variance of the fluctuating molecule number x_i caused by chemical noise. Assume that all other substances have fixed molecule numbers (no fluctuations). *Hint:* Linearize the chemical Langevin equation around the steady-state value and use the Lyapunov equation to compute the variance.

Section 7.3

- 2) *Game of life.* (a) Invent two initial configurations that stay unchanged under the updating rules of the game of life. (b) Simulate the patterns shown below (called “glider” and “lightweight spaceship”) with paper and pencil. The surrounding cells are supposed to be empty (“dead”). (c) Implement the game of life as a computer program and play with random initial configurations.



- 3) *Response matrix for a kinetic model with dilution.* Assume a biochemical reaction system in a cell with constant growth rate κ . Assume that the system has a stable steady state, and derive the formula $\tilde{R}^S = -(\mathbf{N}\tilde{e} - \kappa\mathbf{I})^{-1}\mathbf{N}\tilde{\pi}$ for unscaled concentration response coefficients.
- 4) *Diffusion with cosine profiles.* Show that the diffusion equation is solved by spatial cosine profiles with exponentially decreasing amplitude $c(x, t) = c_0 e^{-\lambda(k)t} \cos(kx)$, and compute the dispersion relation $\lambda(k)$.
- 5) *Bicoid profile with self-enhanced degradation.* Show that the stationary profile

$$\begin{aligned}s^{\text{st}}(x) &= \frac{6D/\kappa}{(x + \sqrt{6D/(\kappa s_0)})^2} \\ &= \frac{6D/\kappa}{(x + (2Da/j(0))^{1/3})^2}\end{aligned}$$

is a solution of the Bicoid reaction–diffusion system with an autocatalytic degradation term $-\kappa s(x, t)^2$. *Hint:* Use the ansatz $s^{\text{st}}(x) = a/(x + b)^2$.

- 6) *Pattern formation.* The pair-rule gene *eve* is expressed in seven stripes in the blastoderm of the fruit fly *D. melanogaster*. The stripes do not arise from spontaneous pattern formation, but from a response to existing patterns of the regulatory proteins *Krüppel*, *Bicoid*, *Giant*, and *Hunchback*. The specific response is hard-coded in the regulation function of the *eve* gene. Speculate in broad terms about advantages and disadvantages of spontaneous and “hard-wired” pattern formation.
- 7) *Diffusion with Gaussian profile.* Show that the diffusion equation (7.37) is solved by a Gaussian-shaped profile of increasing width

$$c(x, t) = \frac{c_0}{\sqrt{2\pi(2Dt)}} e^{-x^2/(2Dt)}.$$

- 8) *Discrete Laplace operator.* The reaction–diffusion equation

$$\frac{dc(x, t)}{dt} = f(c(x, t)) + D\nabla^2 c(x, t)$$

can be approximated by a diffusion process on a one-dimensional grid. Consider the concentrations $c_i(t) = c(x_i, t)$ at discrete grid points $x_i = i\Delta x$ in space and use a second-order Taylor expansion of $c(x, t)$ to derive a system of ordinary differential equations:

$$\frac{dc_i(t)}{dt} = f(c_i(t)) + a(c_{i-1} - 2c_i + c_{i+1}).$$

How is a related to D and Δx ?

- 9) *Activator–inhibitor system.* Implement and simulate the reaction–diffusion system (Gierer–Meinhardt model):

$$\begin{aligned}\frac{\partial a}{\partial t} &= \frac{\rho a^2}{b(1 + ka^2)} - \mu_a a + D_a \nabla^2 a, \\ \frac{\partial b}{\partial t} &= \rho a^2 - \mu_b b + D_b \nabla^2 b,\end{aligned}$$

with parameters $D_a = 0.002$, $D_b = 0.2$, $\rho = 1$, $\mu_a = 0.01$, $\mu_b = 0.015$, $\kappa = 0.1$. Choose a discretization $\Delta x = 0.2$, $\Delta t = 1$ (in arbitrary units).

References

- 1 Kauffman, S.A. and Weinberger, E.D. (1989) The NK model of rugged fitness landscapes and its application to maturation of the immune response. *J. Theor. Biol.*, 141 (2), 211–245.
- 2 Kauffman, S.A. and Johnsen, S. (1991) Coevolution to the edge of chaos: coupled fitness landscapes, poised states, and coevolutionary avalanches. *J. Theor. Biol.*, 149 (4), 467–505.
- 3 Kauffman, S.A. (1993) *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York.
- 4 Kauffman, S.A. and Macready, W.G. (1995) Search strategies for applied molecular evolution. *J. Theor. Biol.*, 173 (4), 427–440.
- 5 Waltermann, C., Floettmann, M., and Klipp, E. (2010) G1 and G2 arrests in response to osmotic shock are robust properties of the budding yeast cell cycle. *Genome Inform.*, 24 (1), 204–217.
- 6 Harvey, I. and Bossomaier, T. (1997) Time out of joint: attractors in asynchronous random Boolean networks, in *Proceedings of the Fourth European Conference on Artificial Life (ECAL97)* (eds P. Husbands and I. Harvey), MIT Press, pp. 67–75.
- 7 Kauffman, S.A. (1991) Antichaos and adaptation. *Sci. Am.*, 265 (2), 78–84.
- 8 Shmulevich, I., Dougherty, E.R., Kim, S., and Zhang, W. (2002) Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18 (2), 261–274.
- 9 Flottmann, M., Schärf, T., and Klipp, E. (2012) A stochastic model of epigenetic dynamics in somatic cell reprogramming. *Front. Physiol.*, 3, 216.
- 10 Bernardinello, L. and de Cindio, F. (1992) *A Survey of Basic Net Models and Modular Net Classes*, Springer, Berlin.
- 11 Küffner, R., Zimmer, R., and Lengauer, T. (2000) Pathway analysis in metabolic databases via differential metabolic display (DMD). *Bioinformatics*, 16 (9), 825–836.
- 12 Reddy, V.N., Lieberman, M.N., and Mavrovouniotis, M.L. (1996) Qualitative analysis of biochemical reaction systems. *Comput. Biol. Med.*, 26 (1), 9–24.
- 13 Matsuno, H., Tanaka, Y., Aoshima, H., Doi, A., Matsui, M., and Miyano, S. (2003) Biopathways representation and simulation on hybrid functional Petri net. *In Silico Biol.*, 3 (3), 389–404.
- 14 Turner, T.E., Schnell, S., and Burridge, K. (2004) Stochastic approaches for modelling *in vivo* reactions. *Comput. Biol. Chem.*, 28 (3), 165–178.
- 15 Kurtz, T.G. (1971) Limit theorems for sequences of jump Markov processes approximating ordinary differential processes. *J. Appl. Probab.*, 8, 344–356.
- 16 Gillespie, D.T. (2000) The chemical Langevin equation. *J. Chem. Phys.*, 113 (1), 297–306.
- 17 Rao, C.V. and Arkin, A.P. (2003) Stochastic chemical kinetics and the quasi-steady-state assumption: application to the Gillespie algorithm. *J. Chem. Phys.*, 118 (11), 4999.
- 18 Jahnke, T. and Huiszinga, W. (2007) Solving the chemical master equation for monomolecular reaction systems analytically. *J. Math. Biol.*, 54, 1–26.
- 19 Gillespie, D.T. (1976) A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comput. Phys.*, 22, 403–434.
- 20 Gillespie, D.T. (1977) Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81, 2340–2361.
- 21 Alfonsi, A., Cancès, E., Turinici, G., Di Ventura, B., and Huiszinga, W. (2005) Adaptive simulation of hybrid stochastic and deterministic models for biochemical systems. *ESAIM Proc.*, 14, 1–13.
- 22 Gillespie, D.T. (2001) Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 115 (4), 1716–1733.
- 23 Cao, Y., Gillespie, D.T., and Petzold, L.R. (2006) Efficient step size selection for the tau-leaping simulation method. *J. Chem. Phys.*, 124, 044109.
- 24 van Kampen, N.G. (1997) *Stochastic Processes in Physics and Chemistry*, 2nd edn, North-Holland, Amsterdam.
- 25 Thattai, M. and van Oudenaarden, A. (2001) Intrinsic noise in gene regulatory networks. *Proc. Natl. Acad. Sci. USA*, 98 (15), 8614–8619.
- 26 Elf, J. and Ehrenberg, M. (2003) Fast evaluation of fluctuations in biochemical networks with the linear noise approximation. *Genome Res.*, 13, 2475–2484.
- 27 Hayot, F. and Jayaprakash, C. (2004) The linear noise approximation for molecular fluctuations within cells. *Phys. Biol.*, 1, 205–210.
- 28 Samoilov, M.S. and Arkin, A.P. (2006) Deviant effects in molecular reaction pathways. *Nat. Biotechnol.*, 24 (10), 1235–1240.
- 29 Ingalls, B.P. (2004) A frequency domain approach to sensitivity analysis of biochemical systems. *J. Phys. Chem. B*, 108, 1143–1152.
- 30 Liebermeister, W. (2005) Response to temporal parameter fluctuations in biochemical networks. *J. Theor. Biol.*, 234 (3), 423–438.
- 31 Wilhelm, T. and Heinrich, R. (1995) The smallest chemical reaction systems with Hopf bifurcation. *J. Math. Chem.*, 17, 1–14.
- 32 Jilkine, A. and Edelstein-Keshet, L. (2011) A comparison of mathematical models for polarization of single eukaryotic cells in response to guided cues. *PLoS Comput. Biol.*, 7 (4), e1001121.
- 33 Diener, C., Schreiber, G., Giese, W., del Rio, G., Schröder, A., and Klipp, E. (2014) Yeast mating and image-based quantification of spatial pattern formation. *PLoS Comput. Biol.*, 10 (6), e1003690.
- 34 Murray, J.D. (2003) *Mathematical Biology. II. Spatial Models and Biomedical Applications*, Springer.
- 35 Heinrich, R. and Rapoport, T.A. (2005) Generation of nonidentical compartments in vesicular transport systems. *J. Cell Biol.*, 168, 271–280.
- 36 Binder, B., Goede, A., Berndt, N., and Holz Äijter, H.-G. (2009) A conceptual mathematical model of the dynamic self-organisation of distinct cellular organelles. *PLoS One*, 4 (12), e8295.
- 37 Poulin, P. and Theil, F. (2002) Prediction of pharmacokinetics prior to *in vivo* studies. II. Generic physiologically based pharmacokinetic models of drug disposition. *J. Pharm. Sci.*, 91 (5), 1358–1370.
- 38 Theil, F., Guentert, T.W., Haddad, S., and Poulin, P. (2003) Utility of physiologically based pharmacokinetic models to drug development and rational drug discovery candidate selection. *Toxicol. Lett.*, 138, 29–49.
- 39 Deutsch, A. and Dormann, S. (2004) *Cellular Automaton Modeling and Biological Pattern Formation*, Birkhäuser.
- 40 Gardner, M. (1970) Mathematical games: the fantastic combinations of John Conway's new solitaire game "life". *Sci. Am.*, 223, 120–123.
- 41 Jol, S.J., Kümmel, A., Hatzimanikatis, V., Beard, D.A., and Heinemann, M. (2010) Thermodynamic calculations for biochemical transport and reaction processes in metabolic networks. *Biophys. J.*, 99, 3139–3144.
- 42 Meinhardt, H. and Klingler, M. (1987) A model for pattern formation on the shells of molluscs. *J. Theor. Biol.*, 126, 63–69.
- 43 Meinhardt, H. (2003) *The Algorithmic Beauty of Sea Shells*, 3rd edn, Springer, Heidelberg.
- 44 Gurdon, J.B. and Bourillot, P.-Y. (2001) Morphogen gradient interpretation. *Nature*, 413, 797–803.
- 45 Nowlan, N.C., Murphy, P., and Prendergast, P.J. (2007) Mechanobiology of embryonic limb development. *Ann. N.Y. Acad. Sci.*, 1101, 389–411.
- 46 Meinhardt, H. and Gierer, A. (1974) Applications of a theory of biological pattern formation based on lateral inhibition. *J. Cell Sci.*, 15, 321–346.

- 47** Meinhardt, H. (1993) A model for pattern formation of hypostome, tentacles, and foot in hydra: how to form structures close to each other, how to form them at a distance. *Dev. Biol.*, 157, 321–333.
- 48** Eldar, A., Rosin, D., Shilo, B.Z., and Barkai, N. (2003) Self-enhanced ligand degradation underlies robustness of morphogen gradients. *Dev. Cell*, 5, 635–646.
- 49** Kosman, D., Small, S., and Reinitz, J. (1998) Rapid preparation of a panel of polyclonal antibodies to *Drosophila* segmentation proteins. *Dev. Genes Evol.*, 208, 290–294.
- 50** Poustelnikova, E., Pisarev, A., Blagov, M., Samsonova, M., and Reinitz, J. (2004) A database for management of gene expression data *in situ*. *Bioinformatics*, 20, 2212–2221.
- 51** Gierer, A. and Meinhardt, H. (1972) A theory of biological pattern formation. *Kybernetik*, 12, 30–39.
- 52** Kondo, S. and Miura, T. (2010) Reaction–diffusion model as a framework for understanding biological pattern formation. *Science*, 329, 1616–1620.
- 53** Turing, A.M. (1952) The chemical basis of morphogenesis. *Philos. Trans. R. Soc. Lond.*, 237 (641), 37–72.
- van Kampen, N.G. (1997) *Stochastic Processes in Physics and Chemistry*, 2nd edn, North-Holland, Amsterdam.
- Stochastic simulation:** Gillespie, D.T. (1977) Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81, 2340–2361.
- Tau-leaping method:** Gillespie, D.T. (2001) Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, 115 (4), 1716–1733.
- Chemical Langevin equation:** Gillespie, D.T. (2000) The chemical Langevin equation. *J. Chem. Phys.*, 113 (1), 297–306.
- Noise in gene expression:** Thattai, M. and van Oudenaarden, A. (2001) Intrinsic noise in gene regulatory networks. *Proc. Natl. Acad. Sci. USA*, 98 (15), 8614–8619.
- Linear noise approximation:** Elf, J. and Ehrenberg, M. (2003) Fast evaluation of fluctuations in biochemical networks with the linear noise approximation. *Genome Res.*, 13, 2475–2484.
- Biochemical networks under stochastic perturbations:** Liebermeister, W. (2005) Response to temporal parameter fluctuations in biochemical networks. *J. Theor. Biol.*, 234 (3), 423–438.
- Pattern formation:** Turing, A.M. (1952) The chemical basis of morphogenesis. *Philos. Trans. R. Soc. Lond.*, 237 (641), 37–72.
- Biological patterns:** Kondo, S. and Miura, T. (2010) Reaction–diffusion model as a framework for understanding biological pattern formation. *Science*, 329, 1616–1620.
- Biological patterns:** Meinhardt, H. (2003) *The Algorithmic Beauty of Sea Shells*, 3rd edn, Springer, Heidelberg.
- Biological patterns:** Murray, J.D. (2003) *Mathematical Biology. II. Spatial Models and Biomedical Applications*, Springer.

Further Reading

- Stochastic models:** Wilkinson, D.J. (2011) *Stochastic Modelling for Systems Biology*, 2nd edn, Chapman & Hall/CRC Mathematical and Computational Biology, CRC Press.
- Stochastic processes:** Honerkamp, J. (1994) *Stochastic Dynamical Systems*, John Wiley & Sons, Inc., New York.

Network Structure, Dynamics, and Function

8

Cells use thousands of proteins to produce and convert substances, to sense environmental stimuli and internal cell states, and to transmit and process this information. Even bacteria contain several thousands of genes, so the number of enzyme-catalyzed reactions (including transcription and translation for all genes) and the number of compounds involved are on the order of 10^5 . A way to deal with such highly complex systems is to disregard all their quantitative details and to simply depict them as networks.

Whenever complex information is reduced to discrete elements and their relations, it can be displayed as a network. Biological networks may describe very different things: causal or mechanistic effects (reaction–metabolite network, transcriptional regulation network), molecule interactions (protein–protein or protein–DNA binding), statistical relationships (Bayesian networks for gene expression, correlated metabolite fluctuations), or functional or evolutionary relatedness.

Many networks are inferred statistically, for example, from high-throughput data or text mining screens, and represent things such as statistical correlations or even co-occurrence in scientific articles. Based on quantitative relationships, which can be formally seen as distances or similarities, networks can be constructed by thresholding. An edge is drawn whenever a correlation exceeds some threshold value or is statistically significant. Such inferred networks may be related to biological networks, but need not represent them directly. In particular, they often capture indirect interactions (e.g., genes influencing each others' expression via changes of the metabolic state) instead of specified mechanisms (e.g., gene products acting directly as transcription factors).

A focus on network structures and omission of quantitative details can be fruitful. Sometimes, the dynamic properties of a system depend mostly on its network

8.1 Structure of Biochemical Networks

- Random Graphs
- Scale-Free Networks
- Connectivity and Node Distances
- Network Motifs and Significance Tests
- Explanations for Network Structures

8.2 Regulation Networks and Network Motifs

- Structure of Transcription Networks
- Regulation Edges and Their Steady-State Response
- Negative Feedback
- Adaptation Motif
- Feed-Forward Loops

8.3 Modularity and Gene Functions

- Cell Functions Are Reflected in Structure, Dynamics, Regulation, and Genetics
- Metabolics Pathways and Elementary Modes
- Epistasis Can Indicate Functional Modules
- Evolution of Function and Modules
- Independent Systems as a Tacit Model Assumption
- Modularity and Biological Function Are Conceptual Abstractions

Exercises

References

Further Reading

structure and less on the dynamics of individual elements. Negative feedback loops, for instance, can have typical effects independent of their physical realization. Thus, even models with an inaccurate dynamics but correct network structure may be helpful to simulate system behavior. However, a main usage of networks is visualization: Networks can highlight structures that would otherwise go unnoticed. When patterns stand out, we may analyze them closely and try to find reasons for them in the system's dynamics or evolution.

8.1

Structure of Biochemical Networks

Summary

Networks provide an easy and common way to represent complex biochemical systems. Nodes typically correspond to molecule species or genes, while edges represent molecular interactions, causal influences, or abstract relationships such as correlations in measured data. Biological networks show characteristic structures, including scale-free degree distributions, small average path lengths, and network motifs. To prove the statistical significance of such structures, networks are compared with random graphs with defined statistical properties.

The prominent biochemical networks in cells – metabolic networks, transcription networks, and signal transduction networks – perform different functions and evolve in different ways. Metabolic networks allow cells to produce and convert metabolites; they support the metabolic fluxes, which vary depending on supply and demand. The network structure is, in principle, determined by the enzymes encoded in an organism's genome, and many metabolic networks have been reconstructed [1]. Depending on the organism's ecological niche, these networks range from small sizes, in intracellular parasites, to very large sizes, for instance, in plants. However, central pathways such as glycolysis are widely conserved and all organisms use a core set of important cofactors. A commonly used reconstruction of the *Escherichia coli* metabolic network [2] contains 1260 genes, 1148 unique functional proteins, and 1039 unique metabolites. The 2077 reactions are assigned to different

compartments (cytoplasm, periplasm, or the extracellular space) and include 1387 chemical and 690 transport reactions. A current reconstruction of the human metabolic network contains 7440 reactions and 2626 metabolites (which, differentiated by cell compartments, yield 5063 molecule species), and has been used to construct 65 cell-type-specific submodels [3].

Transcription networks, as shown in Figure 8.1a, describe the regulation of gene expression by transcription factors; their structure is biochemically determined by transcription factor binding sites. Signal transduction networks rely mostly on interactions between proteins, for example, kinases or phosphatases, which can mutually phosphorylate and dephosphorylate each other. Physical interactions between proteins in cells, such as binding and complex formation, can be derived from yeast two-hybrid screens (see Chapter 14.10) and visualized in protein–protein interaction networks (see Figure 8.1b).

Biological networks can contain repetitive or prominent local structures, which may work as functional units and can be detected with statistical methods. To understand how such structures emerge, we need to see how networks can change during evolution. Bacterial promoter sequences, for instance, can evolve relatively rapidly, so edges in their transcription networks (implemented by transcription factor binding sites) may easily get lost by mutations. If we know how mutations, without selection for network function, would rewire a network, we can compare the structures in actual networks with such random structures. The comparison can point us to network structures that may have been conserved for specific biological functions. Such structures can be candidates for further biological study.

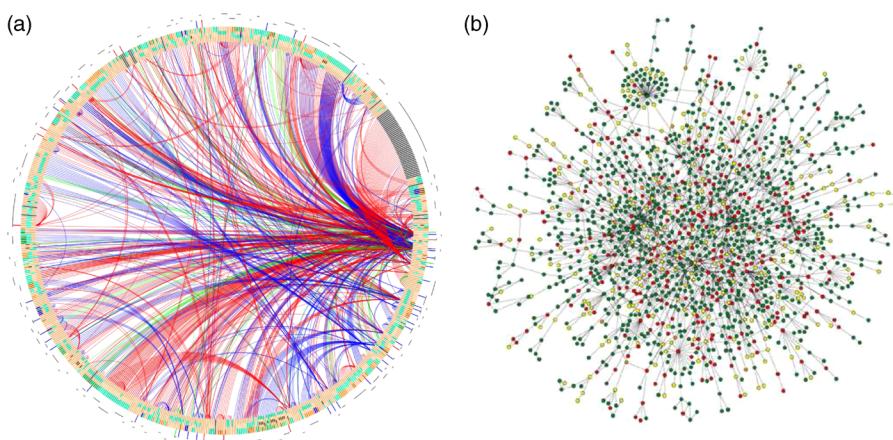


Figure 8.1 Biological networks. (a) Transcription network in *E. coli*. Genes are shown by colored segments. Arcs show different kinds of regulation (blue: activation, red: inhibition, green: dual regulation). Traces around the circle indicate autoregulation. (Courtesy of S. Ortiz, L. Rico, and A. Valencia.) (b) Protein–protein binding interactions. The network shown contains about 80% of all protein species in yeast. Node colors indicate the phenotypic effects of removing a protein (red: lethal; green: nonlethal; orange: slow growth; yellow: unknown). (From Ref. [4].)

8.1.1

Random Graphs

8.1.1.1 Mathematical Graphs

Networks in which edges point from nodes to nodes (and not to other edges) can be described by mathematical *graphs*. A graph consists of a discrete set of *nodes* and a set of *edges*, defined as pairs of nodes. In directed graphs, edges are ordered pairs represented by arrows. In undirected graphs, edges are unordered pairs and displayed by lines. Examples are shown in Figure 8.2.

The structure of a graph can be represented by an adjacency matrix $\mathbf{A} = (a_{ij})$: Edges from node i to node j are represented by matrix elements $a_{ij} = 1$, all other elements have values of 0. For undirected graphs, the adjacency matrix is symmetric. A directed graph with n nodes can have maximally n^2 edges (corresponding to elements of the adjacency matrix \mathbf{A}), n of which would be self-edges (diagonal elements of \mathbf{A}). In a directed graph, a cycle is a sequence of arrows that starts from a node, follows the arrows in their proper direction, and returns to the first node; graphs without cycles are called *acyclic*.

When graphs are drawn, the arrangement of nodes and edges is a matter of convenience. Larger biological networks are usually nonplanar – that is, edge intersections cannot be avoided – and designing well-drawn layouts can be a challenge [5]. In metabolic charts, cofactors are often omitted or displayed multiple times, which greatly simplifies the layout. A unique graph layout, suited to visually compare networks by their statistical properties, is provided by hive plots (www.hiveplot.net).

A graph can be characterized by some basic statistical properties (see Table 8.1). Nodes sharing an edge with node i are called its *neighbors*, and the number of neighbors is called the *degree* or *node size* k_i . In directed graphs, we distinguish between in-degrees k_i^{in} and out-degrees k_i^{out} , referring to incoming and outgoing edges, respectively. In finite graphs – graphs with a finite number of nodes – the count numbers of nodes with degree k form the *degree sequence* $n_k(k)$. A directed graph with n nodes and m edges has an average degree of $\langle k_{\text{in}} \rangle = \langle k_{\text{out}} \rangle = m/n$, and the probability that two

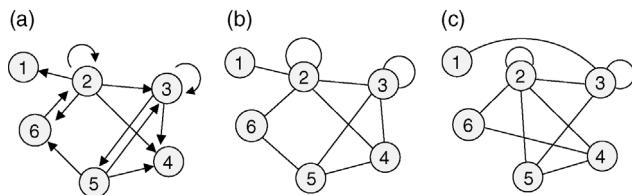


Figure 8.2 Simple graphs. (a) Directed graph. (b) Undirected graph with the same topology. (c) Rewired variant of graph (b) where all degrees (number of edges per node) are preserved.

Table 8.1 Statistical properties of graphs.

Graph	n	$\langle k \rangle$	ℓ	$\langle c \rangle$
Example Figure 8.2b	6	3	23/15	1/5
<i>E. coli</i> metabolite graph [6]	282	7.35	2.9	0.32
Movie actors [7]	225 226	61	3.65	0.79
Power grid [7]	4941	2.67	18.7	0.08

Graphs can be characterized by node number n , average degree $\langle k \rangle$, diameter ℓ , and average clustering coefficient $\langle c \rangle$. Numbers refer to different real-world graphs.

Source: Data from Ref. [8].

randomly chosen nodes share an edge is $m/n^2 = \langle k \rangle/n$. In infinite graphs, the probabilities $p(k)$ for randomly picked nodes to have degree k are called the *degree distribution*.

8.1.1.2 Random Graphs

A random graph, similar to a random number, is defined as a probability distribution over a set of graphs. The single graphs are called realizations. For brevity, we sometimes refer to random graph realizations as “random graphs” and call the random graph an “ensemble.” In some random graphs, the probabilities are specified by a general rule: For instance, to define a random graph with n nodes, we may assign equal probabilities to all possible graphs with exactly m edges and zero probability to other graphs. Another way to define probabilities is by a random process for graph construction: For instance, we may start with a set of unconnected nodes and add edges according to some probabilistic rule.

8.1.1.3 Erdős–Rényi Random Graphs

An *Erdős–Rényi* random graph $\mathcal{G}(n, q)$ is a random graph with n nodes in which possible edges are realized independently and with probability q . The elements of the adjacency matrix are independent binary variables with probabilities $p(a_{ij} = 1) = q$. The number of edges in an Erdős–Rényi graph follows a binomial distribution $p(m) = \binom{n^2}{m} q^m (1-q)^{n^2-m}$ with a maximum edge number n^2 ; similarly, the out-degrees follow a binomial distribution $p(k) = \binom{n}{k} q^k (1-q)^{n-k}$. This distribution has a peak at the mean degree $\langle k \rangle = qn$ and a standard deviation $\sigma_k = \sqrt{nq(1-q)}$. For large graphs ($n \rightarrow \infty$) with a fixed average degree, the degree distribution becomes a Poisson distribution, showing an exponential tail for large degrees.

In an Erdős–Rényi graph, a possible n -node graph has the probability $p = q^m (1-q)^{n^2-m}$, which depends only on

its edge number m . The edge number follows a binomial distribution $p(m) = \binom{n^2}{m} q^m (1-q)^{n^2-m}$ with mean qn^2 and standard deviation $\sqrt{q(1-q)n^2}$. The mean degree (number of edges per node) reads $m/n = qn \pm \sqrt{q(1-q)}$, and for large graphs ($n \rightarrow \infty$), the standard deviation becomes negligible. Therefore, a large Erdős-Rényi random graph can be approximated by a random graph $\mathcal{G}_E(n, m)$ with fixed edge number m . Here, in each realization, $m = qn^2$ edges are randomly distributed over the n^2 possible pairs of nodes. If we consider large directed graphs ($n \rightarrow \infty$) with predefined mean degree $\langle k \rangle$, both types of random graph yield similar results; the parameters must be chosen such that $\langle k \rangle = qn = m/n^2$.

8.1.1.4 Geometric Random Graphs

The topologies of some networks reflect an underlying spatial structure. For instance, the connections between nerve cells (which can be depicted as a graph) may reflect cell distances in space. Geometric random graphs [9] are defined based on such spatial relationships: Nodes correspond to points in a space (e.g., in the plane \mathbb{R}^2) with geometric distances d_{ik} . Two nodes are connected with a probability $p(d_{ik})$ depending on their geometric distance. Assuming a Gaussian probability density $p(d_{ik}) \sim \exp(-d_{ik}^2/(2\sigma^2))$, typical random graph realizations will contain many local connections, but very few connections between distant points.

8.1.1.5 Random Graphs with Predefined Degree Sequence

For statistical tests, we need random graphs that resemble a given network in its basic statistical properties. Random graphs preserving the in- and out-degrees of all nodes can be constructed by a random flipping of edges [10]: In each step, two edges are chosen at random and replaced by two edges with the same origin nodes,

but with their target nodes flipped. This flipping changes the graph, but leaves the degree of each node unchanged (see Figure 8.2b and c). After many iterations, we obtain a randomized graph in which in- and out-degrees are unchanged, but more complex structures have been destroyed. Other random graphs, in which more complicated properties are preserved, can be obtained by simulated annealing [10].

8.1.2 Scale-Free Networks

Many real-world networks, including metabolic networks, social networks, and the Internet, show degree distributions with characteristic power laws [11]:

$$p(k) \sim k^{-\gamma} \quad (8.1)$$

and scaling exponents $2 < \gamma < 3$ (see Section 10.2.5). By taking logarithms on both sides, we obtain

$$\log p(k) = -\gamma \log k + \text{const.} \quad (8.2)$$

In a double-logarithmic histogram plot, power-law distributions show a simple linear decrease (see Figure 8.3). If a network shows this linearity over several orders of magnitude, in particular for large degrees k , a power-law distribution may be suspected. It can be vigorously tested by statistical model selection [12].

Power-law distributions exist, for instance, in word frequencies (Zipf's law: The frequency of words in natural language is inversely proportional to the words' count number ranks) and economy (Pareto's law: The number of people with income larger than x scales with x according to a power law). The power-law distribution (8.1) is *self-similar* under a rescaling of k , satisfying $p(\lambda k) = \lambda^{-\gamma} p(k)$ (see Section 10.2.5). Therefore, it does not define

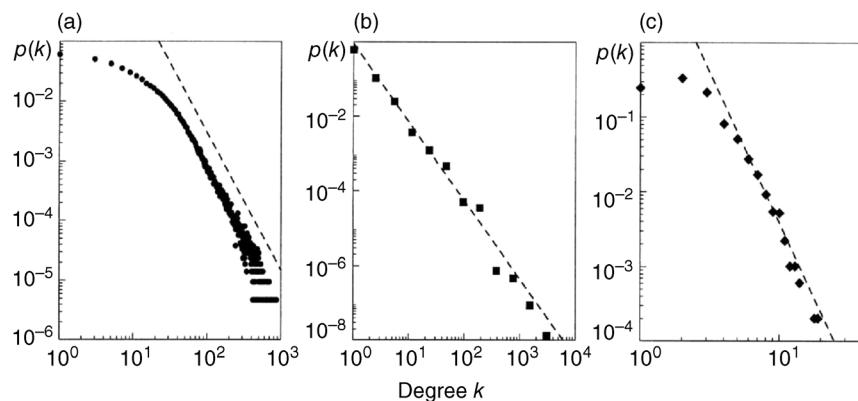


Figure 8.3 Scale-free degree distributions in real-world networks. (a) Collaborating movie actors ($n = 212250$, $\langle k \rangle = 28.78$, $\gamma = 2.3$). In the network, two actors are linked if they have played together in a movie. (b) World Wide Web ($n = 325729$, $\langle k \rangle = 5.46$, $\gamma = 2.1$). (c) Power grid data ($n = 4941$, $\langle k \rangle = 2.67$, $\gamma = 4$). (From Ref. [11].)

a typical range (or “scale”) for the degrees: For finite networks, the mean value $\langle k \rangle$ increases with the network size, and for infinite networks, it diverges. This is why power-law distributions are called *scale-free*.

Scale-free networks contain a few very large (i.e., high-degree) nodes called *hubs*, many nodes with very small degrees, and a hierarchy of differently sized nodes in between. However, this hierarchy arises only for scaling exponents $2 < \gamma < 3$. For small $\gamma < 2$, a “hub-and spokes” network with a single hub will arise. For larger $\gamma > 3$, hubs are not relevant and the network resembles an Erdős–Rényi network. Erdős–Rényi networks show a different, peaked degree distribution with a mean degree $\langle k \rangle$: Most nodes have relatively similar degree, and nodes with large degrees practically do not exist. The mean degree $\langle k \rangle$ and the degree dispersion $\text{var}(k)$ follow from the binomial distribution of degrees, independent of the graph size. In scale-free networks, both quantities increase with the network size.

8.1.2.1 Preferential Attachment Model

Many real-world networks show scale-free degree distributions that distinguish them from Erdős–Rényi random graphs. A possible explanation for these structures refers to the ways in which networks are growing. In the *preferential attachment* model [11], a network grows by successive addition of nodes: A new node attaches randomly to one of the nodes, but with a preference for nodes that have many connections already. Therefore, nodes with large degrees have higher chances to increase their degree (“the rich get richer”). In simulations, preferential attachment with a linear relation between preference and node size leads to graphs with scale-free degree distributions. If the preference increases more strongly than linearly, a single node will become connected to almost all other nodes, while these share very few connections among themselves.

Can scale-free degree distributions in biochemical networks be explained by preferential attachment? In metabolic networks, this growth model would require that newly evolving enzymes metabolize compounds that are already widely used. This has not been shown directly, but the fact that existing hub metabolites have arisen early in evolution [6] supports the preferential attachment assumption. In protein–protein interaction networks, preferential attachment can be realized by gene duplication [13,14]. If the gene for a protein A is duplicated in a genome, all its interaction partners B will obtain one more edge. Now assume that genes are duplicated at random: If a protein B has many interaction partners, it is likely that one of them will be duplicated next; if B has few interaction partners, this is less probable. Thus, the probability of obtaining new interaction partners is proportional to the number of existing

interaction partners, just as required for preferential attachment.

The evolution of real-world networks is much more complex, involving specific preferences between nodes, dynamic rewiring, and removal of nodes and edges. However, the idea of preferential attachment shows that scale-free networks can emerge from a simple evolutionary mechanism without selection. Therefore, statistical network features that would automatically arise in any scale-free networks cannot be taken as a proof of evolutionary selection.

8.1.3

Connectivity and Node Distances

8.1.3.1 Clustering Coefficient

In geometric random graphs, the neighbors of a node have a higher chance to be connected as well. This phenomenon, called *clustering*, also appears in many real-world graphs and can be quantified by the *mean clustering coefficient*. For a node i , the number r_i counts all connections between its neighboring nodes, or in other words, the triangles (three loops) comprising node i . In undirected graphs, a node with degree k_i can maximally have a value of $r_i^{\max} = k(k - 1)/2$. Watts and Strogatz [7] defined the clustering coefficient of node i as the ratio $c_i = r_i/r_i^{\max}$, that is, the fraction of possible edges between neighbors that are actually realized. Self-edges are not counted in the clustering coefficient. Graphs with scale-free degree distributions and strong clustering can be constructed with the *hierarchical network model* [15].

If graphs are clustered, this may indicate an underlying similarity relation between nodes. In social networks, people who live nearby or share similar interests (small distances in physical space or in some abstract “interest space”) are more likely to share other relationships as well; these relationships will therefore be clustered. Clustering arises automatically when bipartite graphs are collapsed (see Figure 8.4): A *bipartite graph* contains two types of nodes (e.g., A and B, black or white), and edges connect nodes of different types. In the collapsed graph, all nodes of type B are removed and the neighbors of a removed node become connected by edges. An example is the graph of collaborating movie actors in Figure 8.3, which stems from a bipartite graph of actors and movies. When collapsing a bipartite graph, each collapsed node gives rise to a fully connected subgraph, realizing a clustering coefficient of 1.

8.1.3.2 Small-World Networks

The topological distance of two nodes is defined as the length of the shortest path between them. For some node pairs, the distance may not be defined, and in directed graphs it is not a symmetric function. An example of a topological distance is the “Erdős number,” the

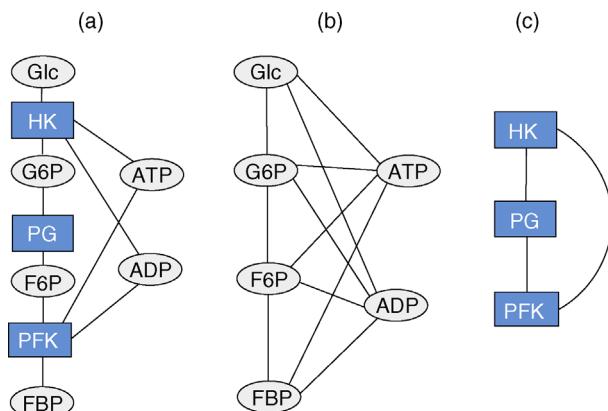


Figure 8.4 Metabolic pathways represented by graphs. (a) Three reactions from upper glycolysis (see Section 4.2.2) shown as a bipartite graph of metabolites and reactions. (b) Collapsed metabolite graph. (c) Collapsed reaction graph. HK: hexokinase; PG: phosphoglucoisomerase; PFK: phosphofructokinase; Glc: glucose; G6P: glucose 6-phosphate; F6P: fructose 6-phosphate; FBP: fructose 1,6-bisphosphate.

collaborative distance of mathematicians to the mathematician Paul Erdős, the father of Erdős–Rényi random graphs. People who published together with Erdős (504 people) have an Erdős number of 1, while those who did not, but published with persons who published with Erdős, have an Erdős number of 2, and so on. Typical Erdős numbers are relatively small (mean 4.65, maximum 15 among mathematicians with a finite Erdős number). Incidentally, low Erdős numbers have been offered at eBay.

The *diameter* of a graph is the longest distance between two nodes in the network. How will this distance depend on the number of nodes and on the graph structure? In Erdős–Rényi random graphs with n nodes and average degree k , the diameter scales logarithmically with n : In a simple approximation, a node has approximately k neighbors, k^2 second neighbors, k^3 third neighbors, and so on. The number of reachable nodes grows exponentially with the node distance, and virtually any point of the network can be reached after relatively few steps.

For clustered networks, the expectation would be different: Compared to Erdős–Rényi random graphs with the same average degree, we expect larger diameters; when counting the first, second, or higher neighbors, we are likely to remain close to our starting point and to count the same nodes several times. However, some real-world networks show a clustered structure and a small diameter (compared to Erdős–Rényi graphs with the same average degree). Watts and Strogatz [7] called such networks “small-world” networks and showed that they can be generated from locally structured networks by adding relatively few global connections.

Example 8.1 Connectivity of Metabolic Networks

A biochemical network, defined by its stoichiometric matrix N , can be depicted as a bipartite graph of metabolites and reactions (see Figure 8.4a and b). By collapsing this graph, we obtain a metabolite graph (Figure 8.4c) in which edges connect metabolites that share a common reaction. A reaction graph (Figure 8.4c) can be obtained accordingly (compare with Figure 3.5). The collapsed metabolite graphs in cells display scale-free degree distributions over two orders of magnitude: Such distributions have been found in 43 species [16] of various sizes and from all three kingdoms of life. There are few metabolites – mostly cofactors – that participate in a large number of reactions, and their order of importance (as measured by the degree) is almost identical in all organisms. These metabolites appear in virtually all metabolic networks studied and may have appeared early in evolution [6], in agreement with the preferential attachment model.

Probably due to these hub metabolites, metabolic networks display small-world properties [16]: Despite their very different sizes, all 43 networks studied had almost the same small diameter. Even if a considerable percentage (8%) of nodes was randomly removed, the network diameter remained constant. However, if specifically the hub metabolites were removed (a “directed attack”), the diameter rose quickly. Since the collapsed metabolite graph does not distinguish between the substrates and products of reactions, shortest paths in this graph need not represent actual metabolic routes. When such routes were considered, the small-world property was not found [17].

8.1.4 Network Motifs and Significance Tests

8.1.4.1 Network Motifs

Many real-world networks contain characteristic local wiring patterns. If a pattern is significantly abundant, it is called a *network motif* [18]. Since their discovery in transcription networks, network motifs have been explored in many real-world networks and networks have been classified by motifs they contain [19]. Characteristic motifs in transcription networks, such as self-inhibition and the feed-forward loop (FFL) [20,21], will be discussed in Section 8.2.

To test if a pattern is significantly abundant in a network, the network is compared with a random graph that serves as a background model. In each realization of the random graph, the pattern will appear with a certain count number; this defines a probability distribution $p(n)$ for the count number n . If the count number in the original network is larger than that in 95% of the random graphs, the pattern, as a motif, is significant at a 5%

confidence level. For Erdős–Rényi random graphs, the probabilities of local structures can be computed analytically. The same type of significance test also applies to other kinds of network structures, for instance, to highly connected subgraphs called *network modules* [22].

8.1.4.2 Null Hypotheses for Detecting Network Structures

We saw that networks can be characterized by statistical properties such as degree distribution, occurrence of motifs, or highly connected subgraphs. For statistical reasons, some of these properties may be related, and prominent structures like modules can result from basic features such as the degree sequence.

To focus our significance tests on structures that are not simply by-products of some basic network statistics, we need to compare the original network with random graphs with the same basic statistics. These random graphs represent a specific null hypothesis, the hypothesis that our structure is in fact a by-product of basic statistical properties. Again, if less than 5% of realization meets our criterion for detecting the structure, the structure is taken to be significant at the 5% confidence level. So, which kinds of random graphs should we choose to compute the significance of network motifs? A geometric random graph contains more self-inhibition loops than an Erdős–Rényi graph with the same mean degree. Using one or the other random graph as a null model would lead to different results; the number of self-inhibitions in a real-world graph could be significant in one case, but not in the other. The selection of background model depends on which network features we take for granted.

Here is an example. The structure of transcription networks is determined by binding sites for transcription factors, and it evolves by mutation or duplication of promoter sequences [23]. However, the possible network topologies seem to be restricted: In-degrees (numbers of regulators per gene) are typically small, whereas out-degrees (number of targets per regulator) can be larger. To study motifs in transcription networks, Milo *et al.* [10] constructed random graphs in which all node degrees from the original network were preserved. If motifs are significant with this background model, they do not simply follow from the degree distribution. With the random graph representing a scenario of neutral evolution, network motifs, being significantly unlikely structures, can be taken as signs of selection at work.

8.1.5 Explanations for Network Structures

Many structures in biological networks can be explained by evolutionary history or biological function. However,

structures like power-law distributions or network motifs also appear in completely different types of networks such as social networks, food webs, public transport networks, or the Internet; there must be other, nonbiological principles behind their emergence. In general, four groups of principles can be considered:

- 1) *Definition of the network* Network structures can arise from how networks are defined or mathematically constructed. For instance, networks that reflect distances (e.g., the wiring scheme of neurons) or similarities (e.g., correlations between gene expression profiles) or that are obtained by collapsing a bipartite graph will also show clustering.
- 2) *Material constraints* Network structures can result from material constraints. Transcription factors, for instance, can have many binding sites in the genome, but the number of binding sites per promoter may be limited. In metabolic networks, chemical reactions are constrained by the conservation of atom numbers.
- 3) *Common origin or similar growth processes* Some structures may reflect the ways in which networks evolve. On the one hand, common features (e.g., the cofactors used in metabolic networks) may stem from common ancestors. Once many processes rely on some feature, this feature cannot be changed anymore and will be conserved (once-forever selection). Also, the growth process itself can induce structure: If biological, social, and technical networks grow by preferential attachment – maybe for very different reasons – this may result in a common degree distribution, which can then induce further similarities.
- 4) *Analogous function and shaping for optimality* Another possible explanation for network structures is a selection for usefulness or cost-efficiency. Systems with similar tasks can evolve independently toward *analogous* structures. The feed-forward loop motif, for instance, appears in many transcription networks, but also in neural connections of the worm *Caenorhabditis elegans*. In both cases, loops may evolve because they can perform specific signal processing tasks with a low material effort. Generally, since network connections (e.g., enzymes establishing chemical reactions, or streets connecting cities) are costly, cost pressures may lead to sparse networks and short connections between central nodes.

Each of the explanations follows a different logic: Aristotle proposed that “why” –questions can be answered by four different types of explanations, traditionally called “causes.” Our explanations of network structures exemplify these types: Network structures arise from the mathematical forms of networks (*causa formalis*: “formal cause”); from their realization by physical objects (*causa*

materialis: “material cause”); from factors that shaped networks in evolution or in their construction (*cause efficiens*: “effective cause”); or because of a network’s usefulness, for example, as a communication or transport system (*causa finalis*: “final cause”). These explanations are complementary, and linking them can provide new insights. For instance, scale-free degree distributions in protein–protein interaction networks may stem from preferential attachment *and* provide robustness against node failure. This could mean that preferential attachment itself is a favorable mode of network evolution *because* it promotes favorable network properties.

8.1.5.1 The Network Picture Revisited

We saw that networks emphasize the structure of interactions while neglecting the nature of the elements and the dynamics of the system. Such an abstraction can be useful for various reasons: (i) Networks may be good starting points to describe systems when little quantitative information is available. (ii) Networks, especially in their graphical representation, are better understandable than detailed quantitative descriptions. (iii) Studies of network structure may reveal similarities between apparently unrelated systems. (iv) Studies of network structure may show which structural features can emerge from basic features such as the degree distribution. (v) Some dynamical processes (e.g., spreading of diseases) [24] depend much more on network structures than on the details of the quantitative process. (vi) Studies of network evolution show how structures emerge from network growth or rewiring and can help to infer selection pressures on specific biological functions. A comparison with random networks may indicate that certain structures are under selection.

In any case, a convincing explanation of network structures must refer to the underlying dynamical systems and evolutionary processes. In particular, networks constructed from statistical correlations should not be overinterpreted: For example, correlated metabolite fluctuations need not imply that the involved metabolites participate in the same pathway. To relate “data networks” to the biological systems behind them, mechanisms and dynamics need to be understood.

8.2 Regulation Networks and Network Motifs

Summary

Biochemical signals encoded in concentrations, modifications, and localization of molecules can be processed by signal transduction pathways and transcription networks.

Signaling compounds activate or inhibit each other, for example, by catalyzing each other’s production (in the case of genetic networks) or chemical modifications (e.g., in MAP kinase cascades). By plotting these interactions, we obtain regulation networks. Common local wiring schemes can allow for specific dynamic behavior or regulatory functions. The adaptation motif, for instance, translates steps in an input signal into transient responses, while its output for constant inputs is completely independent of the input value. Other regulation motifs comprise negative feedback loops, which speed up responses and contribute to stability and oscillations, and feed-forward loops, which can act as filters, sign-sensitive delays, or pulse generators.

The various cellular processes are orchestrated by a complex regulation system. Apart from a direct allosteric regulation of enzymes, there exist signaling pathways, specialized circuits for cell cycle control, growth regulation, or stress response, and a transcription network that adjusts protein levels to current demands. Metabolism, signaling systems, and transcriptional regulation form a large feedback loop, and within this loop, regulation occurs on multiple levels and time scales: A metabolite can, for example, inhibit its own production pathway via either direct allosteric regulation or slower transcriptional regulation. The signaling system is very complex, but we can focus on parts, which we frame as signaling pathways, and see how they process information.

Signaling molecules engage in molecular interactions such as complex formation, protein phosphorylation, or binding to DNA. Specific interactions, enabled by the shapes and binding properties of proteins, can be seen as a form of recognition. In evolution, the strength and specificity of these interactions can be adapted by genetic changes of protein or promoter sequences. Signaling substances or complexes, possibly in different modification states, can be represented as nodes of a network. Edges, possibly with plus or minus signs for activation or inhibition, indicate that substances affect each other, for example, by catalyzing each other’s production or degradation. If several arrows point to one node, multiple input values must be processed at this node, and the processing may be described by Boolean functions. Substances are often regulated by opposing processes such as synthesis and degradation or phosphorylation and dephosphorylation. Compared to metabolic networks, regulation networks can be rewired rather easily by adding or removing individual arrows or by varying their strengths.

Signaling Systems Process Information

Signaling systems translate input stimuli (e.g., the concentration of an extracellular ligand) into output signals

(e.g., active transcription factors binding to DNA). Information can be encoded in concentrations, modifications, and localization of proteins, and either in stationary levels or in temporal patterns. Signaling pathways can sense such signals and transmit, process, and integrate this information. On the one hand, signals are transmitted from one place in the cell to the other (e.g., from a receptor at the cell surface to the transcription machinery in the nucleus). On the other hand, the *input–output relations* of signaling systems can realize information processing tasks such as discrimination, regression, data compression, or filtering of temporal signals, providing informative inputs for downstream processes.

The output of a signaling pathway contains information about the input. Here the term “information” can be taken literally, in the sense of Shannon information: Knowing the output signals would reduce our uncertainty about the input signals. Information transmission through signaling pathways can be measured in units of bits or Shannons [25]. Apart from the statistical Shannon information, signaling systems also provide *useful information*, enabling other systems to respond in adequate ways to the current situation – for example, to express stress response proteins when cells are under threat. Pragmatic information, that is, information supporting advantageous decisions, is quantified by the *value of information*, a concept from Bayesian decision theory (see Section 10.3) [26,27]. In terms of function, regulation networks could also be seen as optimal controllers, for instance, controllers steering metabolic pathways (see Sections 11.1 and 15.5).

8.2.1 Structure of Transcription Networks

Transcription networks describe how gene expression is regulated by transcription factors. Nodes represent genes and arrows indicate that transcription factors (which are again encoded by genes) can bind to a gene’s promoter region and, possibly, regulate its expression. In quantitative models, the arrows are described more precisely by gene regulation functions (see Section 9.3). The network structure is determined by binding sites in the genes’ promoter regions. Binding site sequences of many transcription factors are known [28], and transcription factor binding can be measured *in vivo* on a genome-wide scale [29–31].

When the transcription network of *E. coli* bacteria (Figure 8.1b) is reduced to transcription factors and properly arranged, a clear functional design becomes visible [32]. As shown in Figure 8.5, information is processed in three subsequent layers: An input layer, formed by two-component systems, feeds signals about the cell’s

state or environment into the network. A second, densely connected layer – resembling an artificial neural network – generates various outputs that integrate the input information. The third layer consists of target genes that are regulated directly or through feed-forward loops. The entire system consists of parallel blocks related to general cell functions. Each block contains a *dense overlapping regulon* as its core, which responds to some input stimuli and controls the expression of functionally related genes [32]. One such subnetwork, which regulates the expression of five sugar utilization pathways and contains a large number of feed-forward loops, is shown in Figure 8.6.

Transcription networks contain typical motifs such as negative autoregulation or feed-forward loops [34–37]. Larger clusters formed by such motifs can be described as generalized motifs [38]. The network motifs found in *E. coli* also appear in other organisms. To reconstruct the transcription network in the yeast *Saccharomyces cerevisiae*, Lee *et al.* [29] studied the binding of transcription factors to DNA *in vivo* by chromatin immunoprecipitation. The reconstructed network contains about 4000 interactions between regulators and promoter regions, with an average of 38 target promoters per regulator. The network motifs include the motifs previously found in *E. coli* (examples in Figure 8.7).

Positive and Negative Regulation

Regulation edges can represent activation (+, shown in blue) or repression (–, shown in red), or they can show both modes of regulation (dual regulation) (Figure 8.8). Activating an activator and repressing a repressor lead to net activation, while activating a repressor and repressing an activator lead to repression. When signals pass through a series of edges, the overall sign of the response depends on the number of repressions (even or odd) along the way. Thus, there are multiple ways to realize the same overall response; for example, transcriptional repression of a metabolic pathway by its own product can be realized in two ways: The pathway product, as a ligand, can activate an inducer or inhibit a repressor (see Section 10.3). In theory, both types of regulation should yield the same result. However, evolved networks seem to show preferences: According to Savageau’s demand rule [39], genes that are usually expressed (in an organism’s common environment) are typically controlled by activators, while genes that are usually not expressed are controlled by repressors. In both cases, the binding site is typically occupied. One explanation is that occupied sites reduce the variation in expression levels. Thus, regulation structures following Savageau’s rule may contribute to insulation, that is, to making expression levels robust against biochemical noise [40].

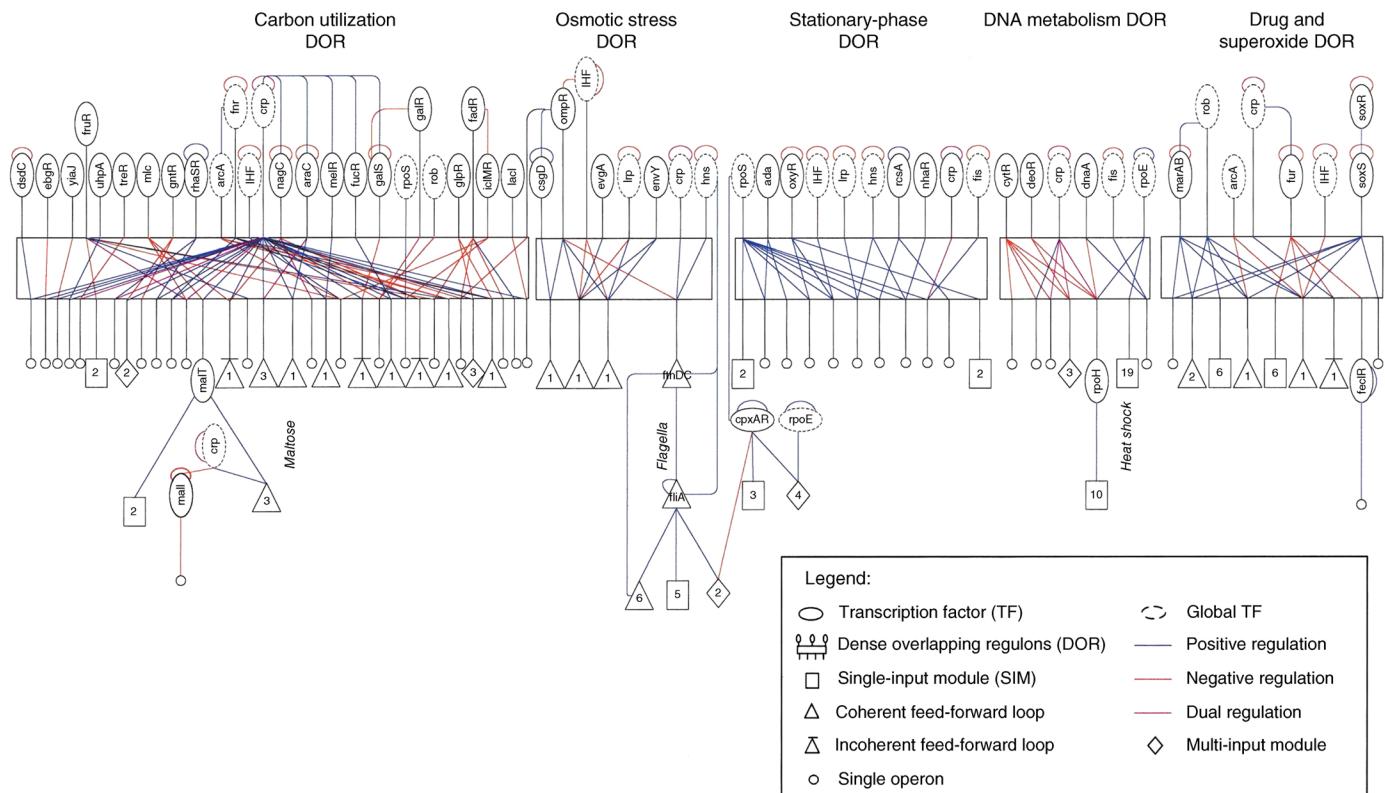


Figure 8.5 Regulation network of transcription factors in *E. coli* bacteria. Extracted from the transcription network (see Figure 8.1a) and arranged in blocks, the network highlights functional subsystems. Information is processed in three layers: Input signals are received via two-component systems, processed and integrated in dense overlapping regulons, and converted into output signals, often through feed-forward loops (marked by triangles). Major cell functions (indicated on top) are controlled by large separate blocks. Activation and repression edges are shown in blue and red, respectively. (From Ref. [32].)

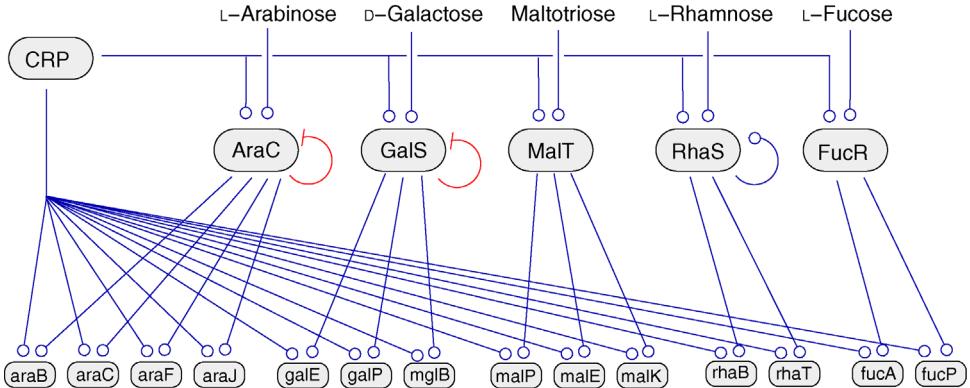


Figure 8.6 Transcriptional regulation of sugar utilization genes in *E. coli* bacteria. Transcription factors signal the availability of specific sugars and control the corresponding pathway genes; all genes are also controlled by CRP, a proxy for energy demand (see Section 9.3). (From Ref. [33].)

Regulation Structures and Network Motifs

Instead of studying regulation networks in their full complexity, we may study small circuits within such networks. For some of them, dynamic behaviors and functions in signal processing have been proposed [41], and some have been realized as genetic circuits in synthetic biology [42–46]. Next, we may study layered networks as in Figure 8.5 and trace how information (encoded in steady-state values or time curves) is transmitted from layer to layer.

When looking for regulation circuits of biological interest, we may focus on patterns appearing in large

numbers, that is, network motifs [34,47,48]. If some of the possible local patterns (see Figure 8.8) are highly abundant in transcription networks, what could be the reasons? One explanation is that the network's statistical properties (e.g., the degree distribution) enforce certain structures as by-products; for network motifs, this sort of explanation can be excluded by choosing the right sort of random graphs in the statistical test. A functional explanation would state that certain circuits underlie active selection during evolution. In fact, some transcription motifs perform specific functions in signal processing [49]. Moreover, network motifs can stabilize networks against dynamic perturbations, which may constitute an additional selection advantage [50,51].

To make such claims, we need to assume that a motif's dynamic behavior is mostly determined by its structure, while kinetic details play a minor role. This, of course, needs to be verified. A first step is to simulate a motif with different rate laws and parameters. Next, it can be simulated under perturbations, or even implemented as a

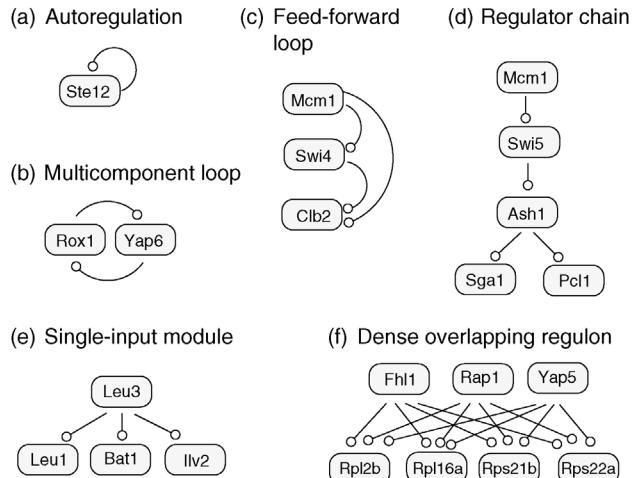


Figure 8.7 Network motifs in the transcription network of the yeast *S. cerevisiae*. Gene names refer to specific examples in the network. (a) Autoregulation: a transcription factor regulating its own expression. (b) Multicomponent loop: a cycle involving two or more factors. (c) Feed-forward loop. (d) Regulator chain. (e) Single-input module: one regulator controlling several genes. (f) Multi-input motif (dense overlapping regulon): several regulators controlling a number of target genes. (Redrawn from Ref. [29].)

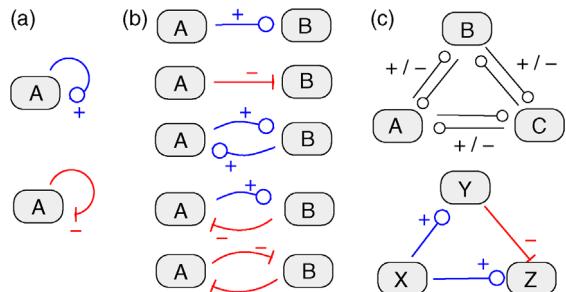


Figure 8.8 Potential regulation patterns with one, two, or three nodes. (a) Positive and negative autoregulation. (b) Possible two-node patterns. (c) Three-node patterns can contain up to six arrows. The incoherent feed-forward loop type I (bottom) is a motif in transcription networks (also see Figure 8.12).

genetic circuit in living cells to see if it performs its predicted behavior robustly. Preferably, the motif should also be minimal, in the sense that it requires lower material efforts than other structures performing similar tasks.

If a regulation circuit promotes favorable behavior, it may appear in different places. For instance, two genes that inhibit each other can form a bistable genetic switch (see Section 6.4 and the example in Figure 2.2). Such a switch can also be useful in other contexts; if it works in transcription networks, it could also be realized, for instance, by mutual inhibitions between immune cells [52].

8.2.2 Regulation Edges and Their Steady-State Response

Before we study regulation circuits, let us see how single edges can be modeled as little dynamic systems. An arrow connects an input S (signal) to an output R (response). Examples of such signal/response elements are kinase/target protein, transcription factor/target gene, and mRNA/protein. In a dynamical model, we can describe signal and response by their strengths s and r , following a rate equation:

$$\frac{dr}{dt} = f(s, r). \quad (8.3)$$

For each input value s , the steady-state condition $0 = f(s, r)$ yields a stationary value of r , and the resulting steady-state response curve $r^{st}(s)$ is the input–output relation for this arrow.

Regulation arrows can symbolize different reaction patterns. Figure 8.9 shows three such patterns, which may

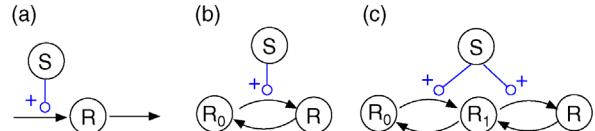


Figure 8.9 An activation arrow $S \rightarrow R$ can represent different underlying reaction patterns. (a) Linear pattern: A substance S (signal) induces the production of R (response). (b) In a loop pattern (e.g., a phosphorylation cycle), S converts inactive R_0 into the active form R . (c) In a double-loop pattern (e.g., double phosphorylation), R is activated in two steps. Black arrows indicate chemical reactions, blue catalysis.

serve as building blocks for larger network models. Their steady-state response $r^{st}(s)$ in the three systems depends on both their reaction scheme and the rate laws (Table 8.2). A linear mechanism with linear kinetics leads to a linear response. Saturable responses can be obtained by Michaelis–Menten kinetics or by a reaction loop with linear kinetics and a conservation relation $r + r_0 = r_t$ for different forms of R (Figure 8.9b). The same loop, with Michaelis–Menten kinetics, yields a sigmoid response curve (Goldbeter–Koshland kinetics). All types of responses (linear, hyperbolic, and sigmoid) depend gradually on the signal strength. Moreover, the steady-state output depends only on the current input signal and not on its previous history. As soon as the signal S stops, the response is switched off: There is no hysteresis.

8.2.3 Negative Feedback

Negative feedback is common in transcription networks and as a regulation pattern in metabolic pathways. In

Table 8.2 Kinetic implementation of signaling arrows.

Structure	Kinetics		Response	$r^{st}(s)$
Linear	Linear	$dr/dt = k_0 + k_1 s - k_2 r$	Linear	$\frac{k_0 + k_1 s}{k_2}$
	MM	$dr/dt = \frac{V_1 s}{K_{m,1} + s} - \frac{V_2 r}{K_{m,2} + r}$		$\frac{V_1 K_{m,2} s}{V_2 K_{m,1} + s(V_2 - V_1)}$
Loop	Linear	$dr/dt = k_1 s(r_t - r) - k_2 r$ $r + r_0 = r_t$	Hyperbolic	$\frac{r_t k_1 s}{k_2 + k_1 s}$
	MM	$dr/dt = \frac{k_1 s(r_t - r)}{K_{m,1} + r_t - r} - \frac{k_2 r}{K_{m,2} + r}$ $r + r_0 = r_t$		$r_t G\left(k_1 s, k_2, \frac{K_{m,1}}{r_t}, \frac{K_{m,2}}{r_t}\right)$
Double loop	Linear	$dr/dt = k_3 s r_1 - k_4 r$ $dr/dt_1 = k_1 s r_0 - (k_2 + k_3 s)r_1 + k_4 r$ $r + r_0 = r_t$	Sigmoid	$\frac{r_t k_1 k_3 s^2}{k_2 k_4 + k_1 s k_4 + k_1 k_3 s^2}$

Formulas correspond to the network structures in Figure 8.9. Linear kinetics and Michaelis–Menten (MM) kinetics lead to different response curves. The Goldbeter–Koshland function $G(u, v, J, K) = 2uK/(v - u + vJ + uK + \sqrt{(v - u + vJ + uK)^2 - 4(v - u)uK})$ is used to model ultrasensitive behavior [53,54].

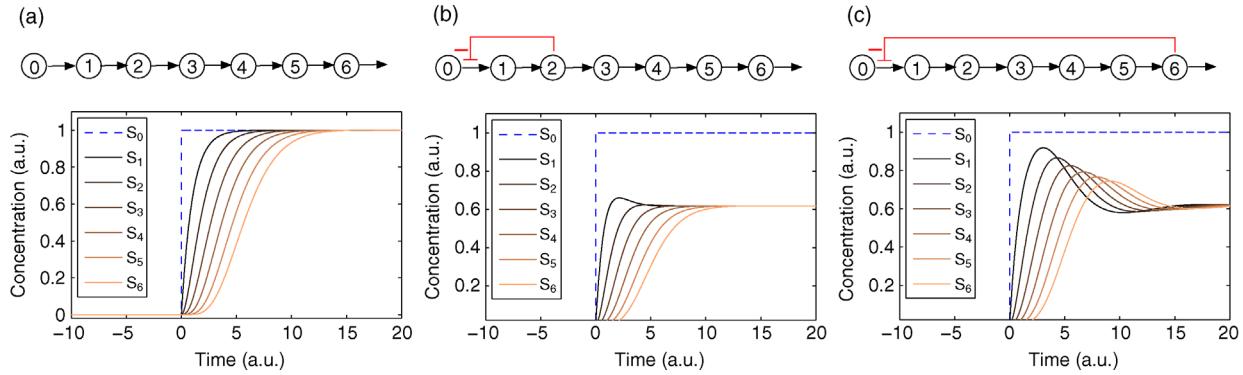


Figure 8.10 Negative feedback in a metabolic pathway. (a) Concentration time series in a chain of reactions. After substrate (- -) becomes available ($s_0 = 1$) at time $t = 0$, the curves rise with different time delays. (b) Negative feedback via the second metabolite decreases the steady-state level and speeds up the response. (c) Negative feedback via the last metabolite leads to an overshoot and damped oscillations. All mass-action constants k_i have values of 1.

synthesis pathways, the first enzyme is often inhibited by the pathway product. This prevents overproduction and stabilizes the product level against fluctuations caused by varying demands. The feedback can be realized by allosteric inhibition, by transcriptional repression of enzymes, or both. Ideally, a feedback system should receive its input exactly from the output variable to be controlled. If this is impossible (as is the case for metabolic fluxes), other variables can be sensed as proxies. For instance, fructose bisphosphate can act as a flux sensor, representing by its concentration the glycolytic flux in *E. coli* [55].

Feedback regulation is used on many levels of physiology: For instance, the stable and material-saving shapes of bones and trees arise from growth processes under feedback regulation by sensed stresses [56–59]. In signaling networks, negative feedback has different functions: It can stabilize a steady state against external and internal fluctuations [43], produce pulse-like overshoots, induce sustained oscillations, and speed up responses [60].

Some of these phenomena can be observed in the pathway model shown in Figure 8.10 (for a detailed analysis, see Ref. [61]). In the model, all reactions follow irreversible mass-action kinetics $v_i = k_i s_{i-1}$, internal metabolites start at levels $s_i = 0$, and after the external substrate is raised to a constant level $s_0 = 1$, the metabolite concentrations approach a new steady state after a short transition period (Figure 8.10a). We also consider variants of this model in which the first reaction is allosterically inhibited by one of the downstream metabolites: The inhibition is implemented by a modified rate law $v_1 = s_1 k_1 / (1 + s_j/K_I)$. If the second metabolite is the inhibitor, the first metabolite shows an overshooting response (Figure 8.10b). With a longer ranging feedback, that is, longer time delays, this effect becomes more pronounced and damped oscillations arise (Figure 8.10c).

The example shows the double nature of negative feedback. A feedback inhibition can shift the eigenvalues of the Jacobian matrix in the complex plane (see Section 15.5). This can stabilize the system state, but a delayed feedback can also lead to damped or sustained oscillations and destabilize the steady state. Metabolic oscillations are observed in reality, but whether they have particular functions, or arise simply from stabilization mechanisms gone wild, is still a matter of debate [62,63].

Figure 8.10 also shows that negative autoregulation can speed up system responses. The response time $\tau_{(1/2)}$ – the time at which the last metabolite S_r reaches its half-maximal level – decreases from (a) to (c). In the arbitrary units used, the values read $\tau_{(1/2)} \approx 5.68$ (no feedback), $\tau_{(1/2)} \approx 5.05$ (short-ranging feedback from second metabolite), and $\tau_{(1/2)} \approx 4.57$ (long-ranging feedback from last metabolite). A similar effect has been shown experimentally in transcription networks: Protein expression responds faster to a stimulus if the protein inhibits its own expression [60]. Fast responses can be crucial for cells in rapidly changing environments. They could also be reached by a faster protein turnover, but this would increase the costs for protein production. Negative autoregulation, in contrast, saves this cost: Initially, protein production is high, but when self-inhibition kicks in, protein synthesis is interrupted and no further production costs arise.

8.2.4 Adaptation Motif

An important characteristic of signaling systems is their transient response to step-like input signals. The system in Figure 8.11 shows a remarkable behavior called *perfect adaptation*: After a step of the input value, it shows some transient dynamics, but after a while, the output returns exactly to its initial value. Perfect adaptation makes

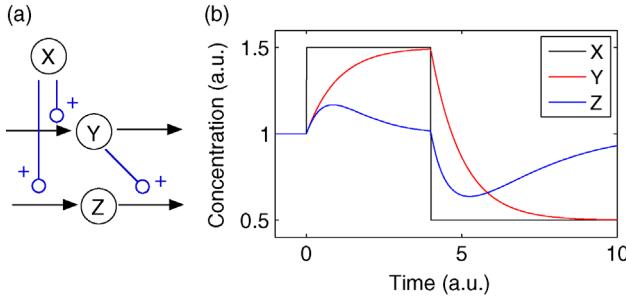


Figure 8.11 Adaptation motif. (a) A signal substance X catalyzes the production of Y and Z, while Y catalyzes the degradation of Z. The reactions follow mass-action kinetics. (b) Temporal dynamics of the adaptation motif. A step-like input level x (black) evokes a sustained response of y (red); the output level z (blue) shows a transient response and returns to its steady-state value (all rate constants set to values of 1).

systems sensitive to temporal changes, but insensitive to the baseline input value. As we will see in Section 10.2.1, this plays a vital role in the bacterial chemotaxis pathway.

In the *adaptation motif* (Figure 8.11), the input X activates the production of Z, but also inhibits it via activation of Y. With mass-action kinetics and linear activation, the levels of Y and Z follow the equations

$$\begin{aligned} \frac{dy}{dt} &= \alpha_y x - \beta_y y, \\ \frac{dz}{dt} &= \alpha_z x - \beta_z yz, \end{aligned} \quad (8.4)$$

which for $x > 0$ lead to the steady state

$$y^{st} = \frac{\alpha_y}{\beta_y} x, \quad z^{st} = \frac{\alpha_z \beta_y}{\beta_z \alpha_y} x. \quad (8.5)$$

In steady state, activation and inactivation cancel out and the level of Z is determined only by rate constants. However, when the input changes, the activation responds faster than the inactivation. This creates a transient peak (Figure 8.11b).

8.2.5 Feed-Forward Loops

The feed-forward loop shown in Figure 8.12 is a common motif in transcription networks [35–37,64]. It consists of three genes that regulate each other: Gene product X regulates gene Z directly and via an intermediate gene Y. Each arrow can represent activation (+) or inhibition (−). In Boolean models, the two inputs of Z can be processed by logical AND or OR functions. In the feed-forward loop, eight sign combinations are possible, but only two are abundant in transcription networks: the so-called

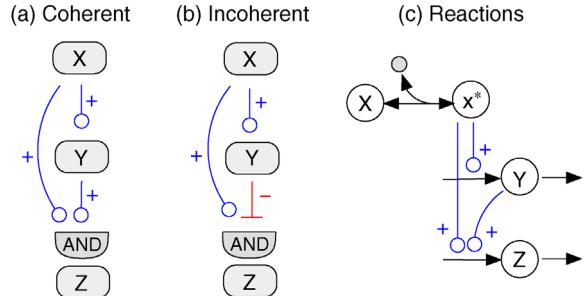


Figure 8.12 Feed-forward loops. An input gene X regulates an output gene Z in two ways: directly and via an intermediate gene Y. (a) Coherent feed-forward loop type 1 with AND gate. (b) Incoherent feed-forward loop type 1 with AND gate. (c) Possible reaction scheme behind a coherent feed-forward loop. X is activated by rapid ligand binding (circle denotes the ligand, X^* its active form). Blue edges represent transcriptional regulation of Y and Z; transcription and translation are lumped into one reaction.

coherent FFL type 1 and the *incoherent FFL type 1* (Figure 8.12). In a coherent FFL type 1, all regulations are activating, while in the incoherent FFL type 1, the edge from Y to Z becomes inhibiting.

At first sight, the second branch, via gene Y, has no obvious function: In the coherent FFL, it seems redundant; in the incoherent FFL, it even cancels the effect of the direct branch. However, this holds only in steady-state situations. If the input X in the incoherent FFL is switched on, gene Y turns up with a delay, so Z is first activated via the direct branch and only later inhibited by Y. Due to the time delay, a step in the input X is translated into a peak of the output Z. Thus, a possible function of feed-forward loops could be the processing of temporal signals [36]: If an external signal (e.g., a ligand concentration) changes the activity of X, the FFL translates the time profile of X into a specific peak profile of Z, which then can serve as an input for downstream processes. Dynamical models and measurements in gene circuits in *E. coli* have shown that feed-forward loops can realize sign-sensitive delays, generate temporal pulses, and accelerate the response to input signals. Moreover, incoherent FFL can create nonmonotonic effective input functions for the target gene Z [65]. The precise behavior of an FFL depends on kinetic parameters or, in the Boolean paradigm, on the signs and the logic regulation function for Z.

Dynamic Model of Feed-Forward Loops

To study the dynamics of feed-forward loops in a simple model, we assume that the product of gene X is expressed constitutively, that it can be rapidly activated by a ligand, and that activities of Y and Z depend directly on their expression levels. Lumping transcription and

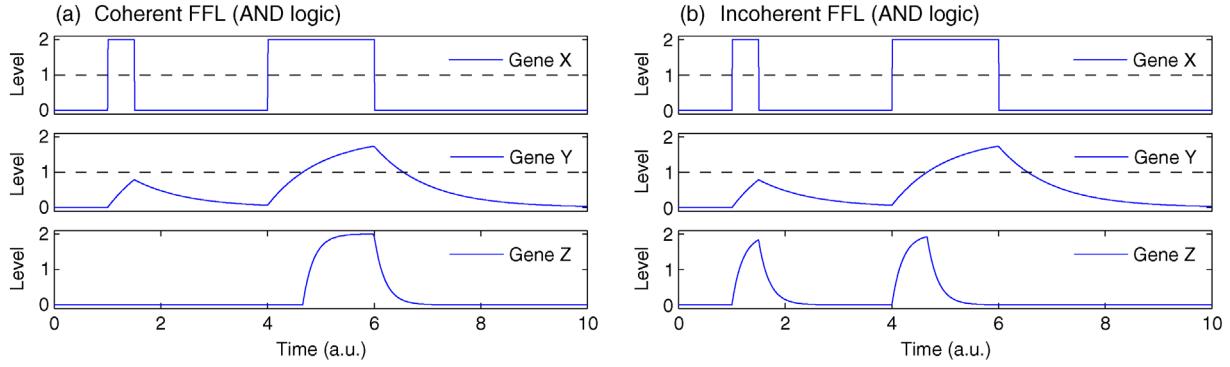


Figure 8.13 Dynamic behavior of feed-forward loops (FFLs). (a) Coherent FFL type 1 with AND logic (see Figure 8.12a). Time curves show the active input X (top), intermediate gene Y (center), and output Z (bottom) in arbitrary units. Short pulses are filtered out; the response to a longer pulse is delayed, but the response to the end of the pulse is immediate. (b) Incoherent FFL type 1 with AND logic. The onset of each input pulse leads to a pulse in Z with a fixed maximal length. Model parameters $\alpha_y = 2$, $\beta_y = 1$, $\alpha_z = 10$, $\beta_z = 5$.

translation into one step, we obtain the following rate equations:

$$\begin{aligned}\frac{dy}{dt} &= f_y(x) - \beta_y y, \\ \frac{dz}{dt} &= f_z(x, y) - \beta_z z,\end{aligned}\quad (8.6)$$

where y and z denote the protein levels of Y and Z , f_y and f_z are the production rates, and β_y and β_z are degradation constants. For a realistic model, protein production could be described by measured gene regulation functions (see Section 9.3). Here we keep the model simple and use a step-like gene regulation function [35]:

$$f_y(x) = \alpha_y \Theta(x > x_0). \quad (8.7)$$

The step function $\Theta(\cdot)$ yields a value of 1 if x is larger than x_0 and a value of 0 otherwise. Thus, when x is below the threshold value x_0 , Y is not transcribed; otherwise, Y is transcribed at a constant rate α_y . We consider two types of FFL, a coherent and an incoherent one, both with logical AND functions. The regulation functions for gene Z read

$$\begin{aligned}\text{coherent : } f_z(x, y) &= \alpha_z \Theta(x > x_0 \text{ AND } y > y_0), \\ \text{incoherent : } f_z(x, y) &= \alpha_z \Theta(x > x_0 \text{ AND } \text{NOT } y > y_0).\end{aligned}\quad (8.8)$$

Figure 8.13 shows simulation results from the model (8.6) with piecewise constant regulation functions (8.7) and (8.8) and predefined input pulses.

Functions of Feed-Forward Loops

The simulations illustrate characteristic features of the feed-forward loop. The coherent-AND FFL shows a delayed response to the onset and an immediate response to the end of pulses, so short input pulses are filtered out.

The incoherent-AND FFL, in contrast, responds immediately to an input pulse, but the response stops after a while: Larger input pulses are translated into standard pulses of similar length. We know this behavior from the adaptation motif, which in fact can be seen as an incoherent feed-forward loop. The dynamics of feed-forward loops in the *E. coli* transcription network has been verified experimentally [36,37,64].

Tightly interlinked feed-forward loops constitute the sporulation system in the bacterium *Bacillus subtilis*. In response to harsh environmental conditions, cells can transform themselves into spores, which can then survive for a long time without metabolic activity. The process, called sporulation, involves several waves of gene expression. In the network, these waves are generated upon stimulation of the sigma factor σ^E by five entangled FFLs (see Figure 8.14).

Paradoxical Regulation

The examples above show that a simultaneous activation and inhibition of the same target – which appears paradoxical at first sight – can be functional to signal processing. The same phenomenon appears in the communication between immune cells [52]: Different cell types can influence each others' growth and death via chemical signal molecules called cytokines. Thus, the growth dynamics of cell types may serve as a signaling device analogous to the expression dynamics of protein levels inside a cell. Unlike the dynamics of protein levels, however, the growth dynamics of cells is inherently unstable because rapidly growing cell types would outcompete all others. In this situation, self-regulation and mutual regulation via cytokines are key to ensure homeostatic cell concentrations, and interaction schemes between cell types (e.g., feed-forward loops) can realize similar dynamics as in gene networks. Thus, apparently paradoxical actions of cytokines

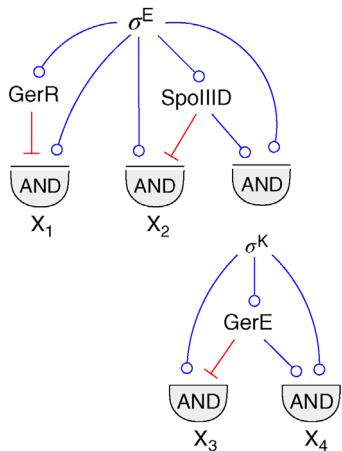


Figure 8.14 Gene network coordinating the process of sporulation. *B. subtilis* bacteria can transform themselves into spores to survive harmful environmental conditions. When sporulation is triggered, a large number of genes are differentially expressed in subsequent waves. The waves are created and coordinated by the regulation network shown [66]. It contains a number of feed-forward loops that activate the downstream target genes. Activation of the master regulator σ^E triggers waves of expression in different groups of target genes (denoted by X₁, X₂, X₃, X₄). (Redrawn from Ref. [66].)

(i.e., causing cell proliferation and death at the same time) can be functionally important.

8.3 Modularity and Gene Functions

Summary

Systems that consist of autonomous subsystems, possibly performing specific functions, are called modular. Enzymes can be seen as biological modules because they can perform the same catalytic function in various contexts. In technical systems, a modular structure facilitates design, interoperability, reusability, and general understanding. In biological networks, modularity can improve robustness and make systems better evolvable. In the description of cells, for example, when biochemical networks are dissected into pathways, modularity is a helpful concept, but needs to be justified. The assumption of modularity, whether or not it holds in reality, is central to the mathematical modeling of cells.

In a modular description, a system is seen as composed of subsystems with characteristic dynamics, specific functions, or sparse or weak connections between them [67]. In nonmodular systems, in contrast, parts are tightly connected and functions are distributed over the system. Modular designs are common in technical systems: Computers, for instance, consist of standardized parts that

exert distinct and defined functions, operate more or less autonomously, communicate via standard interfaces, and can be repeated, replaced, or transferred as independent units. A modular design keeps machines manageable and ensures that parts can be reused in different combinations or be replaced in case of failure.

Modules can be a helpful concept in studies of complex systems: In biology, modules can concern the structure, dynamics, regulation, genetics, and function of organisms. Organisms contain physical modules on various levels: organs, cells, organelles, protein complexes, or single molecules that bear specific functions and can retain them in new contexts (e.g., organs can be transplanted and proteins can be transfected into different cells). Also in cellular networks, we observe modules such as metabolic pathways, signaling pathways, or dense overlapping regulons, which give the transcription network of *E. coli* its pronounced modular structure (see Figure 8.5).

To obtain an overview of how a cell functions, we may first consider general tasks it needs to perform – such as DNA replication, metabolism, transcription and translation, and signal processing. Subdividing these general systems into more specific ones (e.g., metabolic pathways or signaling systems), we obtain a hierarchical classification of cell functions and systems that perform them. Many such classifications exist [68–70], and if proteins are associated with specific functions – for example, catalyzing a reaction, sensing a specific ligand, or acting as a transporter or molecular motor – they can be placed in a functional hierarchy.

Proteomaps [71] visualize proteome data by Voronoi treemaps based on proteins' functional assignments. Figure 8.15 shows how *E. coli* cells allocate their protein resources to different possible functions: A large fraction of the protein mass (as measured by mass spectrometry) is devoted to metabolic enzymes, and another substantial fraction to transcription, translation, and protein processing. Breaking the protein fractions down into more specialized systems, we observe large investments in glycolysis, transporters, and ribosomes, as well as individual highly abundant proteins.

8.3.1 Cell Functions Are Reflected in Structure, Dynamics, Regulation, and Genetics

Biological modules appear on several levels, including network structure, dynamics, regulation, and genetics. In bacterial *operons*, functionally related proteins (e.g., parts of a metabolic pathway) are encoded by a common strand of mRNA and controlled by the same gene promoter. These proteins share very similar expression profiles, and the entire system together with its regulatory region can be transferred to other cells, where it will exert its

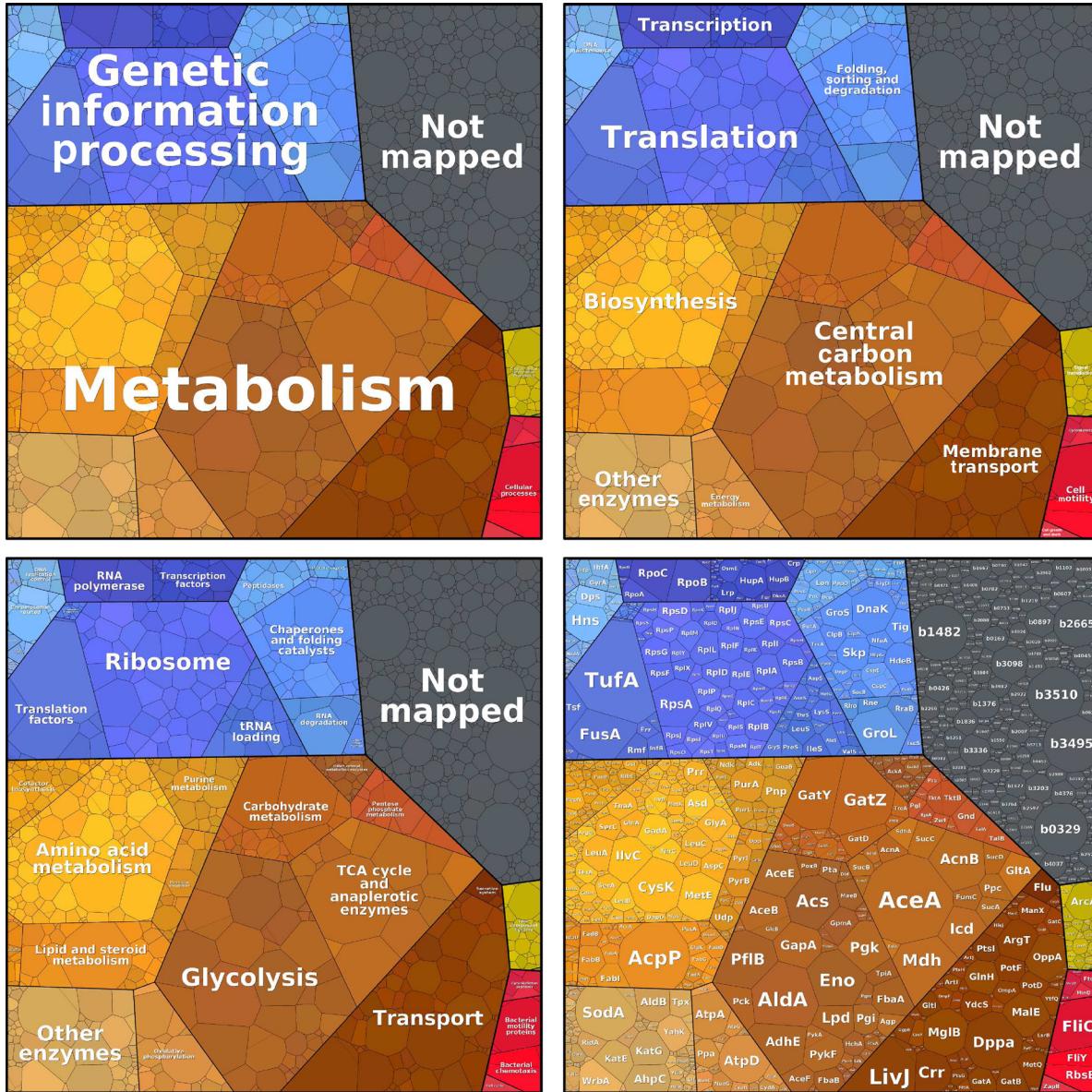


Figure 8.15 Protein investment in different cell functions. The abundance of proteins in *E. coli* bacteria [72], determined by mass spectrometry and arranged by functions, is shown by proteomaps (www.proteomaps.net) [71]. Small polygons represent single proteins; sizes correspond to mass abundance (molecule number multiplied by protein chain length) and give a broad overview of protein investments. The arrangement in larger areas represents a functional hierarchy. The four maps show the same data on different hierarchy levels. In reality, proteins have more than one function; our function assignments are, to an extent, arbitrary.

function in a different biochemical context. The fact that operon structures are established and maintained in evolution suggests a selection advantage – so evolutionary theory should explain how such modules arise.

Notions of function reflect established biochemical knowledge and imply that cell physiology is modular, that is, separate systems exist for metabolism, for processing signals, for establishing cell structure, and so on. Functional assignments are based on evidence from genetics, cell biology, biochemistry, or comparisons

between species. Various criteria have been proposed for defining pathways or modules in cellular networks, ranging from network topologies [73–76] to correlated dynamics [77], regulation systems [78], correlations in high-throughput data [79], and phylogenetic profiles [80]. We can define modularity on the levels of structure, dynamics, regulation, and genetics:

- 1) *Structure* Based on *network topology*, we can define different kinds of modules: dense subnetworks with

- few external connections [76], subnetworks that are connected only via hub elements [73], or recurrent structures such as motifs, single-input modules, or dense overlapping regulons.
- 2) *Dynamic behavior* In dynamic systems, we can define modules by requiring that there is a strong dynamic coupling within, and weak dynamic coupling between modules. Biochemical subnetworks without mass flows between them can be seen as regulation modules. A more empirical criterion is a strongly correlated dynamics within modules [77]: The resulting division into modules may change during the course of a simulation, and there may be a hierarchy of modules defined on different time scales.
 - 3) *Regulation* Gene or protein modules can be defined based on coregulation, either as operons or as the regulon, that is, the target gene set, of a common transcription factor.
 - 4) *Gene exchange and reuse* Due to genetic mechanisms, gene sequences can evolve in modular ways: Chromosomes are inherited as units, but become differently combined in sexual reproduction; in bacteria, DNA sequences can be exchanged between cells by horizontal gene transfer. Mobile elements can copy and duplicate DNA sequences, including coding regions, regulatory elements, or even entire operons. After a gene duplication, and being put into new genetic or functional contexts, genes can become further specialized.
- A modularity on various levels is exemplified by bacterial operons. Operons act as regulatory and genetic modules, and the encoded proteins often form a common pathway. The fact that functionally related genes are located in close vicinity and expressed together can have several advantages:
- 1) To ensure an optimal resource allocation, enzymes must be expressed in appropriate ratios. Otherwise, material and energy would be wasted. Stable expression ratios can be ensured if enzymes share a common regulation system; this reduces uncorrelated fluctuations in their expression. Correlated fluctuations are less problematic because their downstream effects can be compensated by special pathway designs (see Section 10.2).
 - 2) Some bacteria can exchange pieces of DNA (horizontal gene transfer). If genes encoding a pathway are colocalized in the genome, they may be transferred as one unit. However, a successful transfer also requires that the pathway remains functional in cells with a different genetic background. Thus, its dynamics should be relatively robust against changes in the state of the cell in which it resides.
 - 3) If an organism loses a gene by mutation, a second loss-of-function mutation in the same pathway will have little additional fitness effects. This phenomenon, called buffering epistasis, will be discussed below. If a pathway is already incomplete, there will be little selection pressure on preserving the remaining genes: Thus, genes in a pathway should be conserved together or disappear together. This is why a correlated appearance of genes, visible in phylogenetic profiles [80], can be taken as a sign of functional association.

8.3.2

Metabolic Pathways and Elementary Modes

Metabolism can be described in terms of pathways, that is, physiological routes between key metabolites. Metabolic pathways, which also appear in the protein classification in Figure 8.15, can overlap and are linked by cofactors such as ATP. In tightly connected networks, what counts as a pathway is a matter of definition. There are various possible criteria.

First, pathways and modules can be defined based on network connectivity. One possibility is to choose subgraphs with dense internal connections [76]. Another possibility is to eliminate all hub metabolites, for example, metabolites participating in more than four reactions. If enough of these hubs – among them, many cofactors – are removed from the stoichiometric matrix, the remaining network will be split into disjoint blocks [73]. To justify this procedure, we may see the hubs as external metabolites with fixed concentrations, assuming that hub metabolites are either abundant (and therefore insensitive to fluxes), strongly buffered (because stable concentrations are important for many biological processes) or their fluctuations average out (because they participate in many reactions).

Second, metabolic pathways can be defined on the basis of possible fluxes; this leads to notions such as basic pathways [81] or elementary flux modes [82] (see Section 3.1.3). A flux mode is a set of reactions (i.e., a subnetwork) that can support a stationary flux distribution, possibly obeying restrictions on reaction directions. A flux mode is called elementary if it does not contain any smaller flux modes; different elementary modes can be overlapping. Elementary modes can be computed from the stoichiometric matrix, but for larger networks their number grows rapidly. To avoid a combinatorial explosion, one may decompose a network by removing the hub metabolites and then compute elementary modes for the modules [73]. In contrast to textbook pathways, elementary modes are defined not only based on network topology but also by considering stationary fluxes through the entire network.

Flux distributions on elementary flux modes, called elementary fluxes, can be seen as modules in the space of possible chemical conversions: Each of them can convert some external substrates into external products. All stationary flux distributions can be obtained from linear superpositions of elementary fluxes, with arbitrary coefficients for the nondirected modes and nonnegative coefficients for the directed ones. However, this decomposition is not unique; moreover, elementary fluxes and their linear combinations are not guaranteed to respect thermodynamic constraints. A survey of the thermodynamically feasible elementary modes is a good way to characterize the general capabilities of a metabolic network [83].

8.3.3 Epistasis Can Indicate Functional Modules

Can functional associations between genes be inferred objectively, without presupposing any particular gene functions? In fact, deletion experiments can provide such information. If two proteins can compensate for each other's loss, deleting one of them will have little effect on the cell's fitness; however, the effect of a double deletion will be relatively strong. On the contrary, if both proteins are essential for a pathway, a single deletion would already disrupt the pathway's function, and the second deletion would have little effect. Thus, by comparing the fitness losses caused by multiple gene deletions, functional relationships among proteins may be inferred.

Epistasis describes how the fitness effects of gene mutations depend on the presence or absence of other genes. To quantify it, we compare the fitness of a wild-type organism – for example, the growth rate of a bacteria culture – with the fitness of single- and double-deletion mutants. A single-gene deletion (for gene i) will change the fitness (e.g., the growth rate) from the wild-type value f_{wt} to a value f_i , typically giving rise to a growth defect $w_i = f_i/f_{\text{wt}} \leq 1$. For a double deletion of functionally unrelated genes i and j , we expect a multiplicative effect $w_{ij} = w_i w_j$. If a double deletion is even more severe ($w_{ij} < w_i w_j$), we call the epistasis “aggravating”; if it is less severe, we call it “buffering.” In both cases, we can conclude that the genes are functionally associated.

An example can help us understand why the “naive” expectation – that functionally unrelated genes have multiplicative effects – may be justified: The reproduction rate of an organism is proportional to both (i) the probability to reach the age of reproduction and (ii) the mean number of offspring when this age has been reached. If gene A affects only (i) and gene B affects only (ii), we would consider them functionally unrelated, and their effect on reproduction is in fact multiplicative.

To study epistasis between metabolic enzymes in the yeast *S. cerevisiae*, Segrè *et al.* [84] predicted growth rates by flux balance analysis. Based on predicted relative growth rates w_i and w_{ij} after single and double deletions, they computed an epistasis measure

$$\hat{\varepsilon}_{ij} = \frac{w_{ij} - w_i w_j}{|\hat{w}_{ij} - w_i w_j|} \quad (8.9)$$

for each pair of genes. The term \hat{w}_{ij} is defined as follows: In case of buffering interactions, it represents extreme buffering ($\hat{w}_{ij} = \min(w_i, w_j)$), where the less severe deletion does not play a role; in case of aggravating interactions, it represents extreme aggravation ($w_{ij} = 0$), where double mutations are lethal. The statistical distribution of $\hat{\varepsilon}_{ij}$, obtained from the FBA simulation, has a strong peak around $\hat{\varepsilon} = 0$, that is, the growth defects are approximately multiplicative in most cases. However, some gene pairs show strong aggravation (lethal phenotypes) or complete buffering.

To further analyze the epistasis values, genes were grouped into functional categories (see Figure 8.16). In fact, the epistatic effects were strongly related to functional groups, and, with few exceptions, the epistatic effects between two groups are either aggravating or buffering, but not both (monochromatic interactions). This result shows that genes contribute to biological fitness through their roles in functional subsystems. The same kind of analysis can be used to establish functional groups based on measured growth rates.

8.3.4 Evolution of Function and Modules

In engineering, modularity helps developers share their work and facilitates repair because stand-alone parts can more easily be replaced. Most machine parts have one specific function (even though exceptions exist – the wings of a plane are also used to store fuel). Even in cases where nonmodular designs would perform better, a modular design may still be preferable because it keeps machines or software understandable.

Does this also concern modules in biology? Biological systems are not designed, but shaped by mutation and selection, which constantly change them by small modifications. Existing structures can be modified, rewired, and reused for new purposes. This process resembles tinkering rather than engineering [85]: Starting from organisms that already function, gene recombination and mutations introduce innovations or reshape existing structures. Under selection pressures, these random changes can lead to functional adaptations. Natural selection is likely to choose solutions that work best, no matter whether

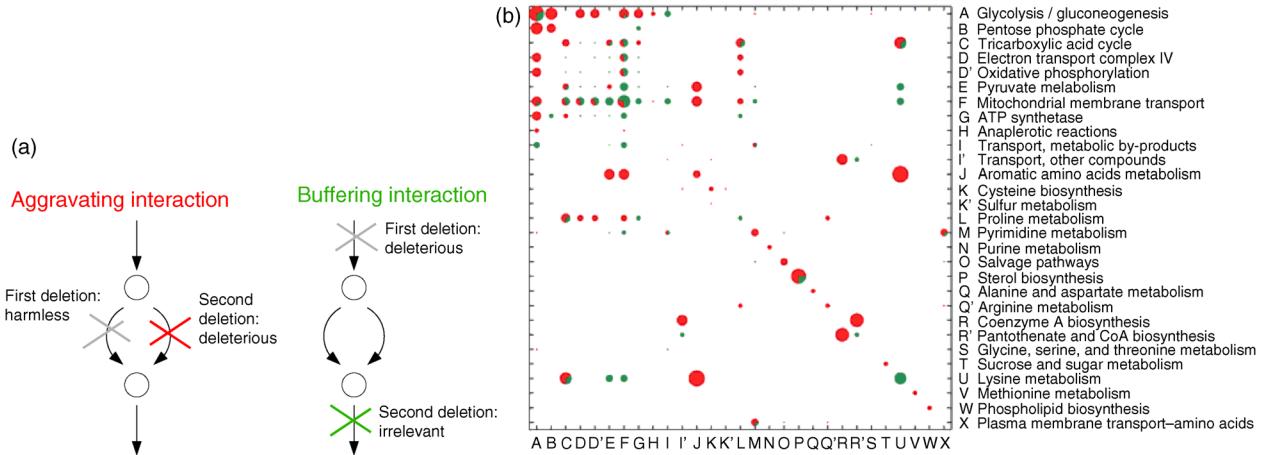


Figure 8.16 Epistatic interactions. (a) Schematic examples of buffering and aggravating epistasis. (b) Epistatic effects reflect functional subsystems. Circles show how many pairs of genes belonging to specific subsystems (rows and columns) are engaged in epistatic interactions. Circle radii represent numbers of epistatic interactions. Fractions of aggravating and buffering interactions are shown in red and green. (From Ref. [84].)

they are understandable to us, and the resulting biological systems may look very different from the solutions that engineers would conceive. The contrast between engineered networks (e.g., in computer chips) and evolved networks (e.g., transcriptional regulation networks) resembles the contrast between structured computer programs and artificial neural networks: A trained neural network may be able to solve complex computational problems of various kinds, but will not reflect the logical structure of these problems in an understandable way.

Evolution of Modularity

Biological systems like the transcription network in Figure 8.5 show modules and recurrent structures and seem to be specialized for particular functions. What could be the reasons for modularity to evolve? A possible explanation is that such modularity can contribute to robustness. The very existence of cells is a good example: Separated from each other by membranes, cells in a tissue can undergo apoptosis without affecting the functioning of neighboring cells. However, modularity also *relies on* robustness: To be used in various biochemical contexts (cell states or cell types), biological modules, for example, signaling pathways, need to be robust against typical variation in cells.

A second-order selection advantage ensues from the fact that modules can contribute to evolvability [86]. In a simulation study [87], Kashtan and Alon evolved hypothetical electronic circuits by random mutation and selection, realizing an optimal performance in two signal processing tasks. In an evolution scenario with one constant task, evolution leads to different, highly optimal circuits, which were nonmodular. In a second scenario,

the task consisted of two subtasks that appeared in varying combinations. Now modular circuits emerged in the evolution: Although being suboptimal for each of the tasks, they were better evolvable because small genetic changes were sufficient for switching from one task to the other. Also, in other computational evolution scenarios, varying modular goals helped speed up evolution toward well-adapted solutions [88].

Evolution of Analogous Traits

A second question, apart from the emergence of modules, is how recurrent structures such as the feed-forward loop can evolve. A possible explanation for this is analogy. Even though evolution is eventually based on random mutations, it is also strongly restricted by functional constraints on the phenotypes. Therefore, species under similar selection pressures may evolve similar traits. Wings, for instance, have independently evolved in birds and bats and show similar shapes due to their common functions and constraints (biomechanics, aerodynamics, and energy balance). Network motifs, which appeared many times independently, are another example of analogous evolution [87,89,90].

Convergence toward optimal performance may also explain the analogies between biological and technical systems: Although being physically very different, biological and technical systems share requirements such as robustness and cost-efficiency, so that optimization may lead to similar structures (in the case of chemical reaction systems versus electronic circuits, regulatory feedback; in the case of wings, the aerodynamic shapes). Technical metaphors can thus improve our understanding of biological systems in terms of function and physical constraints [89,91].

Example 8.2 Stabilization of Protein Levels by Negative Feedback

Protein production with negative feedback is described by a simple kinetic model:

$$\frac{ds}{dt} = \frac{1}{1 + s/K_1} a - bs, \quad (8.10)$$

with protein level s , maximal production rate a , inhibition constant K_1 , and degradation constant b . The kinetic model predicts that self-inhibition stabilizes the protein level against noise: Without inhibition ($K_1 \rightarrow \infty$), the Jacobian of the system reads $-b$; with inhibition, the value is even decreased to $-aK_1/(K_1 + s^{st})^2 - b$. As shown by the Lyapunov equation (7.19), this stabilizes the value of s under random perturbations of the system.

8.3.5 Independent Systems as a Tacit Model Assumption

A tacit assumption in pathway modeling is that networks surrounding a pathway of interest can be neglected. Ideally, experiments should be designed in such a way that this assumption is justified in later modeling.

The model (8.10), for example, makes predictions about an isolated, deterministic system. Details of transcription and translation, as well as chemical noise due to small particle numbers, are ignored; the dynamics of the protein level s depends only on s , while interactions with other processes are neglected; the model parameters are assumed to be constant. In reality, the parameters depend on the cell state, are noisy, or depend on s , which implies additional feedback loops.

Becskei and Serrano [92] have implemented this feedback loop as an engineered genetic circuit in living cells, and the experiment confirmed the predicted stabilizing effect. At first sight, this may not be very surprising – but this only means that we trust the tacit model assumptions. The main insight from the experiment is not that feedback can lead to stability (which is well known), but that the feedback system can be implemented in cells in such a way that the pathway is affected only little by the surrounding cell. This is a precondition for using such circuits as recombinable building blocks.

8.3.6 Modularity and Biological Function Are Conceptual Abstractions

Are modules in cells and organisms real, or just a construct we use for description? Are the biological functions we

attribute to proteins or organs well defined? When describing cells verbally, we cannot avoid using reifying terms such as “pathway,” “cell cycle phase,” or “function”; our language and thinking are based on distinct concepts, and so are our mathematical models. However, if we imagine how complex and flexible cell physiology actually is, we will admit that notions such as “module” or “biological function” are strong simplifications that enable, but may also limit, our understanding. For example, associating brain functions with specific parts of the brain may help understand physiology; yet, functional areas are flexible, common actions such as speech involve many areas simultaneously, and a functional brain relies on connections between areas as much as on the areas themselves. The same holds for cell physiology: Even if we distinguish biochemical processes in our descriptions, we still know that the processes are tightly coupled; that cell states fluctuate; that proteins can be involved in many processes (which we acknowledge when calling them “multifunctional”); and that their functions may keep on changing in evolution.

If we use modular models to describe cells, and if we focus our research on (apparently) modular systems, are we not bound to simply find what we are looking for? In fact, we cannot take for granted that evolution favors modular physiology. Cells function the way they do, and evolution selects for the most functional and evolvable phenotypes. However, if modular and specialized systems provide fitness advantages or contribute to evolvability (see Section 8.2), they may be favored in evolution. In this case, our notions of modules and function are “good to think with” not only because they are simple but also because they capture the evolutionary selection for things that work.

The same holds for our notion of “function”: To define a protein’s function, we may ask about its effects – how will the cell state or the cell fitness change when the component is over- or underexpressed? If a component has specific effects, and if these effects depend specifically on this component, the component will be under a selection pressure; thus, specialized components may be selected, which eventually justifies our notion of function.

That modularity is not just an invented concept is supported by several facts: First, modules can be independently defined based on genome analysis, high-throughput data, and cell physiology, and we observe clear correspondences between these modules. Second, if there was no modularity in biological systems, many of our methods would fail: Not only models, but also most experimental methods in molecular biology and biochemistry rely on the fact that genes, molecules, or cells can be treated as discrete, modular systems (see Section 6.4.2). Finally, synthetic biology, where proteins and their regulatory elements are expressed in new combinations and across different cell types, is an ongoing test of the modularity hypothesis [93].

Exercises

Section 8.1

- 1) *Adjacency matrix.* Determine the adjacency matrices for the graphs in Figure 8.2. Compute the degree and clustering coefficient of each node in parts (b) and (c). How many feed-forward loops are contained in the directed graph? Show that the topological distance in the directed graph is not a symmetric mathematical function.
- 2) *Degree distribution in metabolic networks.* According to Ref. [16], the degrees of metabolites, that is, the number of reactions in which each metabolite is involved, follow good approximation to a power law. In *E. coli* bacteria, the exponent is $\alpha \approx 2.2$. About 1% of the metabolites have a degree $k = 10$. Which percentage of metabolites have a degree $k = 20$?
- 3) *Self-inhibition in E. coli transcription network.* The transcription network of *E. coli* in Ref. [20] contains 424 transcription factors, connected by 519 edges. Among these transcription factors, 42 show self-inhibition. Is this number surprisingly large or would you expect it to appear by chance?
- 4) *Dynamic systems behind networks.* Explain the meaning of arrows in metabolic networks, transcription networks, and protein–protein interaction networks. How can the different types of arrows be confirmed or ruled out experimentally?

Section 8.2

- 5) *Network motifs.* What are network motifs? How would you determine them for a given network? Choose a network motif that appears in transcription networks, describes its dynamic properties, and speculate about its biological function. Explain why network motifs might emerge during evolution of biological networks.
- 6) *Sporulation in B. subtilis.* Consider the genetic network controlling sporulation in *B. subtilis* (Figure 8.14). Find all feed-forward loops in the scheme and determine their types. Assume a dynamic model of this system with piecewise linear kinetics, all thresholds set to values of 1/2, and all other model parameters set equal to 1. Sketch the time-dependent regulator concentrations after the sigma factor σ^E exceeds its threshold value. What qualitative behavior do you

expect for the four groups of target genes X_1 , X_2 , X_3 , and X_4 in terms of pulses and delays?

- 7) *Incoherent feed-forward loop type 1.* Draw and explain the dynamic response of an incoherent feed-forward loop type 1 to short and long input pulses. Discuss possible biological functions of this behavior.
- 8) *Simple cascade and feed-forward loop.* Consider three genes X, Y, Z, that activate each other in a cascade $X \rightarrow Y \rightarrow Z$. The temporal behavior of $x(t)$ is given and the levels of Y and Z are described by rate equations:

$$\begin{aligned}\frac{dy(t)}{dt} &= a_y \Theta(x(t) - x_0) - b_y y(t), \\ \frac{dz(t)}{dt} &= a_z \Theta(y(t) - y_0) - b_z z(t),\end{aligned}$$

with threshold values x_0 and y_0 . The step function Θ is defined by $\Theta(x \geq 0) = 1$, $\Theta(x < 0) = 0$. (a) Assume a constant $x < x_0$ and $y(0) > 0$, and draw $y(t)$. How do the parameters a_y and b_y affect the curve? (b) Assume a constant value $x > x_0$ and an initial value $y(0) = 0$, and draw $y(t)$. (c) Let $x(t)$ show a profile as in Figure 8.13, with a maximal value larger than x_0 , and $a_y = a_z = 0.1 \mu\text{M min}^{-1}$, $b_y = b_z = 0.1 \text{ min}^{-1}$, $x_0 = y_{\text{thr}}$, $b_z = 0.5 \mu\text{M}$. Draw $y(t)$ and $z(t)$ schematically and explain how their shapes arise from the dynamics. (d) In a feed-forward loop, Z is regulated by both X and Y. Synthesis of Z requires that both X and Y are above their threshold values

$$\frac{dz(t)}{dt} = a_z \Theta(x(t) - x_0) \cdot \Theta(y(t) - y_0) - b_z z(t).$$

Draw $z(t)$ schematically (after the onset of x) and discuss the influence of the parameters a_y , a_z , b_y , b_z , x_0 , and y_0 .

Section 8.3

- 9) *Living systems.* Discuss how notions of mechanism, dynamics, regulation, and optimality help us describe inanimate natural systems, living systems, and technical systems. Which similarities and differences do you see between these systems? Is there a need to describe them by different forms of mathematical models?

- 10) *Convergent evolution.* Find examples of homology and analogy in (i) shapes or organs of animals, (ii) biochemical processes and structures on the molecular level, and (iii) biological network structures.
- 11) *Epistasis.* Explain why epistasis between genes may indicate functional association. State the main assumptions on which your argument is based.
- 12) *Synthetic biology.* What insights can be gained from constructing a genetic circuit (e.g., a bistable switch) if similar systems exist already in wild-type cells?
- 13) *Modularity.* How would you define a modular system? Discuss possible criteria for modularity that apply to biological systems. Can modularity convey a selection advantage? How can modular systems emerge in evolution?

References

- 1 Henry, C.S., DeJongh, M., Best, A.A., Frybarger, P.M., Lindsay, B., and Stevens, R.L. (2010) High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nat. Biotechnol.*, 28, 977–982.
- 2 Feist, A.M., Henry, C.S., Reed, J.L., Krummenacker, M., Joyce, A.R., Karp, P.D., Broadbelt, L.J., Hatzimanikatis, V., and Palsson, B.Ø. (2007) A genome-scale metabolic reconstruction for *Escherichia coli* K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information. *Mol. Syst. Biol.*, 3, 121.
- 3 Thiele, I. et al. (2013) A community-driven global reconstruction of human metabolism. *Nat. Biotechnol.*, 31 (5), 419–427.
- 4 Jeong, H., Mason, S.P., Barabási, A.-L., and Oltvai, Z.N. (2001) Lethality and centrality in protein networks. *Nature*, 411, 41–42.
- 5 Suderman, M. and Hallett, M. (2007) Tools for visually exploring biological networks. *Bioinformatics*, 23 (20), 2651–2659.
- 6 Wagner, A. and Fell., D.A. (2001) The small world inside large metabolic networks. *Proc. Biol. Sci.*, 268 (1478), 1803–1810.
- 7 Watts, D.J. and Strogatz., S.H. (1998) Collective dynamics of ‘small-world’ networks. *Nature*, 393, 440–442.
- 8 Albert, R. and Barabási, A.-L. (2002) Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74, 47–97.
- 9 Itzkovitz, S. and Alon, U. (2005) Subgraphs and network motifs in geometric networks. *Phys. Rev. E*, 71, 026117.
- 10 Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002) Network motifs: simple building blocks of complex networks. *Supramol. Sci.*, 298, 824–827.
- 11 Barabási, A.-L. and Albert, R. (1999) Emergence of scaling in random networks. *Supramol. Sci.*, 286, 509.
- 12 Edwards, A.M. et al. (2007) Revisiting Lévy flight search patterns of wandering albatrosses, bumblebees and deer. *Nature*, 449, 1044–1048.
- 13 Pastor-Satorras, R., Smith, E., and Solé, R.V. (2003) Evolving protein interaction networks through gene duplication. *J. Theor. Biol.*, 222 (2), 199–210.
- 14 Wagner, A. (2003) How the global structure of protein interaction networks evolves. *Proc. Biol. Sci.*, 270 (1514), 457–466.
- 15 Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N., and Barabási, A.-L. (2002) Hierarchical organization of modularity in metabolic networks. *Supramol. Sci.*, 297, 1551–1555.
- 16 Jeong, H., Tombor, B., Albert, R., Oltvai, Z., and Barabási, A.-L. (2000) The large-scale organization of metabolic networks. *Nature*, 407, 651–654.
- 17 Arita, M. (2004) The metabolic world of *Escherichia coli* is not small. *Proc. Natl. Acad. Sci. USA*, 101 (6), 1543–1547.
- 18 Alon, U. (2007) Network motifs: theory and experimental approaches. *Nat. Rev. Genet.*, 8, 450–461.
- 19 Milo, R., Itzkovitz, S., Kashtan, N., Levitt, R., Shen-Orr, S., Ayzenstット, I., Sheffer, M., and Alon, U. (2004) Superfamilies of designed and evolved networks. *Supramol. Sci.*, 303, 1538–1542.
- 20 Shen-Orr, S., Milo, R., Mangan, S., and Alon, U. (2002) Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.*, 31, 64–68.
- 21 Lee, T.I., Rinaldi, N.J., Robert, F., Odom, D.T., Bar-Joseph, Z., Gerber, G.K., Hannett, N.M., Harbison, C.T., Thompson, C.M., Simon, I., Zeitlinger, J., Jennings, E.G., Murray, H.L., Gordon, D.B., Ren, B., Wyrick, J.J., Tagne, J.B., Volkert, T.L., Fraenkel, E., Gifford, D.K., and Young, R.A. (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Supramol. Sci.*, 298, 799–804.
- 22 Guimera, R. and Nunes Amaral, L.A. (2005) Functional cartography of complex metabolic networks. *Nature*, 433, 895.
- 23 Dekel, E. and Alon, U. (2005) Optimality and evolutionary tuning of the expression level of a protein. *Nature*, 436, 588–692.
- 24 Brockmann, D. and Helbing, D. (2013) The hidden geometry of complex, network-driven contagion phenomena. *Supramol. Sci.*, 342, 1337.
- 25 Cheong, R., Rhee, A., Wang, C.J., Nemenman, I., and Levchenko, A. (2011) Information transduction capacity of noisy biochemical signaling networks. *Science*, 334, 354.
- 26 Pearl, J. (1988) *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*, Morgan Kaufmann Publishers, Inc.
- 27 Rivoire, O. and Leibler, S. (2011) The value of information for populations in varying environments. *J. Stat. Phys.*, 142 (6), 1124–1166.
- 28 Robison, K., McGuire, A.M., and Church, G.M.A. (1998) A comprehensive library of DNA-binding site matrices for 55 proteins applied to the complete *Escherichia coli* K12 genome. *J. Mol. Biol.*, 284 (2), 241–254.
- 29 Lee, T.I., Rinaldi, N.J., Robert, F., Odom, D.T., Bar-Joseph, Z., Gerber, G.K., Hannett, N.M., Harbison, C.T., Thompson, C.M., Simon, I., Zeitlinger, J., Jennings, E.G., Murray, H.L., Gordon, D.B., Ren, B., Wyrick, J.J., Tagne, J.B., Volkert, T.L., Fraenkel, E., Gifford, D.K., and Young, R.A. (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, 298, 799–804.
- 30 Salgado, H., Gama-Castro, S., Martínez-Antonio, A., Díaz-Pereido, E., Sánchez-Solano, F., Peralta-Gil, M., García-Alonso, D., Jiménez-Jacinto, V., Santos-Zavaleta, A., Bonavides-Martínez, C., and Collado-Vides, J. (2004) RegulonDB (version 4.0): transcriptional

- regulation, operon organization and growth conditions in *Escherichia coli* K-12. *Nucleic Acids Res.*, 32, 303–306.
- 31 Keseler, I.M., Collado-Vides, J., Gama-Castro, S., Ingraham, J., Paley, S., Paulsen, I.T., Peralta-Gil, M., and Karp, P.D. (2005) EcoCyc: a comprehensive database resource for *Escherichia coli*. *Nucleic Acids Res.*, 39, D583–D590.
- 32 Shen-Orr, S., Milo, R., Mangan, S., and Alon, U. (2002) Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.*, 31, 64–68.
- 33 Kaplan, S., Bren, A., Zaslaver, A., Dekel, E., and Alon, U. (2008) Diverse two-dimensional input functions control bacterial sugar genes. *Mol. Cell*, 29, 786–792.
- 34 Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002) Network motifs: simple building blocks of complex networks. *Science*, 298, 824–827.
- 35 Mangan, S. and Alon, U. (2003) Structure and function of the feed-forward loop network motif. *Proc. Natl. Acad. Sci. USA*, 100, 11980–11985.
- 36 Mangan, S., Zaslaver, A., and Alon, U. (2003) The coherent feed-forward loop serves as a sign-sensitive delay element in transcription networks. *J. Mol. Biol.*, 334, 197–204.
- 37 Kalir, S., Mangan, S., and Alon, U. (2005) A coherent feed-forward loop with a sum input function prolongs flagella expression in *Escherichia coli*. *Mol. Syst. Biol.*, 1. doi: 10.1038/msb4100010
- 38 Itzkovitz, S., Levitt, R., Kashtan, N., Milo, R., Itzkovitz, M., and Alon, U. (2005) Coarse-graining and self-dissimilarity of complex networks. *Phys. Rev. E Stat. Nonlin. Soft. Matter. Phys.*, 71 (1 Part 2), 016127.
- 39 Savageau, M.A. (1998) Demand theory of gene regulation. *Genetics*, 149, 1665–1691.
- 40 Sasson, V., Shachrai, I., Bren, A., Dekel, E., and Alon, U. (2012) Mode of regulation and the insulation of bacterial gene expression. *Mol. Cell*, 46, 399–407.
- 41 Tyson, J.J., Chen, K.C., and Novak, B. (2003) Sniffers, buzzers, toggles and blinks: dynamics of regulatory and signaling pathways in the cell. *Curr. Opin. Cell Biol.*, 15 (2), 221–231.
- 42 Elowitz, M.B. and Leibler, S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature*, 403, 335–338.
- 43 Becskei, A. and Serrano, L. (2000) Engineering stability in gene networks by autoregulation. *Nature*, 405, 590–592.
- 44 Hasty, J., McMillen, D., and Collins, J.J. (2002) Engineered gene circuits. *Nature*, 420, 224–230.
- 45 Elowitz, M. *et al.* (2002) Stochastic gene expression in a single cell. *Science*, 297, 1183.
- 46 Pedraza, J.M. and van Oudenaarden, A. (2005) Noise propagation in gene networks. *Science*, 307, 1965–1969.
- 47 Alon, U. (2006) *An Introduction to Systems Biology: Design Principles of Biological Circuits*, CRC Mathematical & Computational Biology, Chapman & Hall.
- 48 Alon, U. (2007) Network motifs: theory and experimental approaches. *Nat. Rev. Genet.*, 8, 450–461.
- 49 Mangan, S. and Alon, U. (2003) Structure and function of the feed-forward loop network motif. *Proc. Natl. Acad. Sci. USA*, 100 (21), 11980–11985.
- 50 Prill, R.J., Iglesias, P.A., and Levchenko, A. (2005) Dynamic properties of network motifs contribute to biological network organization. *PLoS Biol.*, 3 (11), 1881–1892.
- 51 Klemm, K. and Bornholdt, S. (2005) Topology of biological networks and reliability of information processing. *Proc. Natl. Acad. Sci. USA*, 102 (51), 18414–18419.
- 52 Hart, Y., Antebi, Y.E., Mayo, A.E., Friedman, N., and Alon, U. (2012) Design principles of cell circuits with paradoxical components. *Proc. Natl. Acad. Sci. USA*, 109 (21), 8346–8351.
- 53 Goldbeter, A. and Koshland, D.E., Jr (1981) An amplified sensitivity arising from covalent modification in biological systems. *Proc. Natl. Acad. Sci. USA*, 78, 6840–6844.
- 54 Goldbeter, A. and Koshland, D.E., Jr (1984) Ultrasensitivity in biological systems controlled by covalent modification. *J. Biol. Chem.*, 259, 14441–14447.
- 55 Kochanowski, K., Volkmer, B., Gerosa, L., Havercorn van Rijsewijk, B.R., Schmidt, A., and Heinemann, M. (2013) Functioning of a metabolic flux sensor in *Escherichia coli*. *Proc. Natl. Acad. Sci. USA*, 110 (3), 1130–1135.
- 56 Huiskes, R. *et al.* (2000) Effects of mechanical forces on maintenance and adaptation of form in trabecular bone. *Nature*, 405, 704–706.
- 57 Frost, H.M. (2001) From Wolff's law to the Utah paradigm: insights about bone physiology and its clinical applications. *Anat. Rec.*, 262, 398–419.
- 58 Nowlan, N.C., Murphy, P., and Prendergast, P.J. (2007) Mechano-biology of embryonic limb development. *Ann. N.Y. Acad. Sci.*, 1101, 389–411.
- 59 Weinkamer, R. and Fratzl, P. (2011) Mechanical adaptation of biological materials: the examples of bone and wood. *Mater. Sci. Eng.*, 31, 1164–1173.
- 60 Rosenfeld, N., Elowitz, M.B., and Alon, U. (2002) Negative auto-regulation speeds the response times of transcription networks. *J. Mol. Biol.*, 323 (5), 785–793.
- 61 He, F., Fromion, V., and Westerhoff, H.V. (2013) (Im)Perfect robustness and adaptation of metabolic networks subject to metabolic and gene-expression regulation: marrying control engineering with metabolic control analysis. *BMC Syst. Biol.*, 7, 131.
- 62 Klevecz, R.R., Bolen, J., Forrest, G., and Murray, D.B. (2004) A genomewide oscillation in transcription gates DNA replication and cell cycle. *Proc. Natl. Acad. Sci. USA*, 101 (5), 1200–1205.
- 63 Machné, R. and Murray, D. (2012) The yin and yang of yeast transcription: elements of a global feedback system between metabolism and chromatin. *PLoS One*, 7 (6), e37906.
- 64 Mangan, S., Zaslaver, A., and Alon, U. (2006) The incoherent feed-forward loop accelerates the response time of the GAL system in *E. coli*. *J. Mol. Biol.*, 356, 1073–1082.
- 65 Kaplan, S., Bren, A., Erez Dekel, E., and Alon, U. (2008) The incoherent feed-forward loop can generate non-monotonic input functions for genes. *Mol. Syst. Biol.*, 4 (203), 203.
- 66 Eichenberger, P., Fujita, M., Jensen, S.T., Conlon, E.M., Rudner, D.Z., Wang, S.T., Ferguson, C., Haga, K., Sato, T., Liu, J.S., and Losick, R. (2004) The program of gene transcription for a single differentiating cell type during sporulation in *Bacillus subtilis*. *PLoS Biol.*, 2 (10), e328.
- 67 Hartwell, L.H., Hopfield, J.J., Leibler, S., and Murray, A.W. (1999) From molecular to modular cell biology. *Nature*, 402 (6761 Suppl.), C47–C52.
- 68 Mewes, H.W., Dietmann, S., Frishman, D., Gregory, R., Mannhaupt, G., Mayer, K.F.X., Münsterkötter, M., Ruepp, A., Spannagl, M., Stümpflen, V., and Rattei, T. (2008) MIPS: analysis and annotation of genome information in 2007. *Nucleic Acids Res.*, 36, D196–D201.
- 69 Kanehisa, M., Goto, S., Kawashima, S., S., and Nakaya, A. (2002) The KEGG databases at GenomeNet. *Nucleic Acids Res.*, 30, 42–46.
- 70 The Gene Ontology Consortium (2000) Gene ontology: tool for the unification of biology. *Nat. Genet.*, 25, 25–29.
- 71 Liebermeister, W., Noor, E., Flamholz, A., Davidi, D., Bernhardt, J., and Milo, R. (2014) Visual account of protein investment in cellular functions. *Proc. Natl. Acad. Sci. USA*, 111 (23), 8488–8493.

- 72** Arike, L., Valgepea, K., Peil, L., Nahku, R., Adamberg, K., and Vilu, R. (2012) Comparison and applications of label-free absolute proteome quantification methods on *Escherichia coli*. *J. Proteomics*, 75 (17), 5437–5448.
- 73** Schuster, S., Pfeiffer, T., Moldenhauer, F., Koch, I., and Dandekar, T. (2002) Exploring the pathway structure of metabolism: decomposition into subnetworks and application to *Mycoplasma pneumoniae*. *Bioinformatics*, 18 (2), 351–361.
- 74** Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N., and Barabási, A.-L. (2002) Hierarchical organization of modularity in metabolic networks. *Supramol. Sci.*, 297, 1551–1555.
- 75** Shen-Orr, S., Milo, R., Mangan, S., and Alon, U. (2002) Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.*, 31, 64–68.
- 76** Guimera, R. and Nunes Amaral, L.A. (2005) Functional cartography of complex metabolic networks. *Nature*, 433, 895.
- 77** Ederer, M., Sauter, T., Bullinger, E., Gilles, E., and Allgöwer, F. (2003) An approach for dividing models of biological reaction networks into functional units. *Simulation*, 79 (12), 703–716.
- 78** Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., and Friedman, N. (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet.*, 34 (2), 166–176.
- 79** Tanay, A., Sharan, R., and Shamir, R. (2002) Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18 (Suppl. 1), S136–S144.
- 80** Pellegrini, M., Marcotte, E.M., Thompson, M.J., Eisenberg, D., and Yeates, T.O. (1999) Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Natl. Acad. Sci. USA*, 96 (8), 4285–4288.
- 81** Schilling, C.H. and Palsson, B.Ø. (1998) The underlying pathway structure of biochemical reaction networks. *Proc. Natl. Acad. Sci. USA*, 95 (8), 4193–4198.
- 82** Schuster, S., Fell, D., and Dandekar, T. (2000) A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nat. Biotechnol.*, 18, 326–332.
- 83** Jol, S.J., Kümmel, A., Terzer, M., Stelling, Jörg., and Heinemann, M. (2012) System-level insights into yeast metabolism by thermodynamic analysis of elementary flux modes. *PLoS Comput. Biol.*, 8 (3), e1002415.
- 84** Segrè, D., DeLuna, A., Church, G.M., and Kishony, R. (2005) Modular epistasis in yeast metabolism. *Nat. Genet.*, 37, 77–83.
- 85** Alon, U. (2003) Biological networks: the tinkerer as an engineer. *Supramol. Sci.*, 301, 1866–1867.
- 86** Kirschner, M. and Gerhart, J. (1998) Evolvability. *Proc. Natl. Acad. Sci. USA*, 95 (15), 8420–8427.
- 87** Kashtan, N. and Alon, U. (2005) Spontaneous evolution of modularity and network motifs. *Proc. Natl. Acad. Sci. USA*, 102 (39), 13773–13778.
- 88** Kashtan, N., Noor, E., and Alon, U. (2007) Varying environments can speed up evolution. *Proc. Natl. Acad. Sci. USA*, 104 (34), 13711–13716.
- 89** Csete, M.E. and Doyle, J.C. (2002) Reverse engineering of biological complexity. *Supramol. Sci.*, 295 (5560), 1664–1669.
- 90** Conant, G.C. and Wagner, A. (2003) Convergent evolution of gene circuits. *Nat. Genet.*, 34, 264–266.
- 91** Lazebnik, Y. (2002) Can a biologist fix a radio? or, what I learned while studying apoptosis. *Cancer Cell*, 2, 179–182.
- 92** Becskei, A. and Serrano, L. (2000) Engineering stability in gene networks by autoregulation. *Nature*, 405, 590–592.
- 93** Hasty, J., McMillen, D., and Collins, J.J. (2002) Engineered gene circuits. *Nature*, 420, 224–230.

Further Reading

- Network motifs:** Alon, U. (2006) *An Introduction to Systems Biology: Design Principles of Biological Circuits*, CRC Mathematical & Computational Biology, Chapman & Hall.
- Network motifs:** Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. (2002) Network motifs: simple building blocks of complex networks. *Supramol. Sci.*, 298, 824–827.
- Networks:** Barabási, A.-L. (2002) *Linked: The New Science of Networks*, Perseus Publishing.
- Networks:** Barabási, A.-L. and Oltvai, Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, 5, 101–113.
- Repressilator:** Elowitz, M.B. and Leibler, S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature*, 403, 335–338.
- Stochastic gene expression:** Elowitz, M. et al. (2002) Stochastic gene expression in a single cell. *Supramol. Sci.*, 297, 1183.
- Metabolic network reconstruction:** Henry, C.S., DeJongh, M., Best, A.A., Frybarger, P.M., Lindsay, B., and Stevens, R.L. (2010) High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nat. Biotechnol.*, 28, 977–982.
- Signaling networks:** Atlas of Cancer Signalling Networks (acsn.curie.fr/).
- Human metabolic network:** Thiele, I. et al. (2013) A community-driven global reconstruction of human metabolism. *Nat. Biotechnol.*, 31 (5), 419–427.
- Regulation motifs:** Tyson, J.J., Chen, K.C., and Novak, B. (2003) Sniffers, buzzers, toggles and blinks: dynamics of regulatory and signaling pathways in the cell. *Curr. Opin. Cell Biol.*, 15 (2), 221–231.
- Interactome networks:** Vidal, M. et al. (2011) Interactome networks and human disease. *Cell*, 144 (6), 986–998.

Gene Expression Models

9.1 Mechanisms of Gene Expression Regulation

Summary

The expression of genes is a highly regulated process in eukaryotic as well as in prokaryotic cells and has a profound impact on the ability of the cells to maintain vitality, perform cell division, and respond to environmental changes or stimuli. In this section, we describe two basic mechanisms of gene expression regulation: the transcriptional regulation through transcription factors (TFs) that bind to DNA motifs upstream of the transcription start site and thus initiate transcription of the DNA sequence to mRNA, and the posttranscriptional regulation through microRNAs (miRNAs) that bind to the mRNA sequences and act as translational repressors. These two mechanisms of gene expression regulation have been intensively studied in recent years giving rise to many different computational methods, data resources, and databases for the analysis of specific gene regulatory pathways.

9.1.1 Transcription Factor-Initiated Gene Regulation

Gene expression is a fundamental process that involves many different molecular processes from the activation of transcriptional regulators to the synthesis of a functional protein [1,2]. Hundreds of different cell types exist and fulfill specific roles in the organism. Each cell type contains theoretically information on the same set of genes; however, only a proportion of them is expressed determining the specific role of the cells of this type. Gene expression in eukaryotes is controlled at six different steps (see Chapter 13). These steps determine the diversity and specification of the organism [3]:

9.1 Mechanisms of Gene Expression Regulation

- Transcription Factor-Initiated Gene Regulation
- General Promoter Structure
- Prediction and Analysis of Promoter Elements
- Posttranscriptional Regulation through microRNAs

9.2 Dynamic Models of Gene Regulation

- A Basic Model of Gene Expression and Regulation
- Natural and Synthetic Gene Regulatory Networks
- Gene Expression Modeling with Stochastic Equations

9.3 Gene Regulation Functions

- The Lac Operon in *E. coli*
- Gene Regulation Functions Derived from Equilibrium Binding
- Thermodynamic Models of Promoter Occupancy
- Gene Regulation Function of the Lac Promoter
- Inferring Transcription Factor Activities from Transcription Data
- Network Component Analysis
- Correspondences between mRNA and Protein Levels

9.4 Fluctuations in Gene Expression

- Stochastic Model of Transcription and Translation
- Intrinsic and Extrinsic Variability
- Temporal Fluctuations in Gene Cascades

Exercises

References

Further Reading

- 1) *Transcriptional control*: when and how often is a gene transcribed.
- 2) *RNA processing control*: how is the RNA transcript spliced.
- 3) *RNA transport and localization control*: which mRNAs in the nucleus are exported to cytosol and where in the cytosol are they localized.
- 4) *Translational control*: which mRNAs in the cytosol are translated by ribosomes.

- 5) *mRNA degradation control*: which mRNAs in the cytosol are destabilized.
- 6) *Protein activity control*: decide upon activation, inactivation, compartmentalization, and degradation of the translated protein.

Each single step is complex and has been studied extensively in isolation. In computational analyses, the entire gene regulation process is typically approximated with a linear structure of more or less independent modules where the output of the previous module is the input for the current module.

The expression level of the majority of genes is controlled by transcription factors. Transcription factors are proteins that bind to DNA regulatory sequences upstream of the site at which transcription is initiated. Various regulatory pathways control their activities. More than 5% of human genes encode transcription factors [4]. Once activated, transcription factors bind to gene regulatory elements and, through interactions with other components of the transcription machinery, promote access to DNA and facilitate the recruitment of the RNA polymerase enzymes to the transcriptional start site.

In eukaryotes, there are three RNA polymerases, namely, RNAP I, II, and III. RNAP II catalyzes the transcription of protein-coding genes and is responsible for the synthesis of mRNAs and certain small nuclear RNAs, while the others are responsible for generating tRNAs (RNAP III) and ribosomal RNAs (RNAP I) [5].

The RNAP II enzyme itself is unable to initiate promoter-dependent transcription in the absence of complementing factors. It needs to be supplemented by so-called *general transcription factors* (GTFs) [1]. RNAP II together with these GTFs and the DNA template forms the preinitiation complex and the assembly of this complex is nucleated by binding of TBP (a component of TFIID) to the *TATA-box* [6]. The TATA-box is a core promoter (or minimal promoter) that directs transcriptional initiation at a short distance (about 30 bp downstream). Soon after RNAP II initiates transcription, the nascent RNA is modified by the addition of a *cap* structure at its 5' end. This cap serves initially to protect the new transcript from attack by nucleases and later serves as a binding site for proteins involved in export of the mature mRNA into the cytoplasm and its translation into protein.

The start of RNA synthesis catalyzed by RNAP II is the *transcription initiation*. During *transcription elongation*, the polymerase moves along the gene sequence from the 5' to the 3' end and extends the transcript. The transition between these early transcriptional events, initiation and elongation, seems to be coordinated by the capping process. A family of elongation factors then regulates the elongation phase. Upon reaching the end of a gene,

RNAP II stops transcription (*termination*), the newly RNA is cleaved (*cleavage*), and a polyadenosine (*poly(A)*) tail is added to the 3' end of the transcript (*polyadenylation*).

The resulting pre-mRNA contains coding sequences in the gene (*exons*) that are divided by long noncoding sequences (*introns*). These introns are removed by pre-mRNA splicing.

Transcription, that is, the transfer of information from DNA to RNA, and translation, that is, the transfer of information from RNA to protein, are spatially separated in eukaryotes by the nuclear membrane; transcription occurs in the nucleus, whereas translation is a cytoplasmic event. For that reason, processed mRNAs must be transported from the nucleus to the cytoplasm before translation can occur. The bidirectional transport of macromolecules between nucleus and cytoplasm occurs through protein-covered pores in the nuclear membrane. The export of mRNA is mediated by factors that interact with proteins of the nuclear pores and bind to mRNA molecules in the nucleus and direct them into the cytoplasm. Translation of mRNA into protein takes place on *ribosomes*, that is, large ribonucleoprotein complexes, and follows the similar principles as transcription. Important for the translation process is the presence of transfer RNA molecules (tRNAs) that deliver the correct amino acid to the currently considered nucleotide triplet. tRNAs have a common characteristic secondary structure and are bound to the mRNA by means of anticodons complementary to the triplet for which they carry the appropriate amino acid. Subsequently, tRNAs are recruited and the polypeptide is synthesized until the first stop codon is present. The first step is the location of the start codon in conjunction with subunits of the ribosome triggered by translational initiation factors. Subsequent phases are elongation and termination. The nascent polypeptide chain then undergoes folding and often posttranslational chemical modification to generate the final active protein (see Chapter 13).

The transcriptional control is the most important in gene expression, which makes biological sense since the cell invests energy to synthesize products and this energy should not be wasted through subsequent termination of the activity of these products. Gene transcription is controlled by RNAP II and it depends on the presence of several additional proteins in order to transcribe the gene in the proper cellular context. In eukaryotes, gene expression requires a complex regulatory region that defines the transcription starting point and controls the initiation of transcription, the *promoter*. Several algorithms are available that try to identify promoters for specific genes. Some of these algorithms are discussed in this section.

9.1.2

General Promoter Structure

Promoter prediction algorithms implicitly assume a specific model for a typical promoter. The general structure of a RNAP II promoter is described in Figure 9.1a. The typical promoter is composed of three levels of regulatory sequence signals: The first level contains sequence motifs that enable the binding of specific transcription factors. The next level is the combination of binding sites to promoter modules that jointly act as functional units. The third level consists of the complete promoter that modulates gene transcription depending on cell type, tissue type, developmental stage, or activation by signaling pathways.

The promoter must contain binding sites for the GTFs such as the TATA-box. These proximate regulatory motifs constitute the *core promoter* that is able to bind the preinitiation complex and to determine the exact transcription start site. The core promoter needs additional regulatory motifs at varying distances from the transcriptional start point, the regulatory binding sites (transcription factor binding sites, TFBSs). These sites can be situated nearby or kilobases away from the core promoter.

Transcription initiation can be viewed as a process involving successive formation of protein complexes. In the first step, transcription factors bind to upstream promoter and enhancer sequence motifs and form a multi-protein complex. In the next step, this complex recruits the RNAP II/GTF complex to the core promoter and the

transcription start site. This is done through protein–protein interactions either directly or by adaptor proteins [7]. The full complex then starts the transcription process.

The *core promoter* is located in the direct neighborhood of the transcription start site (approximately 30 bp). The core promoter is the best-characterized part of the promoter and is defined as a set of binding sites sufficient for the assembly of the RNAP II/GTF complex and for specifying transcriptional initiation. Several core classes of promoters are known [8].

- 1) TATA-box: If TBP is present in the RNAP II/GTF complex, then this protein binds to the sequence motif and the transcription starts approximately 30 bp downstream.
- 2) TATA-less: No TATA-box is present. The start site is determined by a sequence motif INR (initiator region) surrounding the start site [9].
- 3) A combination of both INR and TATA-box.
- 4) Null promoter: None of the two sequence motifs is present. Transcription initiation is solely based on upstream (or downstream) promoter elements [10].
- 5) In addition to INR, in some cases there exist a downstream promoter element (DPE) and both elements are able to specify the transcription start site [11].

Whereas the core promoter determines the transcription start site, this function cannot explain how genes whose protein products are needed in parallel are coregulated, for example, from genes that are located on different chromosomes. Thus, additional regulatory elements

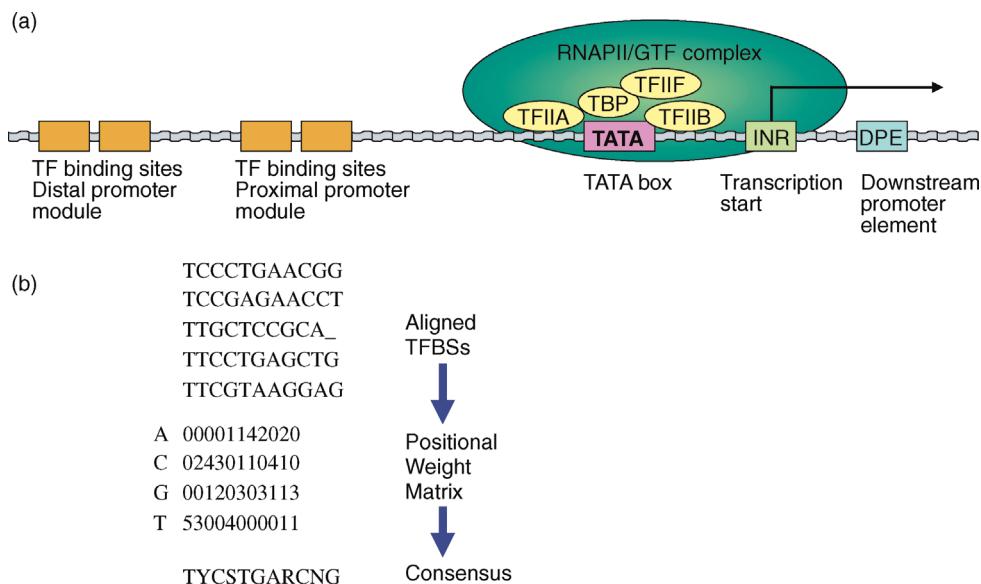


Figure 9.1 (a) General structure of a eukaryotic gene promoter. (b) Example of a positional weight matrix and a consensus sequence derived from different transcription factor binding sites.

are necessary that meet the requirement of higher flexibility and coordinated gene expression.

Typically, a few hundred bp upstream of the core promoter is the proximate promoter module that contains transcription factor binding sites for proteins responsible for the modulation of the transcription. The corresponding factors can influence either the binding of the core promoter components or the chromatin structure (or both). Furthermore, a promoter can contain a distal promoter module (on the order of kb apart from the transcription start site). Although these modules cannot act as promoters on their own, they are able to enhance or suppress the activity of transcription up to orders of magnitude (enhancer or silencer). Enhancer and silencer often exhibit a tissue-specific activity. Likewise to the transcription factors binding to the proximate module of the promoter, the factors binding to the distal module influence gene expression by interactions with the factors in the RNAP II/GTF complex or by changing the chromatin structure. There is no clear boundary for the promoter in the 5'-direction and the common explanation of interactions with distal factors to the transcription apparatus is given by the formation of large loops in the DNA. The function of a promoter is to increase or repress the transcription from the core promoter (basal transcription). Thus, any given gene will have a specific regulatory region determined by the binding sites of the transcription factors that ensure that the gene is transcribed in the appropriate cell type and at the proper point in development. The transcriptional activation is not simply determined by the presence of the binding sites but also through the availability of the corresponding transcription factors. These transcription factors are themselves subjected to regulation and activation, for example, through signaling pathways and the whole process can entail complex procedures such as transcriptional cascades and feedback control loops [12].

9.1.3 Prediction and Analysis of Promoter Elements

As described in the previous section, promoters are complex and diverse, which makes promoter prediction a difficult task. Several reviews have been published that compare the performances of promoter recognition programs [13–15].

9.1.3.1 Sequence-Based Analysis

The modeling of gene transcription regulation follows its combinatorial nature starting from the detection of individual binding sites (5–25 bp in length), to the detection

of specific combinations of binding sites, so-called composite regulatory elements [16], to the detection of the promoter.

The detection of *individual binding sites* is the first level in that process. TFBSs have high sequence variability that distinguishes them, for example, from restriction sites, that is, the recognition sequences of a restriction enzyme. Whereas restriction sites are almost exact in the sense that sites that vary by only a single mismatch will be cut less well by orders of magnitude, transcription factor binding can tolerate high sequence variability of the TFBSs. This variation makes biological sense in that it allows a higher flexibility of the regulatory system and assigns the promoters different activity levels.

In order to meet this flexibility, known TFBSs for the same transcription factor that may vary slightly are often represented by a *consensus sequence* that is close to each single motif according to some criterion. There is a trade-off in the consensus sequences between the number of mismatches that are allowed and the precision of the representation and thus a trade-off between the specificity and the sensitivity of the algorithms. A consensus sequence is typically denoted in the *IUPAC* code to describe ambiguities in nucleotide composition (Figure 9.1b).

Alternatively to consensus sequences is the use of *positional weight matrices* (PWM). A PWM is a matrix representation of a TFBS, with rows representing one of the bases, A, C, G, and T, and columns representing the position within the motif (Figure 9.1b). Each entry in the matrix corresponds to a numerical value indicating the confidence for the specific base at that position. The PWM approach is somewhat more general than the consensus sequence approach in the sense that each consensus can be represented by a PWM (e.g., through frequency counts across the aligned motifs) such that the same set of sites can be matched but not vice versa. The calculation of the matrix elements can be performed in different ways. Stormo [14] applied a neural network learning algorithm to determine the weights of a PWM to distinguish known sites from nonsites in a training sample of *Escherichia coli* sequences. Afterward, he predicted new sequences using the calculated weights. Other approaches used thermodynamical considerations to compute the weights of a PWM [8]. The authors showed that the logarithms of the base frequencies should be proportional to the binding energy contribution of the bases assuming an equal distribution of base pairs through the genome.

Recognition of composite regulatory elements has been proposed in order to meet the combinatorial nature of gene regulation, for example, of two transcription factors that interact with each other in gene regulation.

Statistical approaches have been made to reveal common pairs from DNA sequences by weighing matrices for two corresponding transcription factors. Methods take into account, for example, the matching distances of the matrices on the DNA sequence and the mutual orientation and combine this with binding energy considerations. A number of examples of composite regulatory elements have been collected in the TRANSCompel database [17].

The general principle of *promoter recognition* methods is based on the strategy to determine a promoter model by features that are trained on a set of known promoter sequences and nonpromoter sequences. These features are subsequently used to search for an unknown number of promoters in a contiguous DNA sequence. The methods distinguish each other by the way the features are determined. Typically, they fall into two groups. The first group uses the pure sequence composition and is based on scoring moving sequence windows, whereas the second group employs prediction based on the detection of motifs from the core promoter element such as TATA-box or INR.

The first group of algorithms can be exemplified by the PromFind method described in Ref. [18]. This method is based on the idea of discriminative counts of sequence groups. PromFind uses the frequency of heptamers in coding and noncoding sequences trained on sequences of 300 bp in length. Discrimination is based on the following measure:

$$d_i(s) = \frac{f(s)}{f(s) + f_i(s)}, \quad i = 1, 2. \quad (9.1)$$

Here, $f(s)$ denotes the frequency of heptamer s in the promoter sequences and $f_i(s)$ corresponds to the frequency of the heptamer in the training sample ($i=1$: noncoding; $i=2$: coding). For each sequence in a window of size 300 bp, the two measures are calculated and the window with the best score is returned. Another way of computing discriminative counts is employed in PromoterInspector [19].

The second group of algorithms uses biological sequence features from the core promoter [20]. The hit ratio of known TFBSs within promoters and non-promoters is used as an indicator for the identification of a promoter. The combined ratio scores of all TFBSs in a certain sequence window are used to build a scoring profile. This profile combined with a weight matrix for TATA-boxes is used for predicting the transcription start site. Further methods model the core promoter with artificial neural networks [21], ensembles of multilayer perceptrons for binding sites, or hidden Markov models.

9.1.3.2 Approaches that Incorporate Additional Information

Since it has been shown that the error rates of the promoter prediction programs are fairly unsatisfactory [13], new developments try to incorporate additional information as a backup when predicting TFBSs.

A first class of approaches *combines binding site prediction with gene expression data* derived from DNA arrays. The widespread use of DNA arrays has given rise to the following general program [22]:

- 1) Identify coexpression groups by clustering or other statistical methods.
- 2) Search in the upstream regions of the grouped genes for common regulatory motifs.

This approach has been utilized for the first time for identifying novel regulatory networks in *Saccharomyces cerevisiae*. The authors used a K -means clustering algorithm to identify groups of coregulated genes. They identified common sequence motifs in the upstream sequences of the genes and identified 18 motifs in 12 clusters that were highly overrepresented within their own cluster and absent in the others, thus indicating the existence of different regulation patterns.

This and other studies demonstrated that genes that are coexpressed across multiple experimental conditions underlie often common regulatory mechanisms, and thus share common TFBSs in their promoters. Although these results are promising, methods that work well in yeast are difficult to extend to higher eukaryotes. This is mainly due to the fact that in yeast regulatory sequences are fairly proximal to the transcription start site whereas in higher eukaryotes these sequences can be located many kilobases on either side of the coding region. A recent approach to human data has been published [23]. Here, the authors used DNA array data and human genome sequence data to identify putative regulatory elements that control the transcriptional program of the human cell cycle. They identified several transcription factors (such as E2F, NF-Y, and CREB) whose regulatory sequences were enriched in cell cycle-regulated genes and assigned these factors to certain phases of the cell cycle.

A second class of approaches uses *comparative sequence analysis* from upstream sequences of orthologous genes through different organisms [24]. These authors investigated skeletal muscle-specific transcription factors and found that their binding sites are highly conserved in human and mouse DNA sequences. The general observation of conserved noncoding regions throughout different organisms has given rise to a number of recent developments that incorporate cross-species analysis of promoter elements [25].

A combination of the two approaches has been applied to the detection and experimental verification of a novel *cis*-regulatory element involved in the heat shock response in *Caenorhabditis elegans* [26]. The authors identified coregulated genes with DNA arrays and investigated the upstream regions of these genes for putative binding sites by pattern recognition algorithms. Either in the case of significant overrepresentation or in the case of cross-species conservation, they build biological assays of the regulatory motifs using GFP reporter transgenes.

Additional sequence information is also sometimes incorporated in promoter identification, in particular the identification of CpG islands. It has been reported that these CpG islands correlate with promoters in vertebrates so that their features are used in the computational process. By definition, CpG islands are genomic regions with the following characteristics:

- 1) longer than 200 bp;
- 2) nucleotide frequencies of C and G in that region greater than 50%;
- 3) CpG dinucleotide frequency in that region higher than 0.6 of that expected from mononucleotide frequencies.

Despite all these developments, the recognition and identification of promoter elements remains an error-prone task implied by the highly complex nature of eukaryotic gene regulation. Future approaches will thus have to incorporate additional information to a much larger amount than it is currently done.

9.1.4 Posttranscriptional Regulation through microRNAs

MicroRNAs are small RNA molecules that regulate gene expression at the posttranslational level. MicroRNAs are about 20–25 nucleotides (nt) long and have been identified in plants, animals, and viruses [27]. A microRNA binds specifically at an mRNA and controls gene expression through the regulation of mRNA stability and translation. The general mechanism of microRNA gene regulation is shown in Figure 9.2a. Most miRNAs are transcribed from their DNA sequences as primary miRNAs (pri-miRNAs) by RNA polymerase II. In animals, pri-miRNAs are converted to mature miRNAs by two successive endonucleolytic cleavages. The pri-miRNA is first cut in the nucleus by the ribonuclease III (RNase III) enzyme Drosha into an approximately 70 nt long stem loop, the precursor miRNA (pre-miRNA). This precursor miRNA is exported to cytoplasm by Exportin-5 and cut into mature miRNA by another RNase III enzyme called

Dicer. The mature miRNA is then loaded into an effector complex, the RNA-induced silencing complex (RISC), whose core component is a member of the Argonaute (AGO) family of RNA regulatory proteins [28]. miRNAs match with their mRNA targets by sequence complementarity. In plants, this match is very well positioned in either the coding or the 3' UTR regions of the targets. This nearly perfect binding initiates mRNA degradation. In animals, miRNAs typically regulate gene expression by imperfect binding to the 3' UTRs of the target mRNAs effecting inhibition of protein synthesis or causing mRNA degradation (Figure 9.2a). The 5'-region (seed region) of the miRNA (approximately nucleotides 2–8) is the primary determinant of binding specificity. The effect of the rather imperfect binding of the rest of the miRNA sequence enables regulation of hundreds of different mRNAs per miRNA, thus providing each miRNA with a powerful regulatory potential. A common consequence of such seed-mediated miRNA binding is a decrease in the amount of the protein encoded by the target mRNA. However, the precise molecular mechanisms of miRNA-mediated translational repression are still under discussion. In fact, distinct mechanisms of repression have been proposed by different laboratories for different miRNA–target pairs and even for the same miRNAs.

For most of the detected miRNAs, their functional roles are still unknown. miRNA target discovery is a major issue in that respect. However, for a few cases, it has already been shown that deregulation of miRNA expression has multiple implications for developmental processes and human diseases. For example, it had been shown that deregulation of miRNAs can affect cancer pathways through an interaction with genes that are involved in proliferation and apoptosis [29]. The authors showed that two miRNAs, miR-17-5p and miR-20a, negatively regulate the E2F1 transcription factor expression. These miRNAs are among six miRNAs that are activated by the transcription factor c-MYC that is also an activator of E2F1. The work gives an example of the complex interaction between transcriptional and translational control of gene expression: c-MYC simultaneously activates E2F1 transcription and downgrades its translation through the activation of additional miRNAs giving rise to a tightly controlled proliferative signal. Additional implications and potential downstream effects of this regulatory process are shown in Figure 9.2b.

9.1.4.1 Identification of microRNAs in the Genome Sequence

Identification of miRNAs in the genome sequences incorporates a set of characteristic criteria that define miRNAs

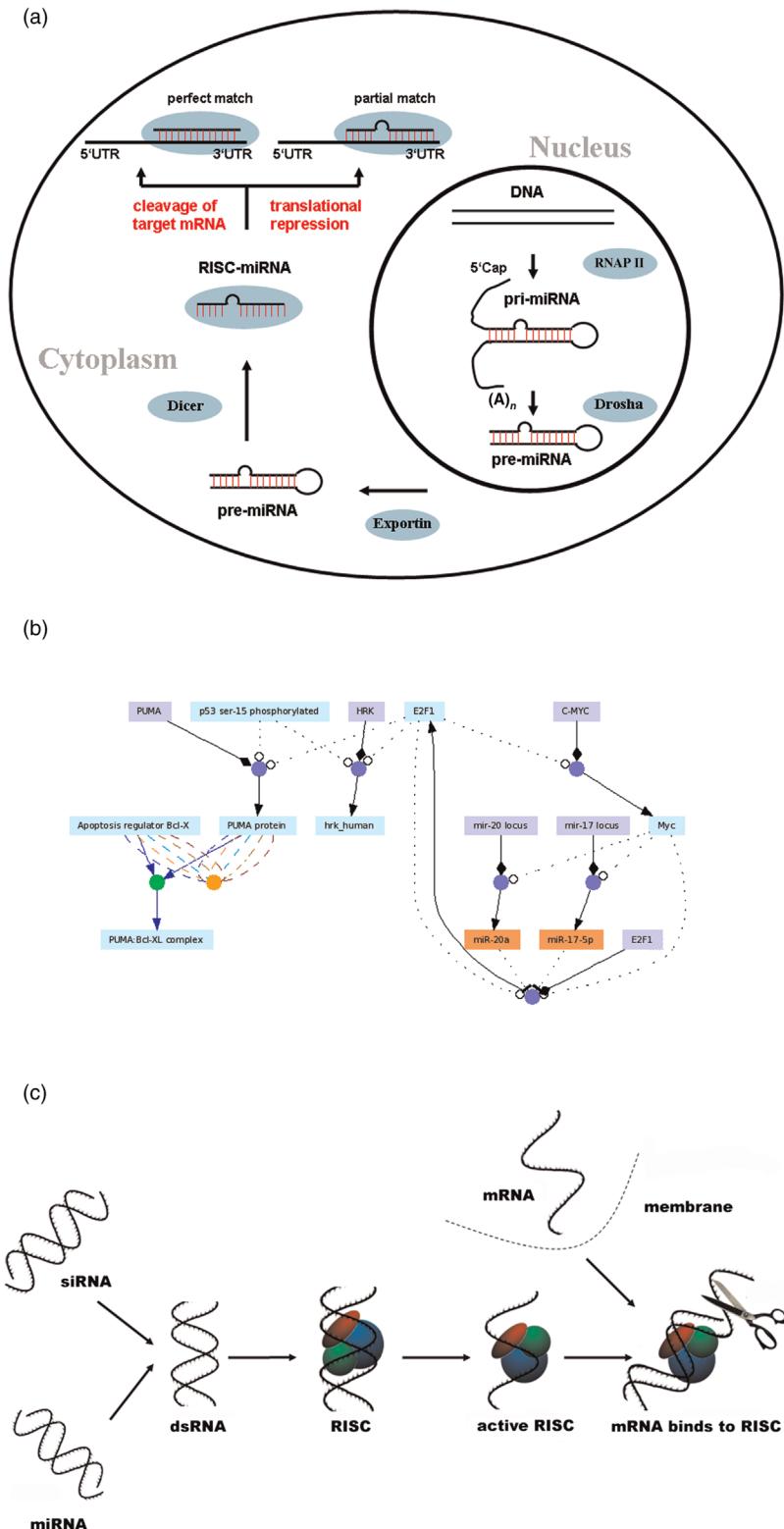


Figure 9.2 (a) Biogenesis of miRNA. (b) Network diagram that shows the tight interaction of two miRNAs with E2F1 and c-MYC regulatory networks. Different colors refer to different interaction types (blue: gene regulatory interaction; orange: protein–protein interaction; green: biochemical interaction). Different edge colors refer to different databases where these interactions had been retrieved from. The edge style refers to the interaction type. The network has been constructed with ConsensusPathDB, a database for pathway integration (cpdb.molgen.mpg.de). (c) Basic mechanism of RNAi.

and that usually involve the identification of pre-miRNA [30]:

- 1) The presence of a characteristic secondary structure involving hairpin modules.
- 2) Phylogenetic conservation.
- 3) Thermodynamic stability of hairpins and sequence relative to known miRNA.

These characteristics can be encoded into sequence alignment algorithms. Mostly, computational prediction of miRNAs is based on machine learning techniques (see Chapter 15) that use known miRNAs as a training set for the algorithm and then try to identify novel miRNAs. These miRNAs are then validated experimentally, for example, by PCR. Computer programs commonly utilize the hairpin shape of the precursor sequence of the mature miRNA. Multiple of these structures can be found in the genome and a majority of them are assumed not to be miRNA precursors. The programs thus search for predictive properties that distinguish the known miRNAs from this control group. Most algorithms depend on evolutionary conservation of miRNAs in different species. Common programs are MirScan, RNAFold, or PalGrade [31].

There are several public databases and resources for the analysis of miRNAs, the most common of which is miRBase (microrna.sanger.ac.uk), the central online repository for miRNA nomenclature, sequences, annotation, and target prediction of the Sanger Center. Currently, miRBase contains 5071 miRNA loci from 58 species. The number of known miRNAs in the genome is increasing rapidly and is under constant discussion. Whereas early studies estimated the number of mature miRNAs in the human genome to be 255 [32], recently this number has been increased to 555 [33]. Numbers for other species are as follows: *Mus musculus* (455), *Danio rerio* (183), *C. elegans* (135), *Drosophila melanogaster* (85), and the plant *Arabidopsis thaliana* (199).

9.1.4.2 MicroRNA Target Prediction

The identification of miRNA targets uses the fact that miRNAs recognize their targets by partial sequence complementarity and thus these targets can be identified by using the miRNA sequence itself. There are several programs available for miRNA target predictions (e.g., DIANAmicro, TargetScan, MiRanda, PicTar, and MicroInspector). These programs essentially perform two steps. In the first step, they identify potential miRNA binding sites according to specific base-pairing rules using different algorithmic approaches such as maximum likelihood and dynamic programming. Different programs distinguish between 5' and 3' base pairing of the miRNA and

also on the proportion of identity of these base pairings. A typical observation of miRNA–mRNA interactions in animals is a contiguous pairing in the miRNA 5'-region (proximal seed at positions 2–8) and less complementarity in the central part of miRNA (positions 10 and 11) that precludes the cleavage of the target mRNA in the middle of the duplex. In the second step, the programs implement cross-species conservation requirements taking advantage of the fact that miRNA regulation is highly conserved. Different programs have been compared with experimentally verified benchmark data sets [34]. Benchmarking data were taken from public data repositories and consisted of 84 experimentally verified miRNA–target interactions involving 23 different miRNAs. As a result of this, relatively small, benchmarking study, authors observed a rather bad performance of all programs. The sensitivity (defined as the true positive interactions divided by all known interactions) was <50% in all cases. More severe, the three programs that came close to this figure (MiRanda, TargetScan, and PicTar) predicted more than 10 000 different interactions, and thus generated high false positive rates.

9.1.4.3 Experimental Implications: RNA Interference

RNA interference uses the above-mentioned mechanism of posttranscriptional translation repression (*gene silencing, knockdown*) by reducing the concentration of the mRNA sequence and thus the quantity of the protein [35,36]. Using RNAi, it is possible to specifically degrade mRNA and reduce the activity of the corresponding protein (see Chapter 15). This has multiple implications for modeling gene regulatory pathways since it allows deducing, for example, target genes of certain transcription factors and enables the experimental measurement of functional gene interdependencies. The degradation of mRNA is initiated by short double-stranded RNA (dsRNA) molecules. Originally, these molecules were observed in the model organisms *C. elegans*, *D. melanogaster*, and *A. thaliana* [37–39], later also in vertebrates [40].

Different classes of regulatory dsRNAs are known, such as miRNAs and siRNAs. Gene silencing is induced as shown in Figure 9.2c. The dsRNA fragments are loaded into the silencing complex RISC and hybridize to the protein-coding RNA. The catalytic components of the RISC, the Argonaute proteins, are endonucleases and degrade the mRNA molecule [35].

RNAi has been developed as a routine and powerful experimental procedure, in particular, if it is combined with high-throughput experiments to screen the inhibition effects on a genome-wide scale. Through transfection of a cell culture with transcription factor-specific

siRNAs, it is possible to reduce the activity of the transcription factor to a large amount, thus allowing to measure the transcriptional consequences for its potential target genes. Figure 9.3 shows an example of an RNAi experiment with the transcription factor OCT-4, a crucial factor in early embryonic development [41]. Recent studies on reprogramming human somatic cells to induced pluripotent stem cells emphasize the function of the transcription factor OCT-4 as key regulator of pluripotency. It is supposed that pluripotency and self-renewal are controlled by a transcription regulatory network governed by OCT-4 regulation and the downregulation of OCT-4 is crucial in initiating for early embryonic differentiation. Clustering (see Chapter 15) has been applied to identify genes that change their expression patterns according to the knockdown measured at 24 and 72 h in comparison with an unspecific knockdown at the same time points. Expression profiles and expression fold changes can then

be interpreted as effects of the knockdown on the underlying molecular network structure.

Figure 9.3 shows specific clusters of coexpressed genes across the different experiments. One cluster, for example, shows genes with a continuous downregulation through time including the transcription factor OCT-4 itself and many of its known target genes. Another cluster shows a continuous upregulation of genes that are downstream effects of the loss of stemness characteristic and indicative of cell differentiation such as BMP4. Most genes show no expression differences indicating the specificity of the knockdown.

It should be noted that RNAi analysis cannot distinguish between direct and indirect targets of a transcription factor since no information on DNA–protein binding is given. The experiment gives an impression on all potential downstream effects of the knockdown and thus the number of predicted targets is typically

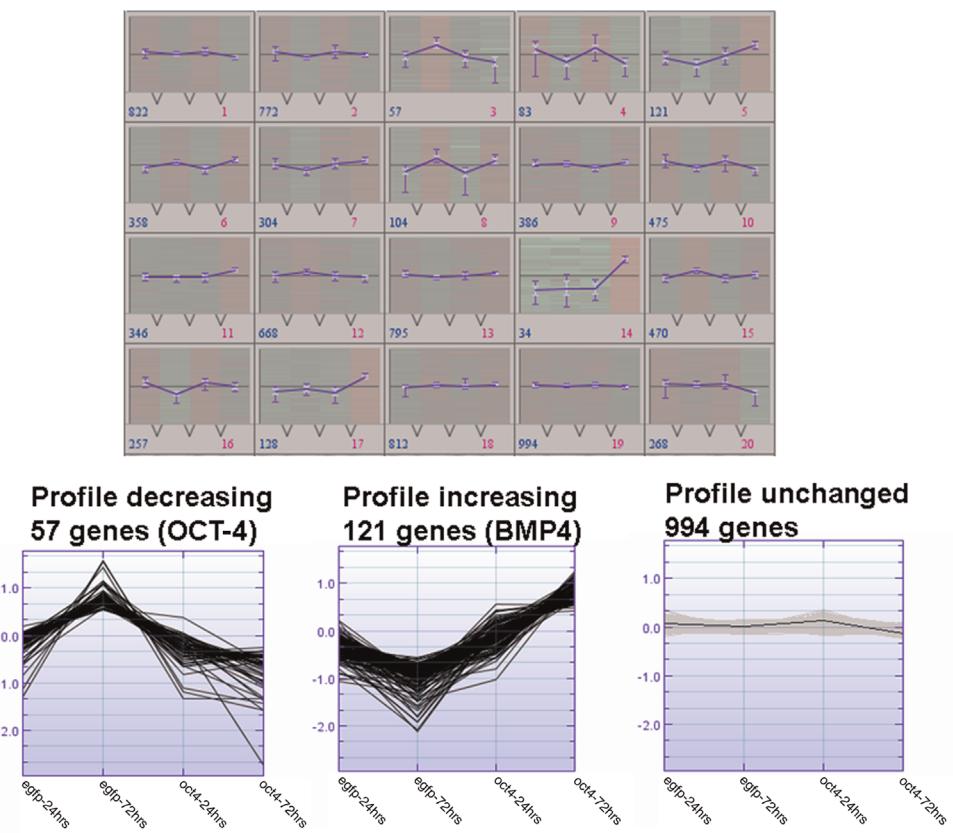


Figure 9.3 Resulting expression patterns from an RNAi experiment. Microarray experiments have been carried out using an OCT-4 knockdown at 24 and 72 h and an unspecific ECFR knockdown as a control at the same time point after transfection. The resulting values were clustered using a K-means algorithm that resulted in 20 clusters, each of which contained genes with a typical expression profile (top). The lower panel shows three examples of clusters with decreasing expression pattern (OCT-4-like genes), increasing expression patterns (BMP4-like genes), and constant expression patterns.

large. In order to reveal the exact transcriptional dependencies, these measurements must be complemented by other techniques, for example, chromatin immunoprecipitation (ChIP) followed by microarray analysis or sequencing and by computational analysis of the promoter sequences.

9.2 Dynamic Models of Gene Regulation

Summary

In order to comprehend the functioning of organisms at the molecular level, we wish to know which genes are expressed, to what level, where, and when. The regulation of gene expression is performed through a network of interactions between DNA, mRNA, proteins, and other molecules. This network comprises many components. According to the central dogma of molecular biology formulated by Francis Crick [42], there is a forward flow of information from gene to mRNA to protein. Moreover, positive and negative feedback loops and information exchange with signaling pathways and energy metabolism ensure the appropriate regulation of expression according to the current state of the cell and the environment.

Modeling of gene expression is used as an example to apply different modeling techniques. The dynamics or the regulatory patterns of gene expression will be mathematically described with various graphs, Boolean networks, Bayesian networks, ordinary and partial differential equation systems, and stochastic equations or with rule-based formalisms.

9.2.1 A Basic Model of Gene Expression and Regulation

Based on the central dogma of molecular biology stating that a gene codes for mRNA that in turn is a template for the protein, gene expression can be mathematically described with systems of ordinary differential equations (ODEs) in the same way as dynamical systems in metabolism, signaling, and other cellular processes. We consider the little network depicted in Figure 9.4a. Its dynamics can be described with the following equation system:

$$\begin{aligned}\frac{dmRNA}{dt} &= v_1 - v_2 = k_1 \cdot TF - k_2 \cdot mRNA, \\ \frac{dprotein}{dt} &= v_3 - v_4 = k_3 \cdot mRNA - k_4 \cdot protein.\end{aligned}\quad (9.2)$$

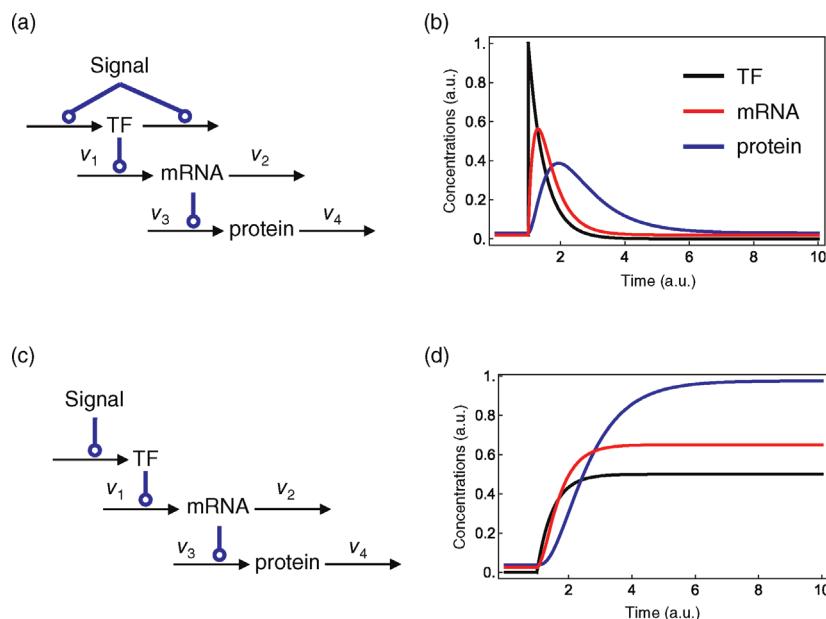


Figure 9.4 Gene expression according to the central dogma. (a) Reaction network. Reaction v_1 denotes transcription, that is, the creation of mRNA from amino acids according to the respective gene sequence. TF is the transcription factor enabling or regulating this process. Reaction v_2 is mRNA degradation. Reaction v_3 stands for translation, that is, the creation of protein from nucleic acids according to the mRNA information. Reaction v_4 represents protein degradation. (b) Dynamic effects of transient activation of transcription factor. (c) Network for sustained activation of TF. (d) Dynamic simulation of the network in (c). Parameter values: $k_1 = 5$, $k_2 = 4$, $k_3 = 1.5$, $k_4 = 1$, $TF(0) = 0.02$, $mRNA(0) = 0.025$, and $protein(0) = 0.0375$ (all arbitrary units).

Here, we used mass action kinetics for production and degradation of mRNA and protein. More advanced gene regulation functions will be discussed in Sections 9.2.2 and 9.3. If the cellular conditions are constant, that is, if amino acids and nucleic acids are available in sufficient amount, if transcription factors remain unchanged, and if also degradation processes are not specifically regulated, then the system reaches a steady state, where mRNA and protein assume the following values:

$$\begin{aligned} mRNA^{ss} &= \frac{k_1}{k_2} \cdot TF, \\ protein^{ss} &= \frac{k_1 k_3}{k_2 k_4} \cdot TF. \end{aligned} \quad (9.3)$$

Typical values for mRNA abundances and for protein concentrations in cells have been determined both in studies focusing on a single or a few genes and in large-scale, genome-wide studies. Global data for human cells are provided, for example, by a recent study [43] and data for the model organism yeast can be found, for example,

in Refs [44,45] (protein numbers) and [46] (RNA expression).

The expression of genes is subject to many different types of regulation. First of all, the availability or amount of transcription factors may change. It is frequently considered that activating transcription factors are switched on, for example, due to an external stimulus that is transmitted via a signaling pathway (see Chapter 12) or due to internal changes of the cell state such as cell cycle progression. Figure 9.5 shows the dynamic behavior of the basic gene expression module upon changes of transcription factor activity. We can distinguish between transient activation of TF (Figure 9.5b) and sustained activation of TF (Figure 9.5c). Although transcriptional regulation has attained most attention in many gene regulation studies, mRNA and protein levels can also be influenced by the regulation of translation (Figure 9.5d) or by adjusted degradation (Figure 9.5e for mRNA and Figure 9.5f for protein).

In experimental studies, it can be of interest to regulate the available mRNA of a specific gene. Genetic knockout

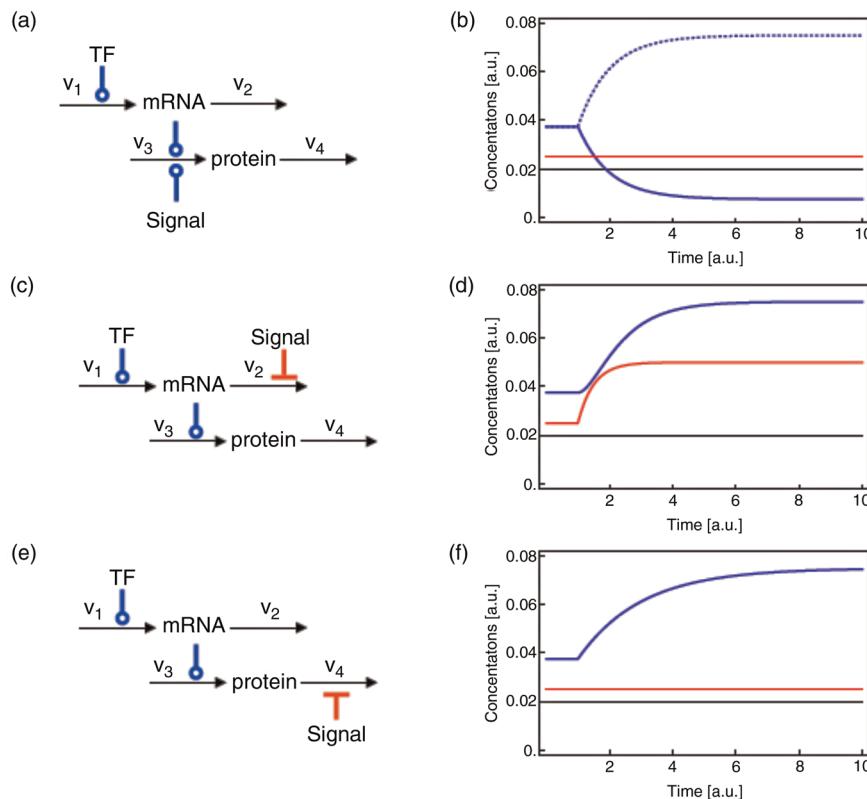


Figure 9.5 Regulation of gene expression by signaling. (a) Reaction network exhibiting translation control: inhibition of translation (solid line) or translational activation (dotted line). (b) Time courses for network shown in (a). (c) Inhibition of mRNA degradation. (d) Time courses for network shown in (c). (e) Inhibition of protein degradation. (f) Time courses for network shown in (e). For notation and parameter values, see Figure 9.4.

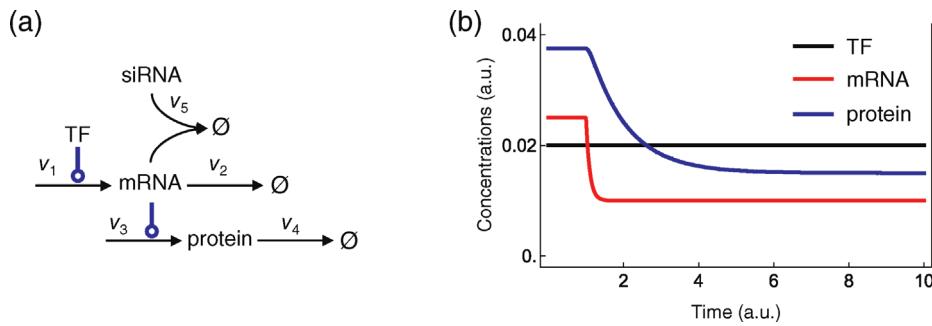


Figure 9.6 Downregulation of mRNA by siRNA. (a) Network scheme. (b) Time course of transcription factor, mRNA, and protein upon addition of siRNA at time point 1. Parameter values are as in Figure 9.4 and in addition $k_5 = 100$ and $siRNA(0) = 10$ (all mass action kinetics).

completely removes the respective gene product. Alternatively, the experimental technique of siRNA (see Chapter 14) gradually removes the mRNA through binding of mRNA and siRNA. This process is exemplarily shown in Figure 9.6.

Modulation of mRNA transcription also happens by natural means through miRNAs (Figure 9.7). Human cells contain more than 1500 different miRNAs and the majority of genes are regulated in one or the other way by these miRNAs. miRNAs are expressed sequences. The microprocessor complex, containing an RNase III-type endonuclease and a cofactor, cleaves ~70 nt long precursor hairpins (pre-miRNA) from exonic or intronic sequences of long primary transcripts in the nucleus. They first form a so-called stem-loop structure with specific base pairing. They are then exported from the nucleus and processed by the endonuclease Dicer resulting in a mature sequence of about 22–25 nucleotides in length. These are bound to the AGO protein. One strand, the passenger strand, is degraded, while the mature miRNA is loaded into AGO forming a functional RISC. This complex binds by complementary base pairing to mRNA, inducing its cleavage and degradation. A number

of functions for miRNAs are discussed, including the suppression of leaky transcription, the suppression of obsolete transcripts to reinforce cellular identity, or fine-tuning of protein expression.

The cellular microRNAs in turn may be regulated by circular RNAs (circRNAs), which are suggested to serve as a sponge for superfluous miRNAs [47]. In eukaryotic cells (human, mouse, and nematode), thousands of well-expressed, stable circRNAs were detected, which often showed expression specific for a tissue or developmental state. Through sequence analysis, important regulatory functions for circRNAs have been revealed. A human circRNA has been found to be densely bound by miRNA effector complexes and shown to harbor 63 conserved binding sites for the ancient miRNA miR-7. These and further data provided evidence that circRNAs form a large class of posttranscriptional regulators. Numerous circRNAs form by head-to-tail splicing of exons, suggesting previously unrecognized regulatory potential of coding sequences [47]. Figure 9.8 shows a potential regulation mechanism.

If such systems are described using systems of ODEs, it is possible to take into account more detailed knowledge

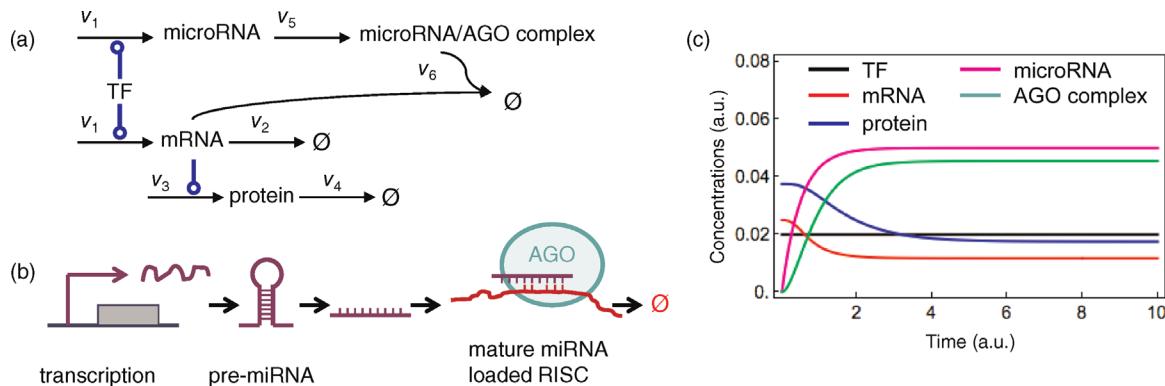


Figure 9.7 The effect of miRNA on mRNA abundance and protein expression. (a) Regulatory network. (b) Scheme of the miRNA biogenesis and mRNA repression. (c) Time courses. Parameter values are as in Figure 9.4 with $k_5 = 1$ and $k_6 = 100$ (all mass action kinetics).

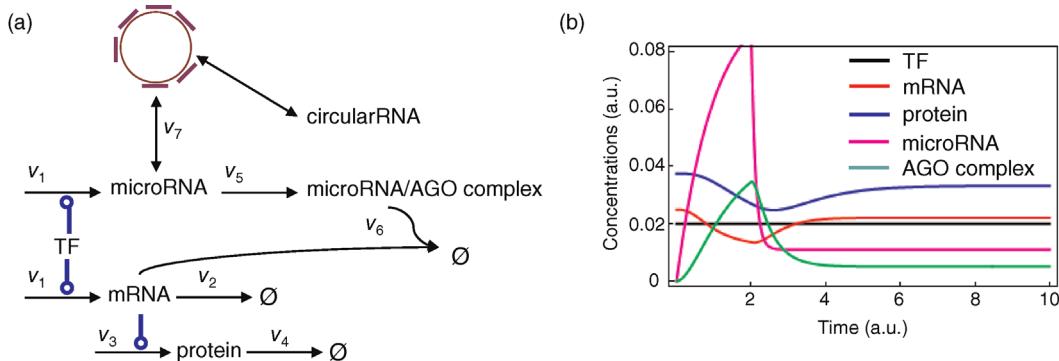


Figure 9.8 Potential effect of circRNA on the regulation of miRNAs. (a) Regulatory network. (b) Time courses. Parameter values are as in Figure 9.7 and $k_7 = \begin{cases} 0, & \text{if } 0 < t < 2 \\ 8, & \text{if } 2 \leq t \end{cases}$ (all mass action kinetics).

about gene regulatory mechanisms, such as information about the kinetics, individual interactions of proteins with proteins or proteins with mRNA, and so on. An obstacle is the current lack of exactly this type of knowledge, the lack of kinetic constants or time-resolved concentration data due to measurement difficulties and uncertainties in the function of many proteins and their interactions.

9.2.2 Natural and Synthetic Gene Regulatory Networks

In their pioneering work entitled “Genetic regulatory mechanisms in the synthesis of proteins,” François Jacob and Jacques Monod introduced a model for gene expression regulation. Based on a manifold of biological and biochemical observations, they distinguished between *structural genes*, which determine molecular organization, and *functional determinants*, which control the rate of protein synthesis. This new class of genes codes for repressors, which can be activated or inactivated by specific metabolites, and hence repress or induce the expression of their controlled genes.

This general concept has been analyzed for many biosynthetic pathways, especially in bacteria such as *E. coli*: An example for repression is a biosynthetic pathway producing amino acids. The synthesis of the enzyme tryptophan synthase is regulated by a structural gene. Enzyme synthesis occurs in the absence of tryptophan, while the presence of tryptophan leads to a stop of synthesis. The synthesis of β -galactosidase, in contrast, has been found to be inducible. In the absence of galactosides, only a very low synthesis of β -galactosidase occurs, while the synthesis rate increases 10 000-fold in the presence of galactoside. Both repression and induction of enzyme synthesis are very specific processes.

Gene regulation has been studied for many example cases. It has gained new interest with the rise of synthetic biology, where researchers try not only to understand the logic of naturally occurring gene regulation networks, but also to implement gene regulation circuits with predictable behavior. The so-called genetic engineering has been used not only to test ideas and principles, but also to create genetically modified organisms with novel traits such as the ability to sense chemicals in the environment, to efficiently synthesize desired molecules, or to be resistant against fungicides. A basic idea behind this approach is that gene regulatory circuits with virtually any desired property can be constructed from networks of simple regulatory elements. Starting with the pioneering work of Gardner, Cantor, and Collins in 2000, a series of gene regulatory networks have been implemented that show behavior such as bistability, multistability, oscillations, transient or sustained activation, repression, or pulse generation.

The general architecture of a transcriptional regulation unit is depicted in Figure 9.9. It comprises the gene coding for the desired protein starting at the transcription start (TS) site and the promoter of this gene. In the promoter region are binding sites for transcription factor, the transcriptional machinery, and the regulatory proteins.

Repressors are proteins binding to the operator in order to prevent binding of the RNA polymerase and thus gene transcription. Inducers bind to the repressor preventing them to bind to the operator. In *E. coli*, a number of repressors are known. In the Tet-On and Tet-Off systems, a recombinant tetracycline-controlled transcription factor interacts with the responsive promoter, P_{tet} , to enable the expression of the gene. Tetracycline or doxycycline, in turn, regulates the expression by influencing the DNA binding of the transcription factor (rtTA or tTA). In case of Tet-On, tetracycline is required for the

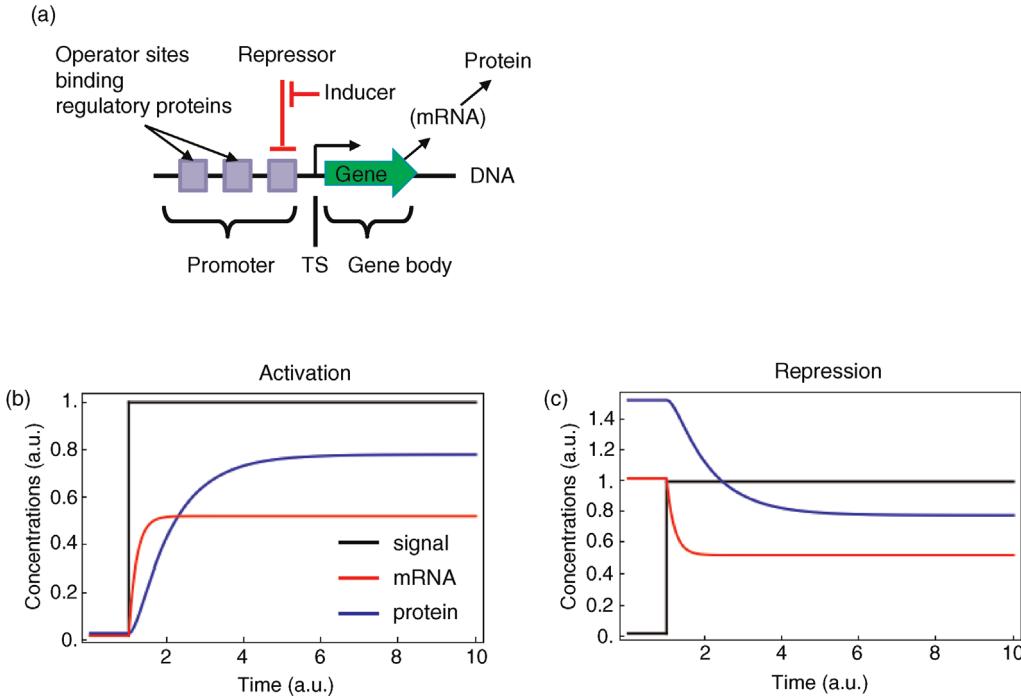


Figure 9.9 Organization and temporal behavior of a transcription regulation unit. (a) The gene codes for an mRNA that is then translated into the desired protein. The expression is dependent on transcription factors and the regulatory protein. The activity of regulatory proteins is, in turn, regulated by modifiers, that is, the repressors or inducers. (b) Dynamics of activation. The inducer is added at time $t = 1$, followed by the expression of the regulated mRNA and protein. (c) Dynamics of repression. Initially, mRNA and protein are expressed. Addition of inducer at time leads to inhibition (reduction) of expression. Remaining basal expression is ensured by the basal transcription rate k_{1b} in Eq. (9.4). Parameters in (b) and (c): $k_{1b} = 0.02$, $k_1 = 5$, $K = 1$, $n = 4$, $k_2 = 5$, $k_3 = 1.5$, and $k_4 = 1$.

DNA binding; hence, addition of tetracycline switches gene expression on. For the Tet-Off system, tetracycline prevents TF-DNA interaction leading to a repression. The protein LacI targets the Lac operon with the regulated promoters P_{Lac} , P_{trc} , or P_{LacO} and its cognate inducers lactose and IPTG (isopropyl- β -D-thiogalactoside). Other regulators are the bacteriophage-derived repressor CI or the activator LuxR, which requires acyl homoserine lactone (AHL) as inducer.

The expression of the genes of interest can be generally measured by their downstream effects. To report expression in single cells, fluorescently labeled proteins are used, such as the green fluorescent protein (GFP) and its descendants such as YFP or CFP.

An important aim in the construction of synthetic regulatory modules is a clear switch-like behavior upon addition of the repressor or inducer. Since this is typically not ensured by the mass action kinetics used in the basic modules introduced above (Section 9.2.1, Eq. (9.2)), synthetic modules are described with Hill kinetics (see Chapter 4 for an extended description of Hill kinetics and other kinetic laws and their properties). Depending on whether we want to describe induction or repression, whether we vary the abundance of the regulatory protein

or its inducer, and depending on the level of desired detail, we can employ different mathematical models.

For the case of gene expression activation by the regulatory protein, mRNA and protein dynamics read

$$\begin{aligned}\frac{dmRNA}{dt} &= k_{1b} + k_1 \cdot \frac{(S/K)^n}{1 + (S/K)^n} - k_2 \cdot mRNA, \\ \frac{dprotein}{dt} &= k_3 \cdot mRNA - k_4 \cdot protein,\end{aligned}\quad (9.4)$$

where k_{1b} denotes a basal transcription rate and $k_1((S/K)^{-n} + 1)^{-1}$ is the regulatory term. The signal S represents the combined action of regulatory protein RP and inducer I, as detailed below.

Repression is described in a similar manner:

$$\begin{aligned}\frac{dmRNA}{dt} &= k_{1b} + k_1 \cdot \frac{1}{1 + (S/K)^n} - k_2 \cdot mRNA, \\ \frac{dprotein}{dt} &= k_3 \cdot mRNA - k_4 \cdot protein\end{aligned}\quad (9.5)$$

(note the difference in the regulatory term).

The behavior of the signal depends on the effect that the inducer I has on the regulatory protein RP. If the inducer is required for RP to bind the promoter or enhances promoter binding, it can be calculated, for

example, as $S = RP \cdot I$ or $S = RP \cdot (1 + (I/K_1)^n)$, respectively, where K_1 and n_1 denote dissociation constant and Hill coefficient for the inducer binding to RP. If the inducer inactivates the repressor, the signal results as $S = RP/(1 + (I/K_1)^n)$. Examples for the dynamic behavior of activation and repression following the addition of inducer are shown in Figure 9.9.

The dynamics described in Eqs. (9.4) and (9.5) entail a certain delay in the production of protein from mRNA. This delay can be required for some kinds of response dynamics such as oscillations. If those dynamic features are of less interest, we can simplify the description and only focus on the protein dynamics using the following model:

$$\frac{d\text{protein}}{dt} = k_b + k_r \cdot \frac{(S/K)^\delta}{1 + (S/K)^n} - k_d \cdot \text{protein}. \quad (9.6)$$

Here, k_b is the basal expression rate, and k_r and k_d are the rate constants for signal-dependent expression and for protein degradation, respectively. The value of δ distinguishes the types of regulation, that is, repression ($\delta = 0$) or activation ($\delta = 1$).

In their pioneering work from 2000, Gardner, Cantor, and Collins constructed a genetic toggle switch in *E. coli* by combining two repressors. Repressor 1 inhibits the transcription from promoter 1 and is induced by inducer 1, while repressor 2 inhibits transcription from promoter 2 and is induced by inducer 2. Promoter 1 regulates expression of repressor 2 and promoter 2 regulates expression of repressor 1. The network and the potential behavior of the toggle switch are shown in Figure 9.10. Figure 9.10b shows the phase plane for the two repressor proteins. The blue and the red lines are the nullclines for proteins 1 and 2, that is, the lines for $d\text{protein}_1/dt = 0$ and $d\text{protein}_2/dt = 0$, respectively. Intersections of nullclines are steady states. In our case, the steady state in the middle (where the separatrix crosses) is unstable, while the other two steady states are stable. The separatrix separates the areas of attraction for the two stable steady states; that is, all initial conditions above the separatrix lead to a dynamic move toward the steady state above it. The precise behavior in the phase plane depends on the choice of parameter values. It is easy to see that two

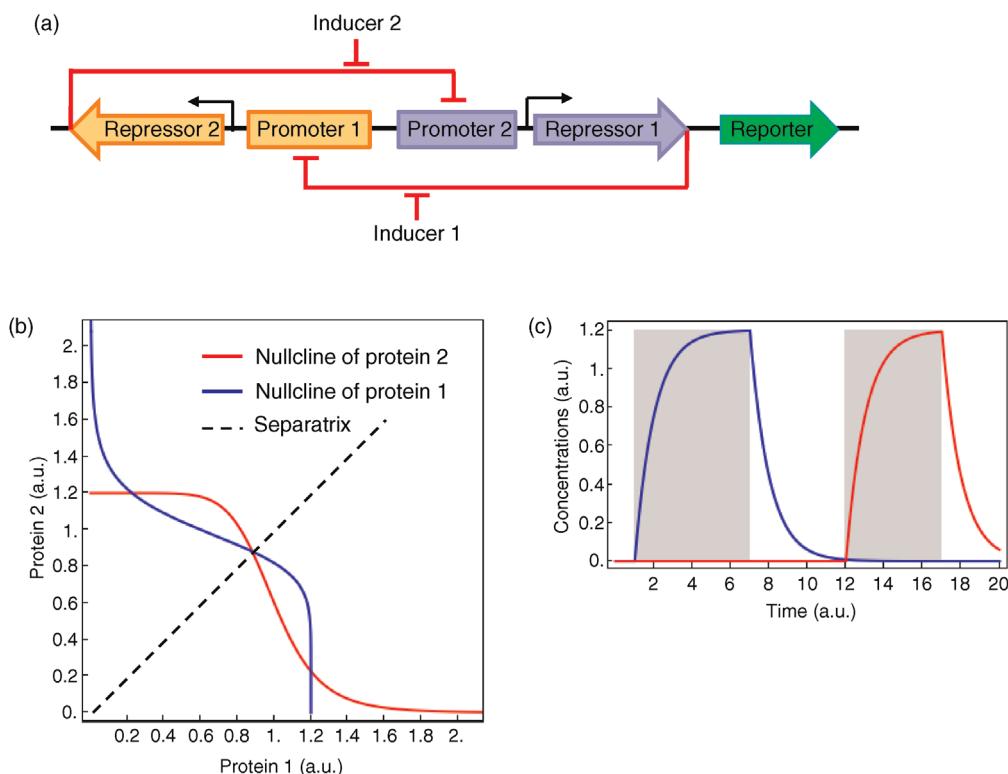


Figure 9.10 Genetic toggle switch. (a) Organization of the regulatory network. (b) Bistable behavior of the toggle network. Red and blue lines are the nullclines for both proteins. Intersections of the nullclines are steady states, where the middle one is unstable and the other two are stable. The separatrix divides the phase plane into two areas where the system moves from all initial conditions within these areas to the stable steady state within the respective area. (c) Dynamic behavior of toggle network. Starting from an all-off state, protein 2 is expressed if inducer 1 is added (in the period $1 \leq t \leq 7$). Afterward, expression declines. When inducer 2 is added (in the period $12 \leq t \leq 17$), protein 1 gets expressed. Parameter values for (b) and (c): $k_{b1} = k_{b2} = 0$, $k_{r1} = k_{r2} = 1.2$, $K = 1$, $n = 4$, and $k_{d1} = k_{d2} = 1$.

of the intersection points would vanish if we move either the blue line upward (by increase of K_2) or the red line to the right (by increase of K_1). Note that bistability in this network is enabled by the use of Hill kinetics, which leads to the sigmoidal shape of the nullclines.

Based on the general architecture of an operon as transcriptional regulation unit, many regulatory modules have been created, which fulfill different functions and exhibit different types of dynamics or steady-state behavior. The simplest modules transmit information in a linear manner – one gene codes for the repressor regulating the next gene. Figure 9.11a and b shows such an example. Linear modules can, for example, also create delays. Most of the other transcriptional modules contain either positive or negative feedback or elaborated combinations of that. They may produce many interesting types of dynamic behavior. A classical example is the repressilator consisting of three genes regulating each other in a circular manner (Figure 9.11c and d). The repressilator shows oscillations with the appropriate choice of parameters.

9.2.3 Gene Expression Modeling with Stochastic Equations

For the regulation of gene expression, it can be argued that the assumptions of continuous and deterministic

concentration changes underlying the description with ODEs are not valid since each gene is present in only one, two, or a few copies. The number of transcription factor molecules is usually small (on the order of tens or hundreds). Even the abundance of mRNA molecules is often below the detection limit. Therefore, it is not sure how many of the objects of the considered processes are actually present, and the character of the events becomes probabilistic. Furthermore, the involved processes can hardly be regarded as continuous. In transcription, for instance, it takes a certain time from the initiation until termination. The discrete and probabilistic character of processes is taken into account in stochastic modeling of gene regulation, which has been discussed in more detail in Section 7.2. In the most common approach, the state variables are discrete molecule numbers x . The number of molecules of each species can change only by simple reaction steps (formation of a molecule, complex formation, and degradation). A convenient way to simulate the dynamics of such a stochastic system is the Gillespie algorithm [48] (see Section 7.2). This algorithm provides an individual realization, that is, trace of the system through the states space as time evolves. It computes molecule numbers for each chemical species along the time axis. Repeated execution of the algorithm with the same initial conditions yields several realizations that can be used for statistical analysis, such as calculation of

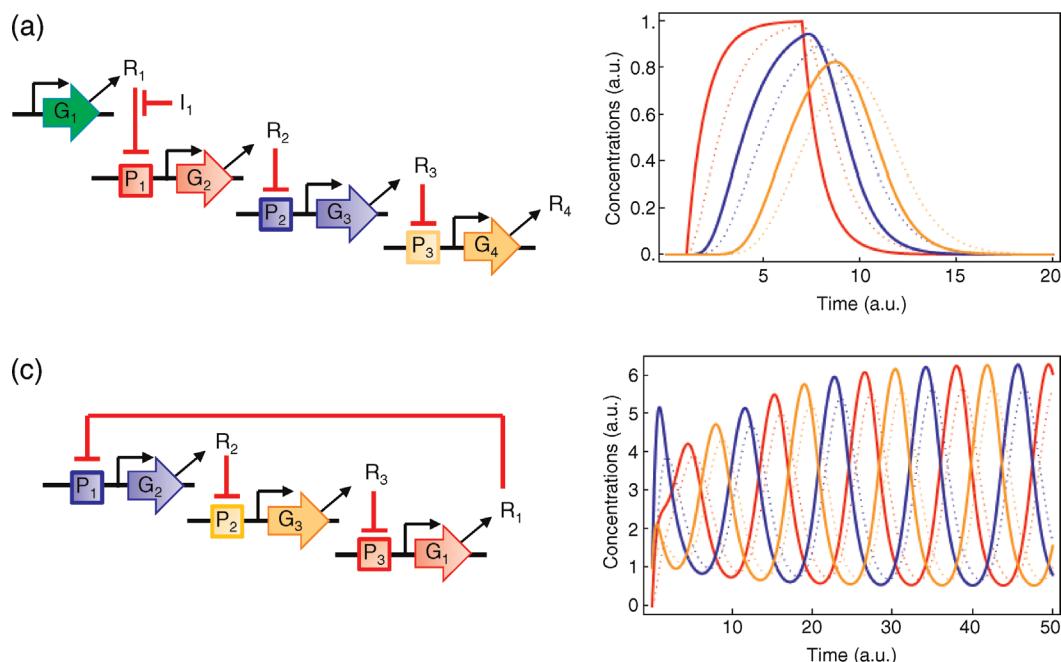


Figure 9.11 Complex regulatory networks. (a) Linear combination of basic transcriptional units. (b) Simulation of the network in (a) with R_2 in red, R_3 in blue, and R_4 in orange. Kinetics of all three units are described by the equation system (9.5). Parameter values for all three units: $k_b = 0.02$, $k_1 = 2$, $k_2 = 1$, $k_3 = 1$, $k_4 = 1$, $K = 1$, and $n = 4$. (c) The repressilator: R_1 exerts feedback on the regulation of R_2 . (d) Oscillatory behavior (colors: R_2 in red, R_3 in blue, and R_4 in orange). Parameter values for all three units: $k_b = 0.02$, $k_1 = 2$, $k_2 = 1$, $k_3 = 1$, $k_4 = 1$, $K = 1$, and $n = 4$.

means and percentiles. Alternatively, we can consider the probability $p(x, t)$ that there are x molecules of type x present at time t .

The advantage of stochastic modeling compared with deterministic approaches is the explicit consideration of uncertainties due to the stochastic and discrete character of processes involving low molecule numbers, which is especially important in gene expression, where mRNA is often present only in small numbers. In some cases, experimentally observed behavior (i.e., switching between different states) could be explained with fluctuations that are inherent to stochastic modeling, but not to differential equations. A problem common to stochastic equations and ODEs is the limited knowledge about appropriate kinetic parameters. Furthermore, the stochastic simulation of large systems demands excessive computational power, especially for higher molecule numbers. Therefore, algorithms are developed to combine stochastic simulation for low-abundance species with deterministic simulations for high-concentration species.

9.3 Gene Regulation Functions

Summary

Gene expression is controlled by specific transcription factors and general cellular resources such as RNA polymerase. The kinetics of transcription can be described by gene regulation functions, which follow from analyzing the possible microscopic binding states of gene promoters. For some promoters, for example, the Lac promoter in *E. coli*, regulation functions have been determined by fitting theoretically derived shapes to measured transcription data. In larger transcription networks, gene regulation functions and regulator activities can be estimated from measured gene expression and known network structures.

Regulation of gene expression is a central control mechanism in cells, which closes the global feedback loop between cell state and protein production [49]. Biochemically, mRNA transcription is controlled by regulatory proteins, for example, transcription factors, which bind to regulatory sites on the DNA and modulate the promoter activities of genes or operons. The effect of regulators on gene expression can be positive or negative, weak or strong, and the inputs by different regulators can be processed in complicated ways.

Figure 9.12 shows a small genetic network, comprising the dual regulator MetR in *E. coli* together with its target genes and other regulators controlling them. To translate

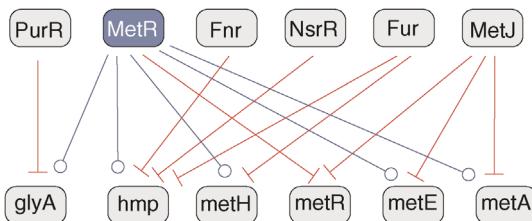


Figure 9.12 A gene regulon in *E. coli*. MetR (blue) regulates a number of target genes (bottom). Other regulators controlling these genes are shown on top. Arrows denote transcriptional regulation (red: repression; blue: induction). (Data taken from the EcoCyc database [50].)

such a picture into a dynamic model of gene regulation, arrows must be replaced by quantitative *gene regulation functions*, the rate laws of transcription. Mechanistically, these functions translate input stimuli (e.g., the levels of active transcription factors) into mRNA transcription rates. Functionally, they reflect the demand for specific protein levels in different cell states and implement a mechanism to satisfy these demands.

Gene expression is regulated both globally, leading to a global up- and downregulation of genes, and specifically by transcription factors acting on smaller sets of target genes. Global expression changes depend on molecule resources such as RNA polymerase that can vary with the cell growth rate. In eukaryotes, models of transcriptional regulation must account for chromatin remodeling by nucleosomes [51]. Histone binding, which affects the accessibility of genes for transcription, may be regulated by the cell's energy status and can affect the balance of anabolic and catabolic processes, which would create another large feedback loop [52]. In this section, we focus on regulation by transcription factors and study how gene regulation functions can be modeled and measured.

9.3.1 The Lac Operon in *E. coli*

The concept of gene regulation was developed by J. Monod, following his discovery of diauxic growth in bacteria [53]. *E. coli* bacteria prefer glucose as their energy source and maintain enzymes for glucose metabolism under all conditions. Aside from glucose, the bacteria can utilize other sugars such as lactose. Three enzymes are necessary for consumption of lactose: β -galactosidase, permease, and thiogalactoside transacetylase; these are encoded and regulated together in a transcription unit called the *Lac operon*. When cells are shifted from a glucose-rich to a glucose-free, but lactose-rich medium, it takes a while until the cell population can assimilate lactose at a high rate. The reason is that the *Lac operon* is induced on demand, that is, if glucose is missing and

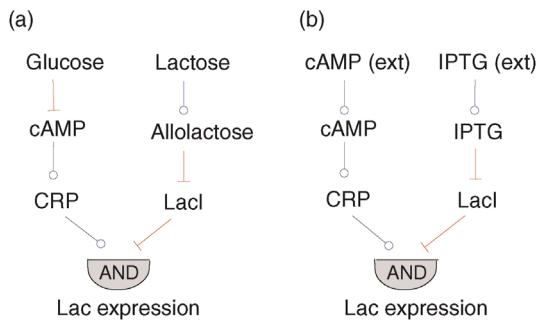


Figure 9.13 Regulation of the Lac operon. (a) *Natural induction*: The Lac operon is controlled by the transcription factors CRP and LacI, which respond to extracellular levels of lactose and glucose. A high expression requires the presence of lactose and absence of glucose. (b) *Artificial induction*: In experiments, the activities of CRP and LacI can be regulated by extracellular levels of the alternative ligands cAMP and IPTG [54]. Effectively, both substances activate Lac expression.

lactose is present in the medium. Approximately, Lac expression follows the logical rule “low glucose AND high lactose.”

Biochemically, the transcription rate is controlled by two signals (see Figure 9.13a). On the one hand, a high glucose level decreases the concentration of cyclic AMP (cAMP), an intracellular messenger that activates the transcriptional activator CRP. Thus, at high glucose levels CRP remains inactive and Lac transcription is low. Lactose, on the other hand, is sensed via its isomer allolactose. Allolactose inhibits the transcriptional repressor LacI, which would otherwise shut down Lac expression. Therefore, if no lactose is present, Lac expression will also be low.

Figure 9.14 shows different binding states of the Lac promoter in a simple scheme with five states. For a strong

expression of the Lac operon, the repression must be released and the activator CRP must be bound. Normally, this requires glucose to be absent and lactose to be present, as stated by the logical function. In experiments, the regulators CRP and LacI can also be controlled by extracellular levels of cAMP and of IPTG, a substitute for allolactose (Figure 9.13b).

9.3.2

Gene Regulation Functions Derived from Equilibrium Binding

To set up kinetic models of gene expression, we need to specify rate laws for the transcription rate. We assume that a transcription rate y depends on input variables such as transcription factor activities x_i and describe it by a *gene regulation function*

$$y(t) = f(\mathbf{x}(t), \mathbf{p}), \quad (9.7)$$

where the vector \mathbf{x} contains the activities of all regulators for the gene in question. The function f and its parameter vector \mathbf{p} are gene-specific; however, the same form of f may be used for different genes. The meaning of function (9.7) may vary from case to case. In some experiments, the expression of a gene is controlled by extracellular substances such as IPTG. In this case, a response function $f(\mathbf{x}(t))$ will describe the overall effect of this substance on gene expression, possibly mediated by a transcriptional regulation cascade. In this chapter, most gene regulation functions refer to *direct inputs* such as transcription factors, but response and regulation functions cannot always be clearly distinguished. Eukaryotic promoters can process a large number of inputs, and their regulation functions can be complicated [56]. For simple prokaryotic genes, plausible gene regulation functions can be derived from theoretical models [57,58] and fitted to experimental data [54,59,60].

What determines a gene regulation function? Effectively, it subsumes microscopic processes that affect transcription initiation, such as the binding of regulators to DNA, and is largely determined by the promoter's nucleotide sequence. Figure 9.15 shows how promoter sequences can determine gene regulation functions: a promoter can assume various microscopic states, with different regulators bound and different conformations of the DNA, allowing or not allowing for initiation of transcription. To translate this idea into a quantitative model, we make three basic assumptions: (i) On the time scale of interest, the different states of the promoter are in thermodynamic equilibrium. (ii) The probability for each state follows from its free energy, which can depend on the concentrations of free regulator molecules. (ii) In

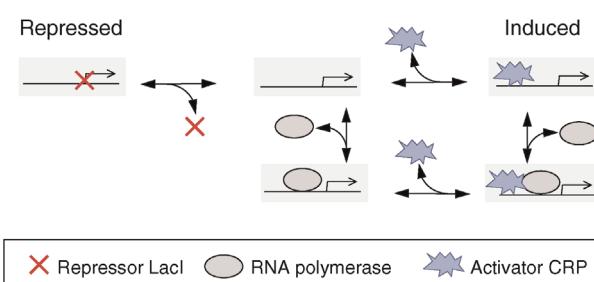


Figure 9.14 Microscopic states of the Lac promoter (schematic model). The promoter can be bound by RNA polymerase, the activator CRP, and the repressor LacI. Bound repressor inhibits binding of other molecules (left), while bound activator increases the probability of polymerase binding (right). Transcription can occur in states with bound polymerase (bottom). In reality, the possible binding states are more complex: LacI can bind to several binding sites, dimerize, and thereby cause DNA looping. A detailed model with 50 states is described in Ref. [55].

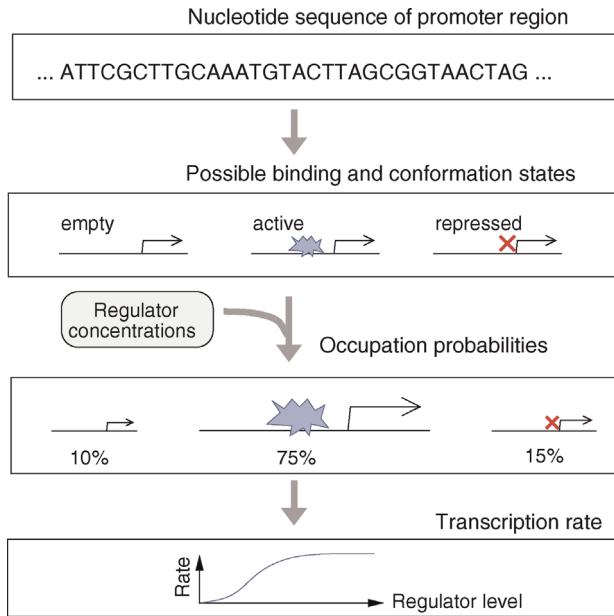


Figure 9.15 Gene regulation functions are encoded by nucleotide sequences in the promoter region. Transcription factor binding sites give rise to a number of possible binding states, which trigger transcription initiation with different rates. The probabilities of these states depend on the level of active transcription factors and on the energies for transcription factor binding and DNA bending, which are determined by the nucleotide sequence.

each state, transcription initiation occurs randomly and at a certain state-specific rate.

With these assumptions, the shape of a gene regulation function f_i can be derived from an analysis of promoter states. Each state has a free energy $F = E - TS$, where E denotes the energy of the state, S the entropy, and T the temperature. The energies depend on the binding of different regulators or on the formation of DNA loops, which may interfere with each other, and on the presence and sequences of regulator binding sites. The entropy term depends, among other things, on the number of free regulator molecules. From the free energy F of a promoter state follows its statistical weight $w_i = \exp(-F_i/(k_B T))$ in the Boltzmann distribution, and the total transcription rate is given by the weighted average

$$y = \frac{\sum_i w_i v_i}{\sum_i w_i} \quad (9.8)$$

over the transcription initiation rates in different states. If we write the transcription rate as a function of regulator concentrations, we obtain a formula for the gene regulation function. Its mathematical form and parameters are determined by the microscopic states and their energies. In practice, it is hard to directly measure or compute the

microscopic energies, but the parameters in the formula can be estimated by fitting the function to measured transcription rates.

9.3.3

Thermodynamic Models of Promoter Occupancy

Macroscopic gene regulation functions arise from a promoter's fluctuating transitions between different binding states. In a model by Bintu *et al.* [57], regulator proteins can bind in various places on the DNA: at the specific binding site, they are bound tightly with a large negative binding energy E_1 ; other sites have a smaller binding energy E_0 . In a simple model, we can assume that all regulator proteins are bound to DNA, either to the specific site or to one of the many unspecific sites. In an equilibrium ensemble, the occupation probability at the specific site depends on the total number of protein molecules. For the calculation, we first assume that n protein molecules are distributed over N unspecific sites: there are $\binom{N}{n} = N!/(n!(N-n)!)$ possible microstates, each representing one pattern of proteins bound to the DNA. Each microstate has a Boltzmann weight $\exp(-\beta n E_0)$ with $\beta = (k_B T)^{-1}$, Boltzmann's constant k_B , and temperature T . To compute a probability for the specific site to be bound, we compare two macrostates, X_0 and X_1 : in X_0 , the specific site is empty, all n proteins are bound nonspecifically, and the free energy is nE_0 . In state X_1 , the specific site is occupied and $n-1$ proteins are bound nonspecifically, so the total free energy is $(n-1)E_0 + E_1$. The probability weights for the two states read

$$Z_0 = \binom{N}{n} e^{-n\beta E_0}, \quad Z_1 = \binom{N}{n-1} e^{-(n-1)\beta E_0} e^{-\beta E_1}, \quad (9.11)$$

and the corresponding free energies are given by $F_i = -k_B T \ln Z_i$. The occupation probability for the specific site reads $p_1 = Z_1/(Z_1 + Z_0)$ and for $n \ll N$ we obtain the approximation

$$p_1 \approx \frac{n/N}{n/N + e^{\beta \Delta E}}, \quad (9.12)$$

where $\Delta E = E_1 - E_0$ denotes the additional energy in specific binding. To write this result in terms of concentrations, we introduce the total regulator concentration $x_{\text{tot}} = n/(VN_A)$, where V is the cell volume and N_A is Avogadro's constant. Furthermore, we assume that only a small fraction of regulator molecules is bound specifically, so we can approximate x_{tot} by x , the concentration of nonspecifically bound regulators. With the effective

Example 9.1 Regulation Function for a Single Regulator

To model the occupancy of a single binding site, we consider two states, unbound (0) and bound (1). Microscopically, the two states have occupation probabilities p_0 and p_1 ; macroscopically, we can also consider their concentrations s_0 and s_1 in a cell population. The two concentrations depend on the concentration of free regulator molecules x and on the total concentration of binding sites s_{tot} (number of binding sites per total cell volume in the cell culture). If binding is fast and reversible, with a dissociation constant $K_D = x^{\text{eq}} s_0^{\text{eq}} / s_1^{\text{eq}}$ (where x^{eq} , s_0^{eq} , and s_1^{eq} are equilibrium concentrations), the concentrations of empty and occupied binding sites

$$s_0(x) = \frac{K_D}{x + K_D} s_{\text{tot}}, \quad s_1(x) = \frac{x}{x + K_D} s_{\text{tot}}$$

follow from a simple binding equilibrium. By dividing them by the total concentration s_{tot} , we obtain the stationary probabilities

$$p_0(x) = \frac{K_D}{x + K_D}, \quad p_1(x) = \frac{x}{x + K_D}$$

for single binding sites to be empty or occupied (see Figure 9.16). To translate the occupation probabilities into a gene regulation function, we assume that in each binding state transcription initiation occurs at some fixed rate. The total transcription rate is the weighted average over the rates y_0 and y_1 for the empty and the occupied state, respectively:

$$f(x) = y_0 p_0 + y_1 p_1.$$

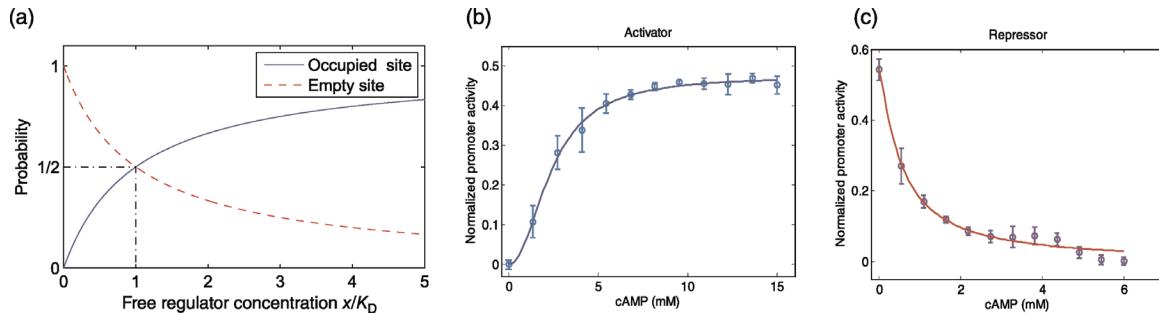


Figure 9.16 Gene regulation function with a single regulator. (a) Occupation probability of a binding site with dissociation constant $K_D = 1$. At low levels x , the function increases almost linearly, while for high levels x , all binding sites are occupied and the curve saturates. The steepness of the curve depends on the dissociation constant K_D . At a regulator concentration $x = K_D$, the binding site is occupied with probability 1/2. (b) Measured gene regulation function between a transcriptional activator (CRP, controlled by the ligand cAMP) and a reporter gene (GFP). (c) If a CRP binding site is inserted close to the polymerase binding sites, CRP can block polymerase binding and shows the typical regulation curve of a repressor. (Experimental curves from Ref. [61].)

This function can be written in two ways:

$$f(x) = y_0 + (y_1 - y_0) \frac{x}{x + K_D}, \quad (9.9)$$

$$f(x) = y_1 + (y_0 - y_1) \frac{K_D}{x + K_D}. \quad (9.10)$$

Formula (9.9) is suitable to describe an activator ($y_1 > y_0$): for small x , the transcription rate starts at the basal value y_0 and rises, according to the higher occupation probability, to the maximal value y_1 . Half-maximal activation is reached at $x = K_D$. For a repressor ($y_0 > y_1$), formula (9.10) describes a decrease from the maximal rate y_1 to a basal rate y_0 . The measured regulation functions for inducers and repressors, shown in Figure 9.16, confirm these formulas.

dissociation constant $K_D = N e^{\beta \Delta E} / (VN_A)$, we reobtain the occupation probability

$$p_1(x) = \frac{x}{x + K_D}. \quad (9.13)$$

Example 9.2 Regulation Functions with Several Regulators

The statistical approach applies also to complex promoter models including several binding sites and regulator types and explicitly modeled RNA polymerase binding. In a general scheme for gene regulation functions, Bintu *et al.* [57] assume that RNA polymerase and regulators can bind cooperatively to different binding sites and that the regulator's binding energy can depend on the binding of other regulators. The transcription rate is assumed to be proportional to the total probability of states in which RNA polymerase is bound. Using the same statistical approach as above, the promoter occupancy for RNA polymerase can be written as

$$p_1 = \left(1 + \frac{N e^{\beta \Delta E_p}}{f_{\text{reg}} n_p} \right)^{-1} \quad (9.14)$$

for the case of N unspecific binding sites for polymerase, n_p polymerase molecules, and an energy difference ΔE_p for specific polymerase binding. The influence of other regulators is summarized in a regulation factor f_{reg} , which changes the effective polymerase concentration. For a simple repressor (n_R molecules, N unspecific binding sites, and energy difference ΔE_R), this regulation factor reads

$$f_{\text{reg}} = \left(1 + \frac{n_R e^{\beta \Delta E_R}}{N} \right)^{-1}. \quad (9.15)$$

Similarly to Eq. (9.13), it can be rewritten as $f_{\text{reg}} = K_R / (K_R + s_R)$ with an inhibition constant K_R , where s_R is the concentration of active repressor. Regulation factors for various combinations of activators or repressors are given in Refs. [57,58]. The resulting gene regulation functions are rational functions such as

$$f(x) = \frac{1 + a_1 x + a_2 x^2}{1 + b_1 x + b_2 x^2} y_0 \quad (9.16)$$

for a regulator X with two binding sites, or

$$f(x_1, x_2) = \frac{1 + a_1 x_1 + a_2 x_2}{1 + b_1 x_1 + b_2 x_2} y_0 \quad (9.17)$$

for two activators X_1 and X_2 . The positive parameters a_1 , a_2 , b_1 , b_2 , and y_0 in such gene regulation functions depend on binding energies and state-dependent transcription rates. In practice, the binding energies are hard to determine, but the function parameters can be fitted to expression data. For a direct fit, inputs and output values need to be known.

9.3.4

Gene Regulation Function of the Lac Promoter

Setty *et al.* [54,62] have determined the regulation function of the Lac operon in living *E. coli* cells [54,62]. In their experiment, transcription rates were measured by a fluorescent reporter protein (GFP) under the control of the Lac promoter (see Section 14.15). The regulator activities were controlled via extracellular levels of cAMP and IPTG (see Figure 9.13b). The transcription rate, plotted against logarithmic concentrations of extracellular cAMP and IPTG, shows four plateaus, corresponding to possible combinations of low and high regulator activities (Figure 9.17a).

As expected, high cAMP and IPTG levels – mimicking low glucose and high lactose concentrations – lead to high Lac expression. In contrast, the expression for low cAMP and low IPTG levels does not vanish and there is always some positive baseline activity. Its function becomes clear if we think of the regulation of the Lac system in the cell: to actually induce the Lac system, some lactose must be able to enter the cell to produce the messenger allolactose. Therefore, some of the lactose transporter LacY must always be present, even at zero external lactose levels, to keep the system responsive. This gives rise to a positive feedback loop between intracellular lactose and the expression of the lactose transporter, creating a bistable system for lactose uptake [55].

To obtain the regulation function in Figure 9.17a, Setty *et al.* assumed a simplified scheme as in Figure 9.14, which leads to gene regulation functions of the form

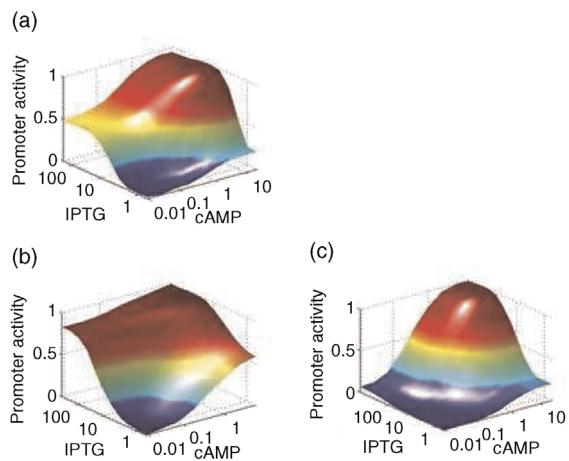


Figure 9.17 Gene regulation functions of the Lac operon (wild type and variants resulting from mutated promoter sequences). (a) Gene regulation function, fitted to promoter activities measured in wild-type *E. coli*. (b) An *E. coli* strain with point mutations in the Lac promoter shows an OR-like regulation function. (c) Another *E. coli* strain shows an AND-like regulation function. (From Ref. [62].)

Q1

shown in Figure 9.17(a). Taking cAMP and IPTG as proxies for transcriptional activators (where inhibition of the repressor LacI counts as an effective activation), the function parameters were determined by fitting the function to measured transcription rates.

According to the microscopic model, the shape of regulation functions depends on the energies of the binding states. How would this function change if the binding energies were changed by mutations in the promoter sequence? For the Lac operon [62], some mutants show much clearer logical AND and OR functions (see Figure 9.17). This suggests that the wild-type Lac function with its different plateaus does not arise from mechanistic constraints, but may stem from an optimization. The example suggests that evolution is relatively free in adapting gene regulation functions, allowing for an evolutionary fine-tuning and optimization of the gene regulatory system.

Kaplan *et al.* [60] have extended the reconstruction of regulation functions to *E. coli* operons coding for sugar utilization proteins (Figure 9.18). For each operon, they measured the response to its cognate sugar and to cAMP (both substances were added to the growth medium). The measured response functions represent not only an

immediate regulation at the gene promoter, but the overall effects of regulation, possibly involving adaptations of the entire cell. Most of the response functions can be factorized into products of functions for the individual inputs, and some of these functions are nonmonotonic. In the case of the galactose utilization genes, nonmonotonic behavior results from an incoherent feed-forward loop (see Section 8.2) formed by CRP, the transcription factor GalS, and the gal genes, as shown by deletion experiments [63].

9.3.5

Inferring Transcription Factor Activities from Transcription Data

To fit gene regulation functions to data, the regulator activities $x(t)$ and the transcription rate $y(t)$ of the target gene must be given. Measuring transcription factor activities directly is difficult, so in the cases described above, other substances were used instead as effective inputs. However, how can regulation functions be inferred if only the expression values of target genes, but none of the input values are known?

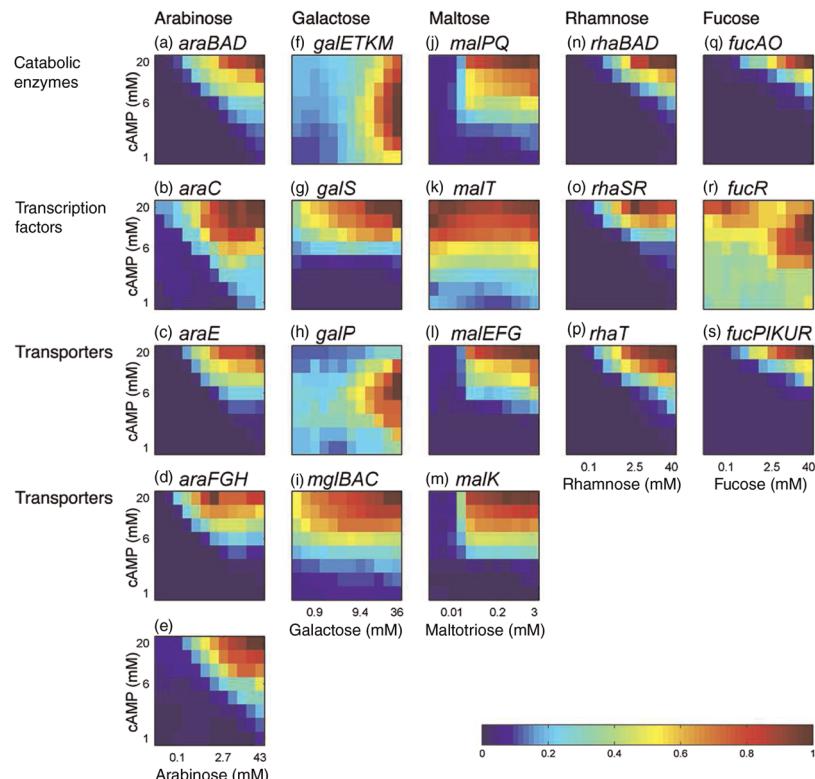


Figure 9.18 Response of sugar utilization genes in *E. coli* bacteria to extracellular levels of sugars and cAMP. The measured regulation functions belong to the genes shown in Figure 8.6. Each panel column corresponds to one sugar and shows the genes coding for enzymes, the transcription factor, and sugar transporters. (From Ref. [60].)

There are several possibilities. First, data analysis methods such as clustering, independent component analysis, or biclustering can be used to see which genes are coregulated under certain conditions. However, since these methods ignore the known wiring between transcription factors and target genes, they cannot distinguish between mechanistic and functional reasons for coexpression and are better suited for exploration than for mechanistic modeling. Second, one can relate the expression of genes

Example 9.3 SOS System in *E. coli*

Transcription factors that act as the only input for a number of genes are called master regulators. This regulation structure, called *single-input motif*, is common in transcription networks and is exemplified by the *SOS system* in *E. coli*, a machinery that senses and repairs DNA damage. The *SOS* system comprises 30 operons, which are usually repressed by the master regulator LexA. When the protein RecA senses DNA damage, it induces autocleavage of LexA, which then derepresses the *SOS* operons. When the damaged DNA has been repaired, LexA starts to repress the *SOS* system again.

Ronen *et al.* [64] have measured the transcription rates of eight of the *SOS* operons after a derepression by LexA. The transcription rates were measured with fluorescent reporter proteins, and temporal transcription rates (in arbitrary units)

$$y(t) = \frac{ds_{\text{GFP}}(t)/dt}{s_{\text{OD}}(t)} \quad (9.18)$$

were computed from the measured fluorescence $s_{\text{GFP}}(t)$ and optical density $s_{\text{OD}}(t)$, a measure of cell count. For justifying the formula (9.18), one needs to assume that the reporter protein is stable and that the turnover of mRNA is fast and unregulated. In the experiment, transcription rates were measured in a time window of 1.5 h after irradiation by UV light. In the model, all operons (denoted by subscripts i) have the same regulation function

$$y_i(t) = \frac{y_i^{\max}}{1 + (x(t)/K_i)^{h_i}}, \quad (9.19)$$

but with different parameters (maximal transcription rates y_i^{\max} , inhibition constants K_i , and Hill coefficients h_i). These parameters as well as the time-dependent activities $x(t)$ of LexA were estimated by a least-squares fit. The absolute scaling of $x(t)$ and K_i is not identifiable because a rescaling $x \rightarrow \lambda x$, $K_i \rightarrow \lambda K_i$ would leave the model's predictions unchanged. Therefore, the initial repressor concentration was set to an arbitrary value $x(0) = 1$. The Hill coefficients were set to $h_i = 1$ (corresponding to Eq. (9.10)) because allowing for deviations $h_i \neq 1$ did not significantly improve the fit (for numerical model details, see Section 14.15).

to the expression of their transcription factors. The problem here is that a transcription factor's activity does not directly depend on its expression value, but rather on protein abundance, cellular localization, and ligand binding, which are hard to control or measure. A promising third possibility is to infer transcription factor activities and gene regulation functions simultaneously from the expression of target genes. Regulation functions for a number of genes have to be inferred in this way, including the *SOS* system [64] and flagella system [59] in *E. coli* and the target genes of the mammalian cell cycle regulator p53 [65].

9.3.5.1 Global Regulation by Transcription Resources

Unlike microarrays, fluorescent reporter proteins permit to measure transcription rates in absolute terms. Such data have revealed a large variation in overall transcription rates, suggesting a strong impact of unspecific global factors such as RNA polymerase on gene expression. In a study of 900 *S. cerevisiae* and 1800 *E. coli* promoters [66], a majority of gene expression levels showed proportional variation, where the common scaling factor depended mainly on the cell's growth rate. Specific regulation adds to this behavior, but plays only a minor role in many genes. The observed expression changes of unregulated genes agree with a simple model of resource allocation: if a cell replicates twice as fast, the protein production rate and thus necessary resources such as RNA polymerase or nucleotides must be roughly twice as large. Genes that are specifically regulated obtain different fractions of the total resource available. Therefore, the expression of non-regulated genes should be proportional to

$$\frac{1}{T} \frac{p_{\text{global}}}{p_{\text{global}} + p_{\text{specific}}}, \quad (9.20)$$

where T is the cell doubling time and p_{global} and p_{specific} are the sums of promoter activities for globally regulated and specifically regulated genes, respectively. A similar approach was used in Ref. [67] to model the regulation of individual genes in the arginine synthesis pathway in *E. coli*. Here, the growth rate μ was directly included into the gene regulation functions

$$\nu = \nu^{\max} \frac{\mu/K_m}{1 + \mu/K_m + [R]/K_r} \quad (9.21)$$

as a proxy for available transcription resources. $[R]$ is the activity of the arginine repressor ArgR and ν^{\max} , K_m , and K_r are the promoter-specific parameters. The model was experimentally verified not only for constant growth, but also for dynamic situations with varying growth rates.

9.3.6

Network Component Analysis

Microarray data, due to their normalization, highlight the effects of specific regulation. Many genes are controlled by more than one transcription factors, whose regulatory effects appear entangled in the data. To derive gene regulation functions from such data, the activities of transcription factors and their effects on individual genes must be estimated in parallel. A method suitable for large networks is network component analysis (NCA), which translates a given network into a quantitative model of gene regulation with simple linear input functions [68,69].

To justify this model, we assume gene regulation functions with a power law form $v_i(\mathbf{c}) \sim \prod_l c_l^{a_{il}}$. If expression levels and transcription factor activities are described by logarithmic values, we obtain a linear model

$$y_i(t) = \sum_l a_{il} x_l(t). \quad (9.22)$$

A logarithmic promoter activity $y_i(t)$ is thus a weighted sum of logarithmic transcription factor activities $x_l(t)$. The index t refers to different samples, for instance, the time points in an experimental time series. The input weights a_{il} indicate whether regulators act as activators ($a_{il} > 0$), as repressors ($a_{il} < 0$), or have no effect on the promoter activity ($a_{il} = 0$). The regulation arrows can be obtained from databases [50,70] or experiments [71]. Given the network structure, many of the coefficients a_{il} can be set to zero values, and known modes of regulation (activation/repression) will restrict the signs of the non-zero values a_{il} . The linear NCA model (9.22) resembles statistical models used in principal component analysis [72] or independent component analysis [73], but in contrast to these methods, it includes biological knowledge about the structure of the genetic network.

To estimate the model parameters (i.e., the nonzero weights a_{il} and the regulator activities $x_l(t)$), we rewrite Eq. (9.22) as a matrix product

$$\mathbf{Y} = \mathbf{AX}, \quad (9.23)$$

where columns in \mathbf{Y} and \mathbf{X} refer to time points or conditions. The sparsity structure of \mathbf{A} (positions and possibly signs of nonzero entries) is defined by the network structure, and only the magnitudes and possibly signs of its nonzero entries are determined from data. To estimate both matrices from measured logarithmic expression levels $y_i^{\text{exp}}(t)$, we require that

$$\mathbf{Y}^{\text{exp}} \approx \mathbf{AX} \quad (9.24)$$

with least-squares errors. Given a data matrix \mathbf{Y}^{exp} and the constraints on \mathbf{A} , the matrices \mathbf{A} and \mathbf{X} can be determined by an iterative optimization: \mathbf{A} is initialized with random values and \mathbf{X} is estimated by the method of least squares; then the resulting \mathbf{X} is kept fixed and \mathbf{A} is updated. This process is iterated until convergence. In the end, the linear NCA model is translated back into power law functions for gene regulation. The functions describe a multiplicative processing of different inputs, but do not account for saturation or interaction effects.

For ideal data (artificial data obtained from a noise-free NCA model), the biquadratic optimization in NCA converges to a global optimum for \mathbf{A} and \mathbf{X} . However, this optimum may be nonunique: for certain network topologies, different parameter choices lead to equally good results, so the NCA model may be nonidentifiable. As an example, consider two regulators that control only a single gene. From the expression profile of this target gene, we can determine some linear combination of the regulators' activities, but not their individual profiles. Whether or not an NCA model is identifiable depends on its network structure and can be tested [68].

Example 9.4 The Transcription Network of *Bacillus subtilis*

NCA has been used to translate the transcription network of *B. subtilis* into a genome-scale model of transcriptional regulation (Figure 9.19) [74]. In the experiments, bacteria were first grown on glucose, and malate was then added as a second carbon source, and vice versa. The expression patterns during both metabolic shifts were used to estimate the influence weights a_{il} between 158 transcription factors and 1754 genes. The signs of some edges (activation or repression), initially unknown, were determined during NCA. The model explains most of the expression data; half of the edges contributed little to the data fit, suggesting that they play a minor role in this process. Among the transcription factors, activity changes are most prominent in regulators of central metabolism. Their activities and regulatory influences, shown in Figure 9.20, complement the metabolic regulation architecture shown in Figure 9.19.

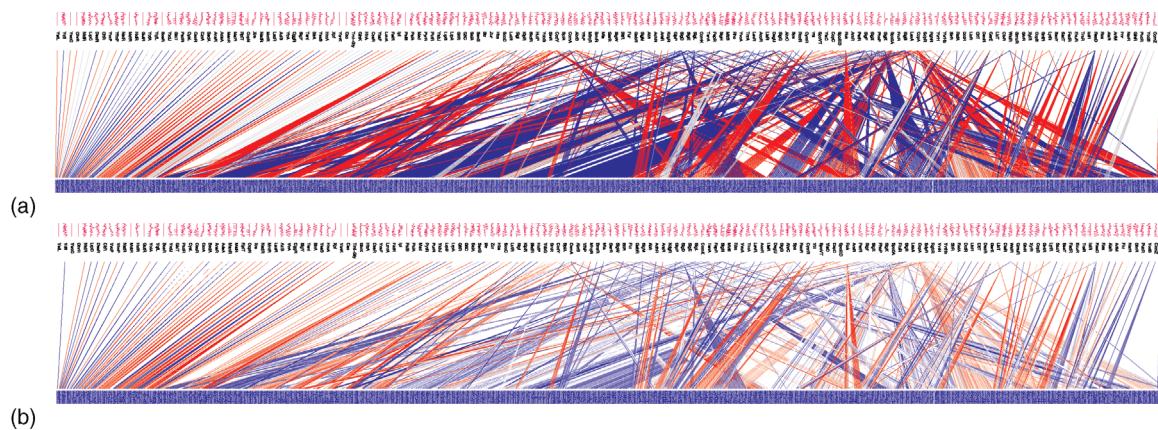


Figure 9.19 Model of transcriptional regulation in *B. subtilis*. (a) The transcription network contains 158 transcription factors (top) and 1754 target genes (bottom), connected by 2900 regulation arrows. Colors indicate activation (blue), repression (red), or unknown modes of action (gray). The network was built based on literature data and ChIP/chip experiments. (b) Influence weights were determined by network component analysis (shaded colors denote weaker influences). The expression data used stem from metabolic shift experiments. (From Ref. [74].)

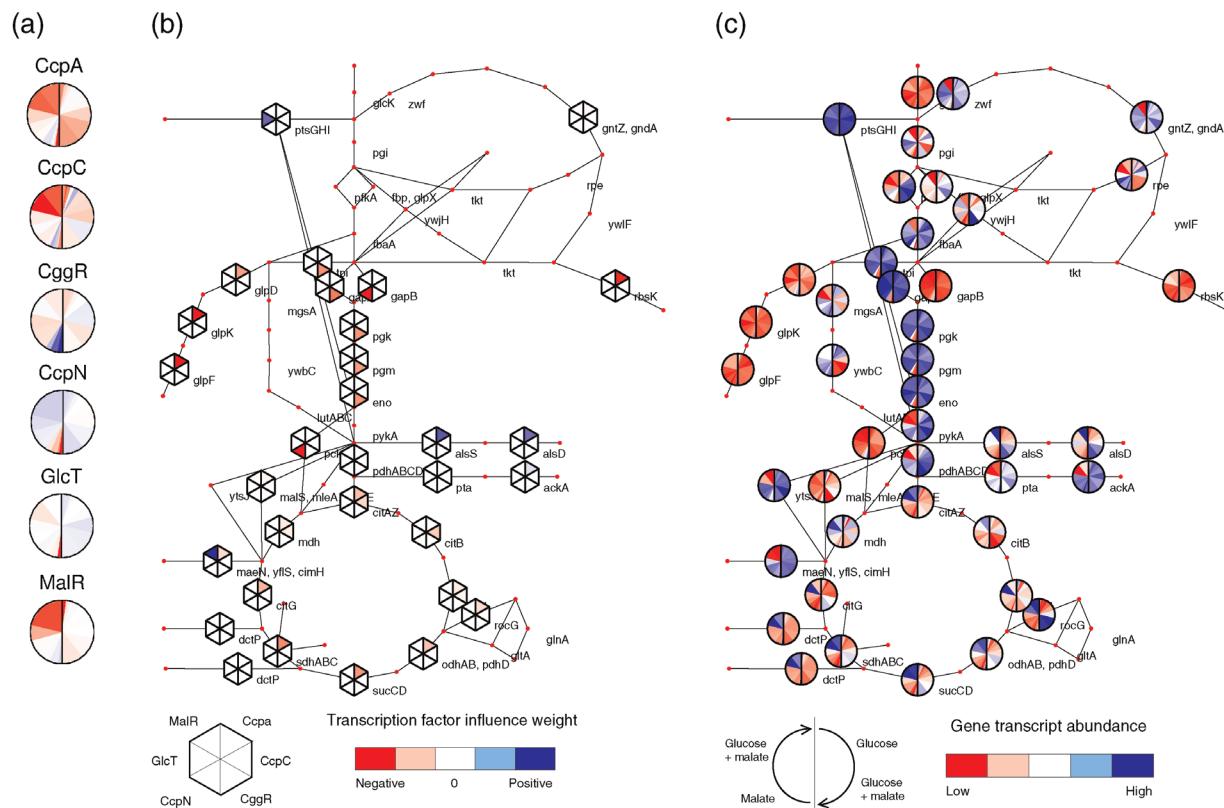


Figure 9.20 Transcriptional regulation of central metabolism in *B. subtilis*. (a) Transcription factor activities during two metabolic shifts (addition of malate to glucose-grown cultures and vice versa). Transcription factor activities were determined by network component analysis (same data as in Figure 9.19) and are shown in two halves of each circle (with time running in clockwise direction; red: negative; blue: positive). During growth on malate, the master regulator CggR represses lower glycolysis (blue segment in CggR circle). After glucose has been added to the culture, the repression is relieved and CcpN becomes active. CcpN suppresses gluconeogenesis via glyceraldehyde-3-phosphate dehydrogenase (encoded by the gene gapB). (b) Influences of the transcription factors CcpA, CcpC, CggR, CcpN, GlcT, and MalR on transcription of enzymes in central metabolism (compare with Figure 9.19). (c) Expression of metabolic enzymes during both metabolic shifts. (Redrawn from Ref. [74].)

9.3.7

Correspondences between mRNA and Protein Levels

Since mRNA levels are relatively easy to measure, they are commonly used as proxies for protein abundance even if empirical correlations between mRNA and protein levels are often low. Similarly, if we use Eq. (9.18) to infer transcription rates from the fluorescence of reporter proteins, we need to assume that mRNA production, mRNA level, and protein production are basically proportional. To see whether such assumptions are justified, let us now have a closer look at the relationship between mRNA and protein levels. Consider a simple model of mRNA production and degradation. The concentration s_i of an mRNA species follows the rate equation

$$\frac{ds_i(t)}{dt} = y_i(t) - \beta_i s_i(t) \quad (9.25)$$

with synthesis rate y_i and an effective degradation constant β_i , possibly capturing mRNA dilution. If mRNA turnover is unregulated and fast (large, constant β_i), we can expect an instantaneous quasi-steady state $\frac{ds_i}{dt} = 0$ in which the mRNA concentration $s_i(t) = y_i(t)/\beta_i$ is proportional to the transcription rate.

In reality, however, this proportionality may not hold. First, with an mRNA lifetime on the order of minutes [75], mRNA dynamics may not be much faster than some regulation events in metabolism. Second, mRNA degradation is regulated dynamically; even if two mRNA species share the same transcription rate, different degradation rates can alter their concentration profiles considerably (see Figure 9.21). To account for such effects in models, degradation rates β_i may be estimated together with the gene regulation functions [65].

From Eq. (9.25), we can obtain two limiting cases: mRNA production can be proportional to mRNA levels (in cases of fast degradation) or proportional to their time derivatives (if degradation can be neglected). In general, the mRNA production rate will be a linear combination of both. Protein production can be described analogously. Assuming that translation rates are proportional to mRNA levels, one obtains two limiting approximations: (i) mRNA levels are proportional to protein levels (in cases of fast protein turnover or in steady state) and (ii) mRNA levels are proportional to temporal changes of protein levels (if protein turnover is much slower than the dynamics studied). The latter possibility can explain some observed discrepancies between mRNA and protein levels, including the fact that strong changes in mRNA levels may translate into mild changes in protein abundance.

9.4

Fluctuations in Gene Expression

Summary

Protein levels vary both in time and between cells in a cell population. Due to random fluctuations in transcription and translation, protein expression can show temporal bursts, leading to typical time correlations and non-Poisson distributions of protein molecule numbers. Variability in cell variables consists of two contributions: intrinsic variability arises within a process under study, for example, gene transcription, while extrinsic variation is imported from other processes. In the case of protein expression, the two contributions can be quantified by analyzing the expression of fluorescent reporter proteins

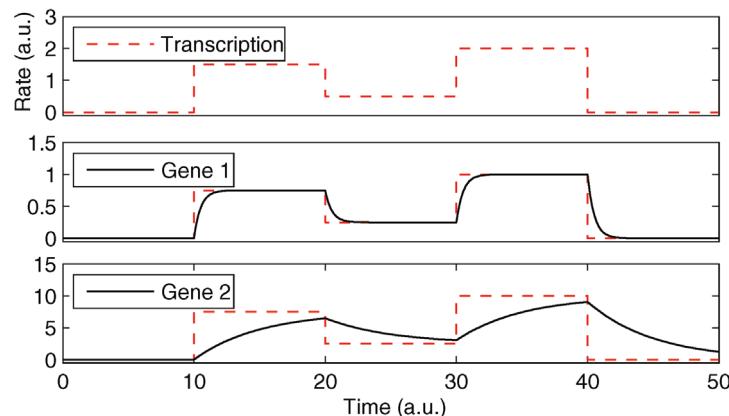


Figure 9.21 Influence of mRNA degradation on expression profiles. In a hypothetical model, genes 1 and 2 share the same transcription rate (top). Center: expression profile of gene 1. Due to fast turnover ($\beta_1 = 2$, in arbitrary units), the expression levels rapidly approach the steady-state value defined by the momentary synthesis rate (dashed lines). Bottom: gene 2 shares the same synthesis rate, but is degraded more slowly ($\beta_2 = 0.2$), and its temporal changes are less sharp.

in individual cells. Randomness in gene expression enables cells to generate diversity, for example, subpopulations in bacterial cultures.

Cells behave differently even if they are genetically identical and grown under the same conditions (see Figure 9.22). Variability in protein levels, metabolic state, and cell morphology ensues from internal factors such as cell cycle phase and from factors in the environment (e.g., nutrients, temperature, and cell density). However, cells can also generate spontaneous random behavior. Fluctuations in gene expression, caused by molecular noise, enable cells to switch randomly between different states, thus creating diversity in cell populations.

Random behavior can emerge from microscopic processes such as gene transcription that involve small molecule numbers. If molecules act as catalysts (as mRNA does in translation), fluctuations in their abundance will influence downstream processes, propagate through biochemical networks, and contribute to the total variability. Generally, we can distinguish between *intrinsic noise*, generated within a system, and *extrinsic noise* enforced from the outside. Transcription, for instance, is both intrinsically stochastic and dependent on fluctuating transcription factor activities, which contribute additional extrinsic noise.

In this section, we shall study stochastic variability in gene expression and see how it can be quantified using single-cell measurements (see Section 14.16). Flow cytometry reveals the distribution of protein levels in a cell population, while temporal fluctuations within single cells can be measured by fluorescence microscopy (see Figure 9.22). Treating cells as different realizations of one random process, we can model fluctuations in gene expression by stochastic processes as described in Section 7.2. Such models have been validated in experiments with artificial genetic circuits [76–79].

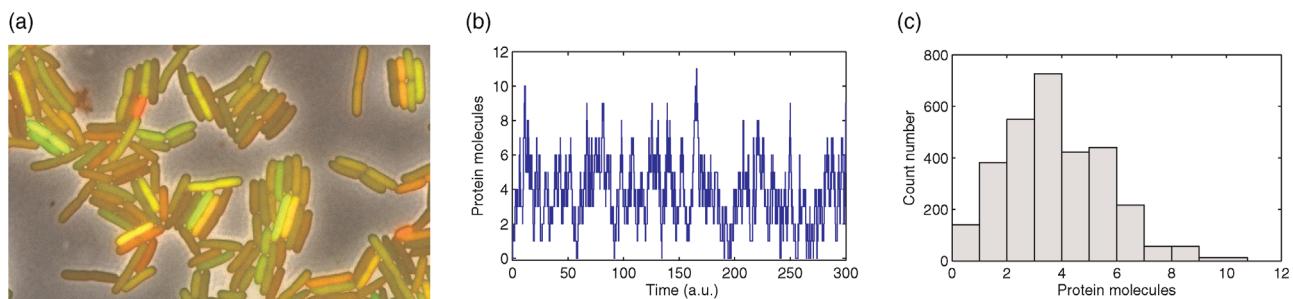


Figure 9.22 Variability in protein levels. (a) *E. coli* bacteria expressing fluorescent proteins (see Section 14.15). With two proteins, shown in red and green, total brightness reflects the sum of both protein levels. Depending on their expression ratios, cells appear in red, yellow, or green. Different expression levels are partly caused by random events. (Courtesy of M. Elowitz.) (b) Temporal changes of molecule numbers, obtained from simulating a birth-death process with constant translation rate (see Section 7.2.1; parameters $w_+ = 2$ and $w_- = 0.5$). (c) Histogram of molecule numbers sampled over time. For an infinite time interval, the numbers would follow a Poisson distribution.

9.4.1

Stochastic Model of Transcription and Translation

If protein production and degradation acted as a simple birth-death process (see Section 7.2.1), the molecule numbers x would follow a Poisson distribution (see Figure 9.22c). In Poisson distributions, mean value and variance are identical. At larger mean molecule numbers $\langle x \rangle$, the absolute standard deviation σ_x becomes larger (since $\sigma_x = \sqrt{\langle x \rangle}$), but the relative standard deviation $\eta_x = \sigma_x/\langle x \rangle$ becomes smaller ($\sigma_x/\langle x \rangle = \sqrt{1/\langle x \rangle}$). In reality, protein levels emerge from a series of processes – transcription factor activation, chromatin remodeling, transcription initiation, mRNA splicing, translation, and the degradation of mRNA and proteins – which can all be treated as random processes. This leads to non-Poisson distributions. To keep things simple, we consider only two processes, transcription and translation, as in the model in Section 7.2.6.

9.4.1.1 Macroscopic Kinetic Model

We consider a deterministic model of gene expression describing the mRNA and protein levels in a cell population (following the scheme in Figure 9.24). Molecule species are described by real-valued concentrations, and their production and degradation follow kinetic rate laws. Specifically, we consider a gene that is transcribed at a rate k_{+x} while translation occurs with a fixed rate k_{+y} per mRNA molecule. The kinetic model (with mRNA concentration s_x , protein concentration s_y , and degradation rates k_{-x} and k_{-y}) reads

$$\begin{aligned} \frac{ds_x}{dt} &= k_{+x} - k_{-x}s_x, \\ \frac{ds_y}{dt} &= k_{+y}s_x - k_{-y}s_y. \end{aligned} \quad (9.26)$$

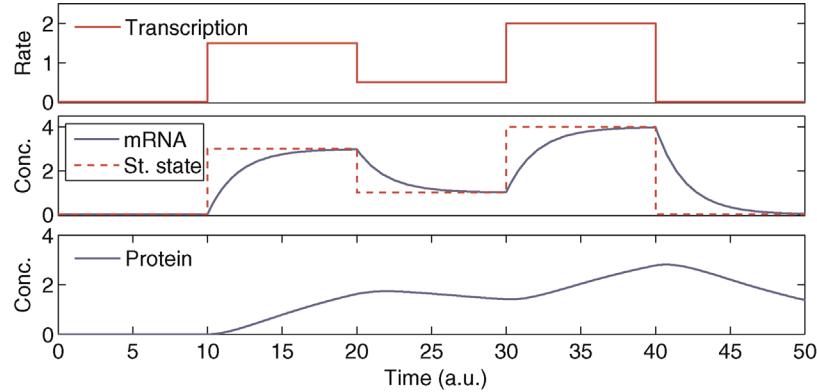


Figure 9.23 Kinetic model of transcription and translation (compare with Figure 9.24, parameters $k_{+x} = k_{-x} = 0.5$ and $k_{+y} = k_{-y} = 0.1$). The curves show a simulation with a predefined step-like time profile for the transcription rate k_{+x} (top, arbitrary units). After each step, the mRNA level (center) approaches its current steady-state value (dashed). The protein level (bottom) follows more slowly.

The transcription rate k_{+x} may depend on the presence of transcription factors. With a constant rate k_{+x} (constitutive expression), the concentrations converge to their stationary values

$$s_x^{\text{st}} = k_{+x}/k_{-x}, \quad s_y^{\text{st}} = s_x^{\text{st}} k_{+y}/k_{-y}. \quad (9.27)$$

Figure 9.23 shows a predefined transcription rate $k_{+x}(t)$ and the resulting temporal concentration profiles. After each jump in the transcription rate, the mRNA level relaxes exponentially toward a steady state and the protein level follows more slowly.

9.4.1.2 Microscopic Stochastic Model

On a microscopic scale, transcription and translation are random processes between single molecules. The actual molecule numbers in a cell may deviate strongly from the population average, and random fluctuations of the mRNA number may lead to strongly increased protein production (“bursts”) in certain cells. All this remains invisible in the kinetic population model. In a stochastic model, we describe the molecule numbers by coupled birth–death processes for mRNA and protein amounts (Figure 9.24). Realizations of this process can be depicted as walks on a two-dimensional grid (Figure 9.25) with transition rates given by

Process	Transition	Rate
Transcription	$x \rightarrow x + 1$	$a_x^+ = w_{+x}$
mRNA degradation	$x \rightarrow x - 1$	$a_x^- = w_{-x}x$
Translation	$y \rightarrow y + 1$	$a_y^+ = w_{+y}x$
Protein degradation	$y \rightarrow y - 1$	$a_y^- = w_{-y}y$

The mRNA molecule number x is determined by a simple birth–death process; in the stationary ensemble, it follows a Poisson distribution. The protein production rate depends

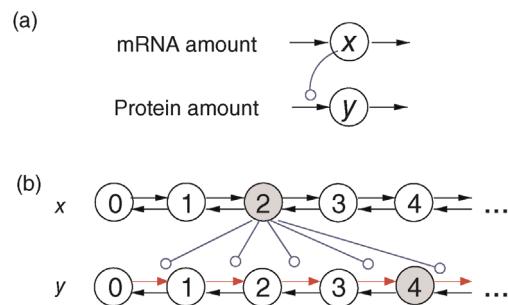


Figure 9.24 Stochastic model of transcription and translation. (a) Transcription and translation are modeled by a coupled birth–death process for the molecule numbers of mRNA and protein. The translation rate depends on mRNA abundance. (b) The two random processes are coupled. Current molecule numbers of mRNA and protein are marked by circles. One momentary state (two mRNA molecules and four protein molecules) is shown in gray. The rate a_y^+ for protein synthesis (red arrows) depends on the number of mRNA molecules present.

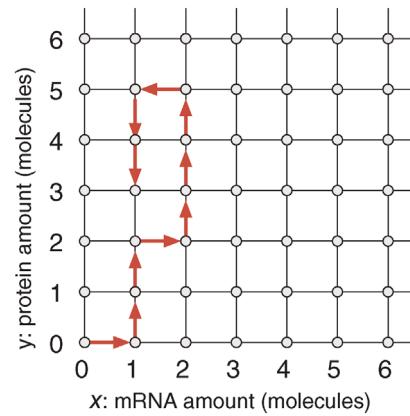


Figure 9.25 State-space trajectory in the model of transcription and translation (Figure 9.24). The state space consists of pairs (x, y) of mRNA numbers x and protein numbers y . A realization of the process is shown as a path (transitions shown by arrows, time not shown).

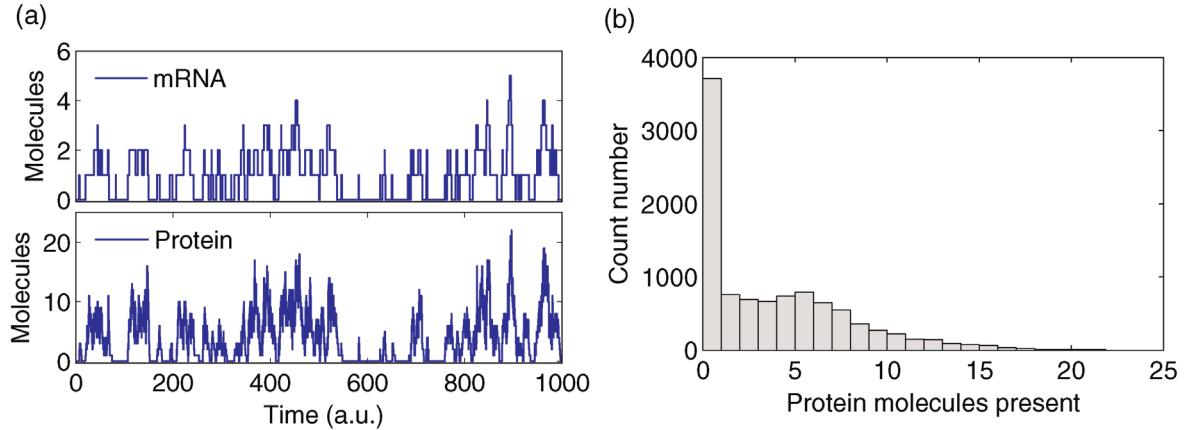


Figure 9.26 Stochastic dynamics of transcription and translation (model from Figure 9.24 with parameters $w_{+x} = w_{-x} = 0.1$, $w_{+y} = 2$, and $w_{-y} = 0.5$). (a) Simulated time series for mRNA and protein molecule numbers. In moments of high mRNA abundance, there can be bursts of protein expression. (b) Due to the protein bursts, the histogram of protein numbers differs from a Poisson distribution.

on the current amount of mRNA molecules present (Figure 9.26), so the protein level y shows a more complicated dynamics than the mRNA level. Models of this type can also be used to describe other two-level processes [80] and regulatory circuits consisting of several genes [81].

The stationary distribution of the process (9.28) can be computed with the help of moment-generating functions [81] or by solving the chemical Langevin equation. The mean values

$$\langle x \rangle = \frac{w_{+x}}{w_{-x}}, \quad \langle y \rangle = \langle x \rangle \frac{w_{+y}}{w_{-y}} \quad (9.29)$$

resemble the deterministic result in Eq. (9.27). The variances are determined by

$$\frac{\text{var}(x)}{\langle x \rangle} = 1, \quad \frac{\text{var}(y)}{\langle y \rangle} = 1 + \frac{w_{+y}}{w_{-x} + w_{-y}}. \quad (9.30)$$

The ratio between variance and average value, called *Fano factor*, has a value of 1 for the mRNA molecules, as expected for a Poisson distribution. For proteins, it is larger than 1, indicating that the distribution has a broader shape than a Poisson distribution. According to Eqs. (9.29) and (9.30), the mean value and variance of y can be set by tuning the rates w_{+x} , w_{-x} , w_{+y} , and w_{-y} [81].

In simulations of the random process (9.28), the protein number y shows temporal bursts (see Figure 9.26), which are also observed in experiments [82]. Such bursts arise in periods of high mRNA abundance and lead to a broader, non-Poisson distribution of protein molecule numbers. If the number of mRNA molecules were constant, the protein number would follow a birth-death process with constant translation rate $w_{+y}x$, leading to a Poisson distribution. If the mRNA number varies, but remains constant during longer time intervals, the overall distribution of protein numbers will be a mixture of Poisson distributions, giving rise to its broader shape.

9.4.1.3 Fluctuations in a Genetic Network

Linear propagation of expression noise can also be modeled for complex genetic networks. To obtain stochastic models model of larger gene regulation networks, we describe all mRNA and protein species by coupled birth-death processes. Given all molecule numbers x_1, \dots, x_M , a reaction event can produce or degrade a single molecule of type i :

Process	Transition	Rate
Production	$x_i \rightarrow x_i + 1$	$a_i^+(\mathbf{x}) = w_i^+(\mathbf{x})$
Degradation	$x_i \rightarrow x_i - 1$	$a_i^-(\mathbf{x}) = x_i w_i^-(\mathbf{x})$

Since mRNA molecules serve as templates for translation and proteins can regulate transcription, both processes are coupled and their propensities $a_{\pm i}$ depend on the entire system state (x_1, \dots, x_M). We can approximate the propensities $a_{\pm i}$ by linear functions, assuming that substances are linearly degraded and the production rates are either constant, proportional to some molecule amount, or given by a linear function of the molecule numbers [81]. For practical reasons, we introduce a virtual substance with constant amount $x_0 = 1$ and propensities $a_i^+ = a_i^- = 0$. We can now write all propensities as

$$a_i^+ = \sum_{l=0, \dots, M} w_{il}^+ x_l, \quad a_i^- = w_{ii}^- x_i \quad (9.32)$$

with a production matrix \mathbf{W}^+ and a diagonal degradation matrix \mathbf{W}^- . The transitions in (9.31) lead to a master equation for the time-dependent probabilities $p(x_1, \dots, x_M; t)$. For each molecule type i , the equation reads

$$\begin{aligned} \frac{d}{dt} p(\dots, x_i, \dots; t) &= E^{-i}(a_i^+(\mathbf{x})p(\mathbf{x}; t)) + E^{+i}(a_i^-(\mathbf{x})p(\mathbf{x}; t)) \\ &\quad - a_i^+(\mathbf{x})p(\mathbf{x}; t) - a_i^-(\mathbf{x})p(\mathbf{x}; t), \end{aligned} \quad (9.33)$$

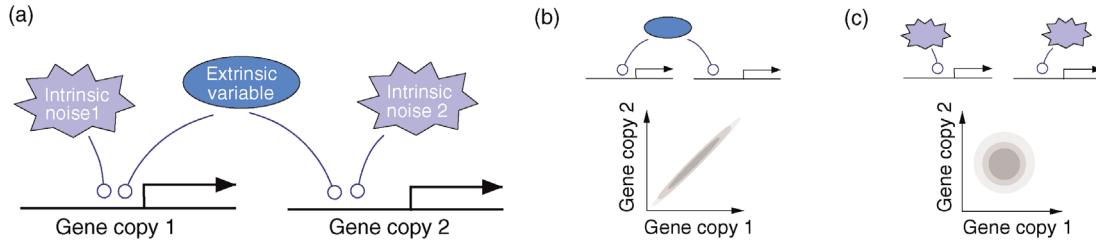


Figure 9.27 Measurement of intrinsic and extrinsic noise. (a) Two reporter genes with identical promoter sequences respond to the same extrinsic variables (e.g., transcription factors). Moreover, the transcription rates show intrinsic fluctuations represented by uncorrelated noise variables. (b) With extrinsic noise only, expression levels are completely correlated and the scatter plot yields a line. (c) Intrinsic noise leads to uncorrelated expression levels.

where the symbol $E^{\pm i}$ (“step operator”) denotes the addition or removal of one molecule of type i . Approximating the molecule numbers by real-valued numbers, the process can be described by a chemical Langevin equation

$$\frac{dx}{dt} = N(\mathbf{W}^+ - \mathbf{W}^-)x + N Dg((\mathbf{W}^+ - \mathbf{W}^-)x)^{1/2} \xi, \quad (9.34)$$

which is fairly easy to simulate (see Section 7.2.4).

9.4.2

Intrinsic and Extrinsic Variability

9.4.2.1 Measurement of Intrinsic and Extrinsic Variability

Is the behavior of cells predetermined by their internal state and their environment, or is there something fundamentally stochastic in it? And if there is randomness, how much does it contribute to differences between cells? To answer this question, one would have to prepare two cells in exactly the same state and expose them to exactly the same environment. This is, of course, impossible. However, Elowitz *et al.* [78] found a way to study randomness in individual genes within one cell. To measure random effects in gene expression, they inserted two fluorescent reporter proteins, CFP and YFP, into the genome of *E. coli* bacteria. The two genes were controlled by identical copies of the Lac promoter (see Section 9.3) and exposed to the same cell state.

In the experiment, the expression levels of both gene copies were recorded in single cells by fluorescence microscopy. In the analysis, differences between measured expression levels are attributed to random behavior (see Figure 9.27). If expression is completely determined by extrinsic factors, the protein levels will be correlated; if intrinsic expression noise dominates, the gene copies will show uncorrelated expression. The actual contributions of intrinsic (uncorrelated) and extrinsic (correlated) noise can be estimated from scatter plots between CFP and YFP intensities (Figure 9.28). The mean expression level $\langle x \rangle$ affects the relative noise levels $\eta_x = \sigma_x / \langle x \rangle$; at high expression levels (induced by a completely derepressed

promoter), the relative cell-to-cell variation was about $\eta_{\text{tot}} = 8\%$, including about $\eta_{\text{int}} = 5\%$ caused by intrinsic noise. At normal expression levels (using a wild-type Lac promoter), which are about 20 times lower, the relative variation reaches about $\eta_{\text{tot}} = 40\%$ in total and $\eta_{\text{int}} = 20\%$ from intrinsic noise. The intrinsic noise decreases with the expression level roughly as $\eta_{\text{int}}^2 \approx c_1 / \langle x \rangle + c_2$ with constants c_1 and c_2 [83]. The first term alone would be expected from a Poisson distribution.

9.4.2.2 Calculation of Intrinsic and Extrinsic Variability

The distinction between extrinsic and intrinsic noise can be generalized [83]. Assume that an output variable y depends on noise variables u_i collected in vectors \mathbf{u}_E

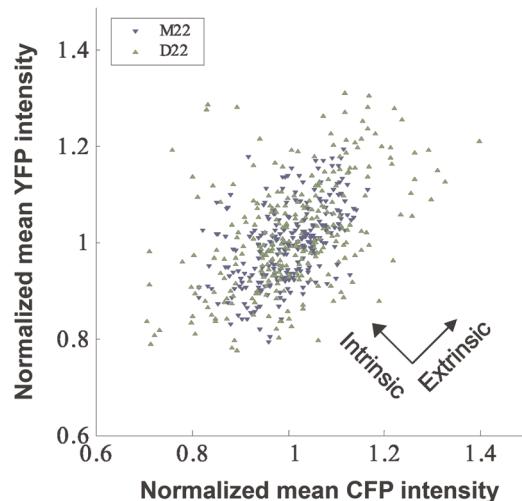


Figure 9.28 Two copies of a gene promoter in the same cell show different, but correlated expression levels (see Figure 9.27). The scatter plot shows levels of the two reporter proteins CFP (x-axis) and YFP (y-axis), controlled by Lac promoters, in a population of *E. coli* bacteria. The cloud shape reflects intrinsic and extrinsic noise in protein production. Data from a “quiet” strain (blue, with constitutive Lac expression, relative noise levels $\eta_{\text{int}} = 5.5 \pm 0.5\%$ and $\eta_{\text{ext}} = 5.1 \pm 0.5\%$) and a “noisy” strain (green, $\eta_{\text{int}} = 10.5 \pm 1\%$ and $\eta_{\text{ext}} = 4.6 \pm 1.2\%$) are shown. (From Ref. [78].)

(extrinsic variables) and $\langle \cdot \rangle_I$ (intrinsic variables). With the symbols $\langle \cdot \rangle_I$ and $\langle \cdot \rangle_E$ for averages over intrinsic or extrinsic noise variables, the variance of y can be written as

$$\text{var}(y) = \langle (y^2) \rangle_E - \langle (y) \rangle_E^2 \\ = \langle (y^2) \rangle_I - \langle (y) \rangle_I^2 + \langle (y) \rangle_I^2 \langle (y^2) \rangle_E - \langle (y) \rangle_E^2. \quad (9.35)$$

Thus, the scaled variance of y can be split into two terms:

$$\frac{\text{var}(y)}{\langle y \rangle^2} = \frac{\langle (y^2) \rangle_I - \langle (y) \rangle_I^2}{\langle y \rangle^2} + \frac{\langle (y) \rangle_E^2 - \langle (y) \rangle^2}{\langle y \rangle^2}, \quad (9.36)$$

or briefly $\eta^2 = \eta_I^2 + \eta_E^2$. This division into intrinsic and extrinsic noise holds for arbitrary probability distributions

Example 9.5 Variability Caused by Cellular Subsystems

The splitting of noise contributions can be extended to entire cellular subsystems. Colman-Lerner *et al.* [84] quantified several sources of variability in a well-studied signaling system, the pheromone response pathway in the yeast *S. cerevisiae*. The pheromone response in cells of mating type "a" is triggered by a pheromone called α -factor: a signaling cascade activates the transcription factor Ste12, which then induces expression changes. In the experiment, a reporter protein induced by Ste12 was used as a readout. After addition of α -factor, the reporter level increases with a temporal slope y , taken to be the observable output. The output depends on both Ste12 activation and the expression machinery.

In a simple model (shown in Figure 9.29a), the output y is given by the product $y = EP$ of a signaling pathway output P (Ste12 activity) and an expression output E (protein production per promoter activity). Each of the factors P and E can be split into a mean value (the "capacity") and fluctuations (the "noise"), so we can set $P = A + a$ and $E = B + b$. In reality, all these terms will be noisy and contribute to the total output variability. The contributions can be distinguished by their different correlations. The capacities A and B vary only between cells, while the noise variables describe independent fluctuations within cells. In particular, the expression noise b of different gene copies will be independent. The pathway capacity may depend on expression capacity (because a large general expression will also increase the expression of the signaling pathway), so A and B are correlated with a linear correlation coefficient ρ_{AB} . Altogether, the relative variance of the output can be written as a sum

$$\eta_y^2 = \eta_A^2 + \eta_B^2 + \eta_a^2 + \eta_b^2 + \rho_{AB}\eta_A\eta_B.$$

Two experiments suffice to measure these contributions (Figure 9.29b). In the first experiment, two identical fluorescent reporters for the Ste12-responsive promoter PRM1 were used. The comparison yields the total variance η_y^2 and the contribution of gene expression noise η_b^2 . In the second experiment, PRM1 expression was compared with expression from the nonresponsive promoter ACT1. Together, the measured variances and correlations allowed for resolving the contributions $\eta_P^2 = \eta_A^2 + \eta_a^2$ and η_B^2 , as well as the term $\rho_{AB}\eta_A\eta_B$. For an α -factor concentration of 20 nM, the total variability after 3 h is $\eta_y^2 = 11.5\%$, with contributions $\eta_b^2 = 0.2\%$, $\eta_P^2 = 2.5\%$, $\eta_B^2 = 8.8\%$, and a correlation coefficient $\rho_{AB} = 0.87$ between pathway and expression capacity. Thus, differences in expression capacity between cells are the main source of variation, while expression noise turned out to be negligible. The variation between cells was relatively constant in time, which suggests that cells already differed before the addition of α -factor. At a lower α -factor concentration of 1.25 nM, intercell differences in the pathway become more important. The relative variance rises to $\eta_P^2 = 7.8\%$, which is counterbalanced by a smaller variance from expression capacity, $\eta_B^2 = 4.8\%$.

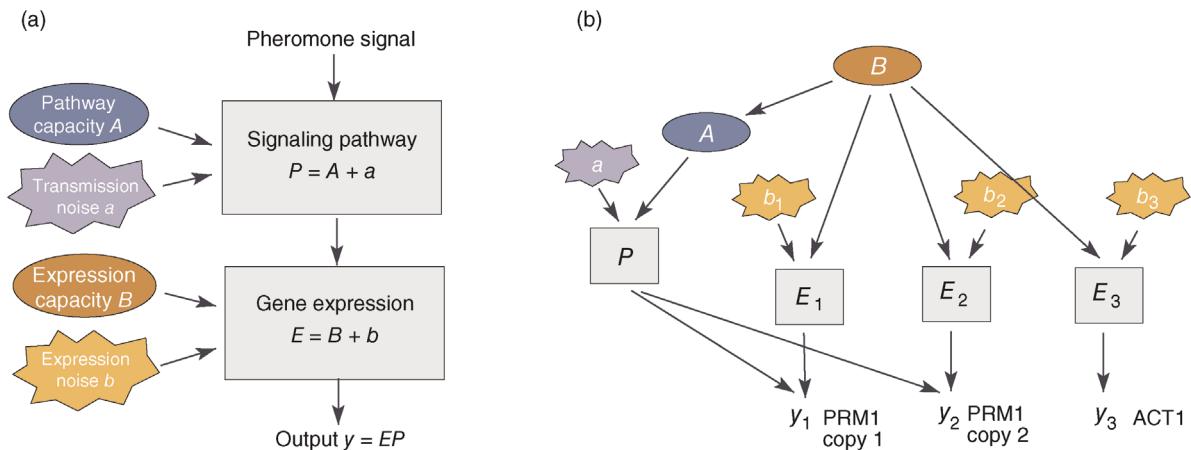


Figure 9.29 Sources of variation in the yeast pheromone system. (a) In a simple model [84], the pheromone response output $y = EP$ is influenced by signaling (P) and expression (E). Each of these contributions results from cell-dependent capacities and random noise. (b) Measurement of noise contributions. By comparing the output from two copies of the PRM1 promoter and the nonresponsive promoter ACT1, one can dissect the different sources of noise.

and arbitrary functions $y(\mathbf{u}_I, \mathbf{u}_E)$. Measurements of expression variability yield $\langle y \rangle$ and $\text{var}(y)$, so η^2 can be directly computed. To split it as in Eq. (9.36), we need to estimate the term $\langle \langle y \rangle_I^2 \rangle_E$ from measurements. This is where the two gene copies come into play: they share the same extrinsic, but not the same intrinsic noise. For given extrinsic variables, the fluctuations of u_I are uncorrelated, so $\langle y_1 y_2 \rangle_I = \langle y_1 \rangle_I \langle y_2 \rangle_I = \langle y \rangle_I^2$. By averaging over the extrinsic variables (i.e., over different cells), we obtain the term $\langle \langle y_1 \rangle_I \langle y_2 \rangle_I \rangle_E$.

9.4.3 Temporal Fluctuations in Gene Cascades

Protein levels do not only vary between cells, but also fluctuate in time. To quantify such fluctuations, we compare the protein level in a cell with the average value in the population: if the level is above average in a certain moment, it will also stay above average for some time. Typical time scales of such fluctuations can be quantified by autocorrelation functions (see Section 15.4). They reflect the dynamics of protein levels, determined by network structure, feedback regulation of protein production, and the effective lifetimes of proteins.

9.4.3.1 Linear Model with Two Genes

If a genetic circuit is perturbed by extrinsic fluctuations, its protein levels will fluctuate and their frequency spectra and time correlations can be computed from a Langevin equation (see Section 7.2.4). As an example, consider a gene X_1 that activates a second gene X_2 ; the synthesis rate of X_1 is subject to time-dependent extrinsic noise. If

transcription and translation are lumped in one step, the expression levels s_1 and s_2 can be described by a stochastic model

$$\begin{aligned}\frac{ds_1}{dt} &= \alpha_1(1 + \xi) - \beta s_1, \\ \frac{ds_2}{dt} &= \alpha_2 s_1 - \beta s_2,\end{aligned}\quad (9.37)$$

which formally resembles the kinetic model in Eq. (9.26). The production rate of X_1 fluctuates with a noise term ξ and we assume equal degradation constants β for both genes. Without noise ($\xi = 0$), the system would show a steady state with concentrations $s_1^{st} = \alpha_1/\beta$ and $s_2^{st} = s_1^{st} \alpha_2/\beta$. The noise ξ , however, leads to deviations $x_1(t) = s_1(t) - s_1^{st}$ and $x_2(t) = s_2(t) - s_2^{st}$ from the steady state, which follow a Langevin equation

$$\frac{d}{dt} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} -\beta & 0 \\ \alpha_2 & -\beta \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} \alpha_1 \\ 0 \end{pmatrix} \xi. \quad (9.38)$$

We can rewrite this equation in the standard form (Eq. (7.23)) with $\mathbf{u} = \xi$, $\mathbf{y} = \mathbf{x}$, and

$$\mathbf{A} = \begin{pmatrix} -\beta & 0 \\ \alpha_2 & -\beta \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} \alpha_1 \\ 0 \end{pmatrix}, \quad \mathbf{C} = \mathbf{I}, \quad \mathbf{D} = 0. \quad (9.39)$$

From these matrices, we can compute the frequency response function $\mathbf{H}(i\omega)$ as well as the spectral densities $\Phi(\omega)$ and the autocorrelation function $R(\tau) = \Phi(\tau)/\Phi(0)$ for both genes (see Section 15.5). If the input ξ is given by white noise, we obtain the spectral density functions (see Section 7.2.5)

$$\Phi_{x_1}(\omega) = \frac{\alpha_1^2}{\beta^2 + \omega^2}, \quad \Phi_{x_2}(\omega) = \frac{\alpha_1^2 \alpha_2^2}{(\beta^2 + \omega^2)^2}. \quad (9.40)$$

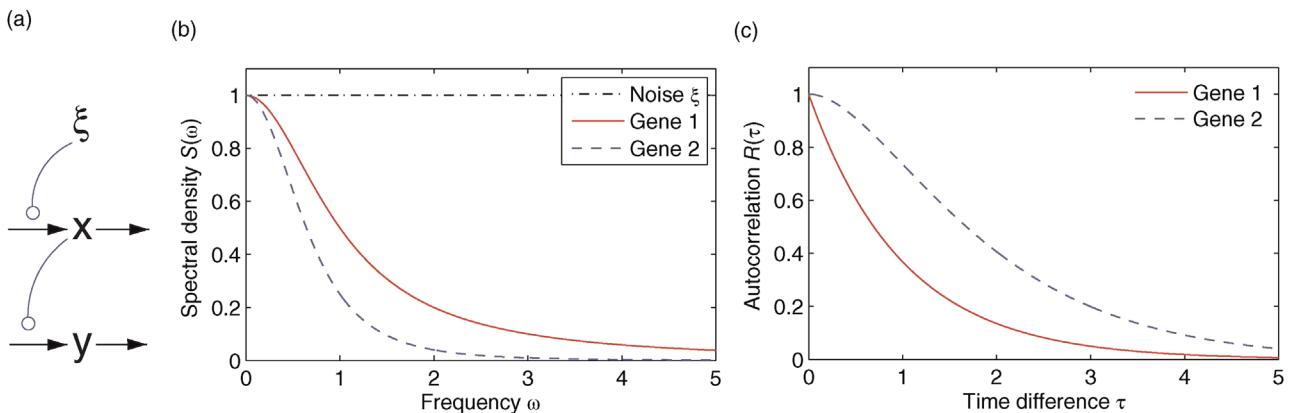


Figure 9.30 Expression fluctuations in a noisy two-gene cascade (Eq. (9.37)) with parameter values $\alpha_1 = \alpha_2 = \beta = 1$. (a) Two-gene cascade with a white noise input ξ . (b) Spectral densities of input noise ξ and gene expression levels. (c) Autocorrelation functions of both genes. As the effects of noise propagate from gene 1 to gene 2, high-frequency fluctuations are filtered out, the typical fluctuations become longer, and the autocorrelation function becomes broader. Autocorrelations are related to spectral densities via a Fourier transformation. All quantities are shown in arbitrary units.

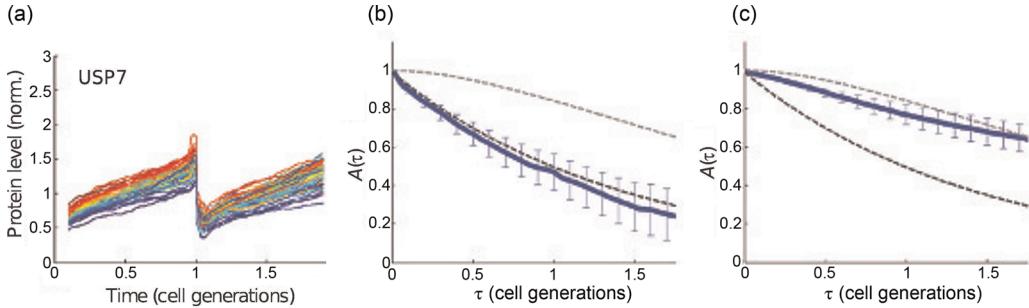


Figure 9.31 Temporal fluctuations in protein levels. (a) Time-dependent protein levels in human cells, measured using fluorescently labeled proteins and time-lapse microscopy. Curves show amounts of the human protease USP7 in different cells (marked by colors) over two cell cycles. The protein content (measured by the fluorescence of YFP-labeled protein) increases during the cell cycle and drops approximately by half during cell division. To make curves comparable, time was rescaled to yield equal cell cycle periods. (b) Rank autocorrelation function ($A(\tau)$) obtained from the curves in (a). Dashed lines show predicted autocorrelations based on different scenarios for protein production noise (compare with Figure 9.30c): uncorrelated white noise (lower dashed line) and time-correlated noise, possibly resulting from uncorrelated noise in upstream processes (upper dashed line). The measured curve resembles the prediction based on white noise. (c) Result for the protein HMGA2. The autocorrelation function suggests that the noise in protein production is already correlated in time. (From Ref. [85].)

An inverse Fourier transformation of the spectral densities yields the covariance functions [85]

$$\Phi_x(\tau) = \frac{a_1^2}{\beta} e^{-\beta|\tau|}, \quad \Phi_y(\tau) = \frac{(\alpha_1 \alpha_2)^2}{4\beta^2} \left(\frac{1}{\beta} + |\tau| \right) e^{-\beta|\tau|} \quad (9.41)$$

and the autocorrelation functions

$$R_{x_1}(\tau) = e^{-\beta|\tau|}, \quad R_{x_2}(\tau) = e^{-\beta|\tau|}(1 + \beta|\tau|). \quad (9.42)$$

Figure 9.30 shows the spectral densities (9.40) and the autocorrelation functions (9.42). White noise ξ has a constant spectral density $\Phi_u(\omega) = 1$ for all frequencies ω . In gene X_1 , fluctuations at higher frequencies become damped, and in gene X_2 , this effect is even more pronounced. In the time domain, the damping at higher frequencies leads to slow fluctuations, as visible from the autocorrelation function. The autocorrelation curves of the two genes are qualitatively different, reflecting the (direct or indirect) way in which output noise emerges from white noise.

9.4.3.2 Measuring the Time Correlations in Protein Levels

In our two-gene model, the time scale of fluctuations is mainly determined by the protein degradation constant β . However, the degradation term need not only describe actual degradation, but may effectively cover other processes. Dilution in growing cells (with cell cycle time $T_{1/2}$) contributes such a term with $\beta = -\ln 2/T_{1/2}$, and if proteins linearly control their own production, we obtain additional terms $\alpha_x^{\text{act}} x$ (for self-induction) or $-\alpha_x^{\text{inh}} x$ (for self-repression). All these terms can be formally incorporated into degradation and will influence the time scale of fluctuations. The time scale of protein fluctuations *in vivo* can be estimated from time-lapse microscopy of single cells [85]. As an example, Figure 9.31b shows measured rank autocorrelations for the human protease USP7 and autocorrelations predicted from Eq. (9.37). The autocorrelations resemble those of our protein X_1 , with white noise fluctuations in production. Another protein, the architectural HMGA2 protein (Figure 9.31c), behaves like protein X_2 with pink (i.e., low frequency) noise in the production rate.

Exercises

Section 9.2

- 1) *Statistical model of binding site occupancy.* Show that the occupation probability $\frac{Z_1}{Z_1+Z_0}$ in Section 9.3.3, with $n \ll N$, yields approximately Eq. (9.12). Use Stirling's approximation formula $n! \approx \sqrt{2\pi n} n^n/e^n$.
- 2) *Effect of mRNA degradation on expression profiles.* Assume that the concentration of mRNA species follows a kinetic model

$$dc(t)/dt = v(t) - \mu c(t)$$

with a degradation rate constant μ . How could the transcription rate $v(t)$ be computed from measurements of $c(t)$? Assume that $c(t)$ is given as a continuous curve and that the measurements are exact. Describe the limiting cases $\mu \rightarrow 0$ and $\mu \rightarrow \infty$. Which difficulties will arise with real-world data?

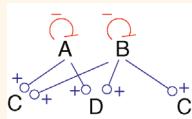
- 3) *Binding of a transcription factor.* Consider a simple model of transcription factor binding: the concentrations of the free transcription factor, empty binding sites, and occupied binding sites are

denoted by x , c_0 , and c_1 , respectively. The total concentration of binding sites $c_{\text{tot}} = c_0 + c_1$ is constant and K_D is the dissociation constant for regulator binding. Derive the equation

$$c_1(x) = \frac{x}{x + K_D} c_{\text{tot}}$$

for the concentration c_1 of occupied binding sites as a function of free transcription factor concentration.

- 4) *Network component analysis.* (a) Consider the transcription network shown below and determine the structure (zeros and signs) of the matrix A for network component analysis. (b) Find the corresponding matrix structure for the MetR regulon (Figure 9.12).



Section 9.4

- 5) *Typical particle numbers.* Typical substance concentrations are on the order of mM (for metabolites), μM (for proteins), and nM (for mRNA). Compute the corresponding orders of magnitude for the particle numbers: compare a prokaryotic cell (diameter 1 μm) to a eukaryotic cell (5 μm). Estimate roughly the range of fluctuations in numbers. Which of the modeling frameworks (deterministic or stochastic) should be used in the different cases?
- 6) *Mean and variance for transcription/translation-model.* Derive the formulae $\langle x \rangle = w_{+x}/w_{-x}$, $\langle y \rangle = \langle x \rangle w_{+y}/w_{-y}$, $\text{var}(x)/\langle x \rangle = 1$, $\text{var}(y)/\langle y \rangle = 1 + \frac{w_{+y}}{w_{-x} + w_{-y}}$ for mean values and Fano factors in the model Eq. (9.28) of transcription and translation. Hint: Use the Langevin equation framework, consider the constant DNA concentration $g = 1$ as an auxiliary variable in the variable vector $\mathbf{z} = (g, x, y)^T$, express the propensities by a linear function $\mathbf{a} = \mathbf{W}\mathbf{z}$, and use the Lyapunov equation.
- 7) *Spectral densities and correlation functions.* (a) Derive the spectral densities Eq. (9.39) for the two-gene model (9.37). (b) Use the result to compute the covariance functions (9.41).
- 8) *Kinetic model of transcription and translation.* Consider, as a simple version of Eq. (9.26), the

kinetic model for mRNA levels s_x, s_y

$$\begin{aligned}\dot{s}_x &= 1 - s_x \\ \dot{s}_y &= s_x - s_y\end{aligned}$$

with initial conditions $s_x(0) = s_x^*, s_y(0) = s_y^*$. Derive the solution

$$\begin{aligned}s_x &= 1 + (s_x^* - 1)e^{-t} \\ s_y &= 1 + ((s_y^* - 1) + (s_x^* - 1)t)e^{-t}.\end{aligned}$$

Sketch the solutions for the following initial values (s_x^*, s_y^*) : (0,0), (1,0), (0,1), (1,1).

- 9) *Variability of protein levels.* (a) Consider the model of transcription and translation Eq. (9.28). How should rate parameters be chosen to allow for high protein levels with small fluctuations? What is the lower limit of protein variance? What practical issues may prevent cells from reaching this limit? (b) Speculate about possible biological functions and about disadvantages of protein bursts.
- 10) *Extrinsic and intrinsic noise in a linear model.* Chemical noise causes mRNA level to fluctuate around its average value. In a linear approximation, the random deviation y can be written as a sum of extrinsic and intrinsic noise terms $y = u_{\text{ext}} + u_{\text{int}}$. Assume that two copies of the same gene, described by variables y_1 and y_2 , share the same extrinsic noise with variance C_{ext} , but are also influenced by independent intrinsic noise variables $u_{\text{int},1}$ and $u_{\text{int},2}$ with variance C_{int} :

$$\begin{aligned}y_1 &= u_{\text{ext}} + u_{\text{int},1} \\ y_2 &= u_{\text{ext}} + u_{\text{int},2}.\end{aligned}$$

- (a) Assume that the noise variables $u_{\text{ext}}, u_{\text{int},1}$, and $u_{\text{int},2}$ are independent, and compute the relative noise levels η_{int} and η_{ext} for the single genes, as well as the linear correlation between y_1 and y_2 . (b) How can the relative intrinsic and extrinsic noise be computed from measurements of y_1 and y_2 ? (c) How can the calculation argument be generalized to many variables and noise parameters?

- 11) *Relative contributions of intrinsic and extrinsic noise.* A biochemical quantity x is affected by intrinsic and extrinsic noise. Show that its total variance can be split into intrinsic and extrinsic contributions $\text{var}(x) = [\langle \langle x^2 \rangle_{\text{int}} - \langle x \rangle_{\text{int}}^2 \rangle_{\text{ext}}] + [\langle \langle x \rangle_{\text{int}}^2 \rangle_{\text{ext}} - \langle x \rangle^2]$.

References

- 1 Orphanides, G. and Reinberg, D. (2002) A unified theory of gene expression. *Cell*, 108 (4), 439–451.
- 2 Proudfoot, N.J., Furger, A., and Dye, M.J. (2002) Integrating mRNA processing with transcription. *Cell*, 108 (4), 501–512.
- 3 Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002) *Molecular Biology of the Cell*, 4th edn, Garland Science, New York.
- 4 Tupler, R., Perini, G., and Green, M.R. (2001) Expressing the human genome. *Nature*, 409, 832–833.
- 5 Allison, L.A., Moyle, M., Shales, M., and Ingles, C.J. (1985) Extensive homology among the largest subunits of eukaryotic and prokaryotic RNA polymerases. *Cell*, 42, 599–610.
- 6 Woychik, N.A. and Hampsey, M. (2002) The RNA polymerase II machinery: structure illuminates function. *Cell*, 108, 453–463.
- 7 Ptashne, M. and Gann, A. (1997) Transcriptional activation by recruitment. *Nature*, 386, 569–577.
- 8 Berg, O.G. and von Hippel, P.H. (1987) Selection of DNA binding sites by regulatory proteins. Statistical mechanical theory and application to operators and promoters. *J. Mol. Biol.*, 193, 723–750.
- 9 Smale, S.T. (1994) Core promoter architecture for eukaryotic protein-coding genes, in *Transcription: Mechanisms and Regulation* (eds R.C. Conaway and J.W. Conaway), Raven Press, New York, pp. 63–81.
- 10 Novina, C.D. and Roy, A.L. (1997) Core promoters and transcriptional control. *Trends Genet.*, 12, 351–355.
- 11 Burke, T.W. and Kadonga, J.T. (1997) The downstream promoter element, DPE, is conserved from *Drosophila* to humans and is recognized by TAFII60 of *Drosophila*. *Genes Dev.*, 11, 3020–3031.
- 12 Pedersen, A.G., Baldi, P., Chauvin, Y., and Brunak, S. (1999) The biology of eukaryotic promoter prediction: a review. *Comput. Chem.*, 1999, 191–207.
- 13 Fickett, J.W. and Hatzigeorgiou, A.C. (1997) Eukaryotic promoter recognition. *Genome Res.*, 7, 861–878.
- 14 Stormo, G.D. (2000) DNA binding sites: representation and discovery. *Bioinformatics*, 16, 16–23.
- 15 Werner, T. (2003) The state of the art of mammalian promoter recognition. *Brief. Bioinform.*, 4, 22–30.
- 16 Kel, A., Kondrakhin, Y., Kolpakov, P., Kel, O., Romashenko, A., Wingender, E. et al. (1995) Computer tool FUNSITE for analysis of eukaryotic regulatory genomic sequences. *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, 3, 197–205.
- 17 Kel-Margoulis, O.V., Kel, A.E., Reuter, I., Deineko, I.V., and Wingender, E. (2002) TRANSCompel: a database on composite regulatory elements in eukaryotic genes. *Nucleic Acids Res.*, 30 (1), 332–334.
- 18 Hutchinson, G.B. (1996) The prediction of vertebrate promoter regions using differential hexamer frequency analysis. *CABIOS*, 12, 391–398.
- 19 Scherf, M., Klingenhoff, A., and Werner, T. (2000) Highly specific localization of promoter regions in large genomic sequences by PromoterInspector: a novel context analysis approach. *J. Mol. Biol.*, 297 (3), 599–606.
- 20 Prestridge, D.S. (1995) Prediction of pol II promoter sequences using transcription factor binding sites. *J. Mol. Biol.*, 249, 923–932.
- 21 Reese, M.G. (2001) Application of a time-delay neural network to promoter annotation in the *Drosophila melanogaster* genome. *Comput. Chem.*, 26 (1), 51–56.
- 22 Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., and Church, G.M. (1999) Systematic determination of genetic network architecture. *Nat. Genet.*, 22 (3), 281–285.
- 23 Elkon, R., Linhart, C., Sharan, R., Shamir, R., and Shiloh, Y. (2003) Genome-wide *in silico* identification of transcriptional regulators controlling the cell cycle in human cells. *Genome Res.*, 13, 773–780.
- 24 Wasserman, W.W., Palumbo, M., Thompson, W., Fickett, J.W., and Lawrence, C.E. (2000) Human–mouse genome comparisons to locate regulatory sites. *Nat. Genet.*, 26 (2), 225–228.
- 25 Dieterich, C., Wang, H., Rateitschak, K., Luz, H., and Vingron, M. (2003) CORG: a database for COmparative Regulatory Genomics. *Nucleic Acids Res.*, 31 (1), 55–57.
- 26 Thakurta, D.G., Palomar, L., Stormo, G.D., Tedesco, P., Johnson, T.E., Walker, D.W. et al. (2002) Identification of a novel *cis*-regulatory element involved in the heat shock response in *C. elegans* using microarray gene expression and computational methods. *Genome Res.*, 12, 701–712.
- 27 Ambros, V. (2004) The functions of animal microRNAs. *Nature*, 431 (7006), 350–355.
- 28 Bartel, D.P. (2004) MicroRNAs: genomics, biogenesis, mechanism, and function. *Cell*, 116 (2), 281–297.
- 29 O'Donnell, K.A., Wentzel, E.A., Zeller, K.I., Dang, C.V., and Mendell, J.T. (2005) c-Myc-regulated microRNAs modulate E2F1 expression. *Nature*, 435 (7043), 839–843.
- 30 Berezikov, E., Cuppen, E., and Plasterk, R.H. (2006) Approaches to microRNA discovery. *Nat. Genet.*, 38 (Suppl.), S2–S7.
- 31 Bentwich, I. (2005) Prediction and validation of microRNAs and their targets. *FEBS Lett.*, 579 (26), 5904–5910.
- 32 Lim, L.P., Glasner, M.E., Yekta, S., Burge, C.B., and Bartel, D.P. (2003) Vertebrate microRNA genes. *Science*, 299 (5612), 1540.
- 33 Griffiths-Jones, S., Saini, H.K., van Dongen, S., and Enright, A.J. (2008) miRBase: tools for microRNA genomics. *Nucleic Acids Res.*, 36 (Database issue), D154–D158.
- 34 Sethupathy, P., Megraw, M., and Hatzigeorgiou, A.G. (2006) A guide through present computational approaches for the identification of mammalian microRNA targets. *Nat. Methods*, 3 (11), 881–886.
- 35 Carrington, J.C. and Ambros, V. (2003) Role of microRNAs in plant and animal development. *Science*, 301 (5631), 336–338.
- 36 Paddison, P.J. and Hannon, G.J. (2002) RNA interference: the new somatic cell genetics? *Cancer Cell*, 2 (1), 17–23.
- 37 Fire, A., Xu, S., Montgomery, M.K., Kostas, S.A., Driver, S.E., and Mello, C.C. (1998) Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegans*. *Nature*, 391 (6669), 806–811.
- 38 Lim, L.P., Lau, N.C., Weinstein, E.G., Abdelhakim, A., Yekta, S., Rhoades, M.W. et al. (2003) The microRNAs of *Caenorhabditis elegans*. *Genes Dev.*, 17 (8), 991–1008.
- 39 Reinhardt, B.J., Weinstein, E.G., Rhoades, M.W., Bartel, B., and Bartel, D.P. (2002) MicroRNAs in plants. *Genes Dev.*, 16 (13), 1616–1626.
- 40 Lagos-Quintana, M., Rauhut, R., Yalcin, A., Meyer, J., Lendeckel, W., and Tuschl, T. (2002) Identification of tissue-specific microRNAs from mouse. *Curr. Biol.*, 12 (9), 735–739.
- 41 Babaie, Y., Herwig, R., Greber, B., Brink, T.C., Wruck, W., Groth, D. et al. (2007) Analysis of Oct4-dependent transcriptional networks regulating self-renewal and pluripotency in human embryonic stem cells. *Stem Cells*, 25 (2), 500–510.
- 42 Crick, F. (1970) Central dogma of molecular biology. *Nature*, 227, 561–563.

- 43** Schwanhausser, B., Busse, D., Li, N., Dittmar, G., Schuchhardt, J. *et al.* (2011) Global quantification of mammalian gene expression control. *Nature*, 473, 337–342.
- 44** Picotti, P., Bodenmiller, B., Mueller, L.N., Domon, B., and Aebersold, R. (2009) Full dynamic range proteome analysis of *S. cerevisiae* by targeted proteomics. *Cell*, 138, 795–806.
- 45** Ghaemmaghami, S., Huh, W.K., Bower, K., Howson, R.W., Belle, A. *et al.* (2003) Global analysis of protein expression in yeast. *Nature*, 425, 737–741.
- 46** Miller, C., Schwalb, B., Maier, K., Schulz, D., Dumcke, S. *et al.* (2011) Dynamic transcriptome analysis measures rates of mRNA synthesis and decay in yeast. *Mol. Syst. Biol.*, 7, 458.
- 47** Memczak, S., Jens, M., Elefsinioti, A., Torti, F., Krueger, J. *et al.* (2013) Circular RNAs are a large class of animal RNAs with regulatory potency. *Nature*, 495, 333–338.
- 48** Gillespie, D.T. (1977) Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81, 2340–2361.
- 49** Ptashne, M. and Gann, A. (2002) *Genes & Signals*, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.
- 50** Keseler, I.M., Collado-Vides, J., Gama-Castro, S., Ingraham, J., Paley, S., Paulsen, I.T., Peralta-Gil, M., and Karp, P.D. (2005) EcoCyc: a comprehensive database resource for *Escherichia coli*. *Nucleic Acids Res.*, 33 (Database issue), D334–D337.
- 51** Kim, H.D. and O’Shea, E.K. (2008) A quantitative model of transcription factor-activated gene expression. *Nat. Struct. Mol. Biol.*, 15 (11), 1192–1198.
- 52** Machné, R. and Murray, D. (2012) The yin and yang of yeast transcription: elements of a global feedback system between metabolism and chromatin. *PLoS One*, 7 (6), e37906.
- 53** Jacob, F. and Monod, J. (1961) Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.*, 3, 318–356.
- 54** Setty, Y., Mayo, A.E., Surette, M.G., and Alon, U. (2003) Detailed map of a *cis*-regulatory input function. *Proc. Natl. Acad. Sci. USA*, 100 (13), 7702–7707.
- 55** Santillán, M. and Mackey, M.C. (2004) Influence of catabolite repression and inducer exclusion on the bistable behavior of the lac operon. *Biophys. J.*, 86, 1282–1292.
- 56** Yuh, C.H., Bolouri, H., and Davidson, E.H. (1998) Genomic *cis*-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science*, 279, 1896–1902.
- 57** Bintu, L., Buchler, N.E., Garcia, H.G., Gerland, U., Hwa, T., Kondev, J., and Phillips, R. (2005) Transcriptional regulation by numbers: models. *Curr. Opin. Genet. Dev.*, 15, 116–124.
- 58** Bintu, L., Buchler, N.E., Garcia, H.G., Gerland, U., Hwa, T., Kondev, J., Kuhlman, T., and Phillips, R. (2005) Transcriptional regulation by numbers: applications. *Curr. Opin. Genet. Dev.*, 15, 125–135.
- 59** Kalir, S. and Alon, U. (2004) Using a quantitative blueprint to reprogram the dynamics of the flagella gene network. *Cell*, 117, 713–720.
- 60** Kaplan, S., Bren, A., Zaslaver, A., Dekel, E., and Alon, U. (2008) Diverse two-dimensional input functions control bacterial sugar genes. *Mol. Cell*, 29, 786–792.
- 61** Sasson, V., Shachrai, I., Bren, A., Dekel, E., and Alon, U. (2012) Mode of regulation and the insulation of bacterial gene expression. *Mol. Cell*, 46, 399–407.
- 62** Mayo, A.E., Setty, Y., Shavit, S., Zaslaver, A., and Alon, U. (2006) Plasticity of the *cis*-regulatory input function of a gene. *PLoS Biol.*, 4 (4), e45.
- 63** Kaplan, S., Bren, A., Erez Dekel, E., and Alon, U. (2008) The incoherent feed-forward loop can generate non-monotonic input functions for genes. *Mol. Syst. Biol.*, 4, 203.
- 64** Ronen, M., Rosenberg, R., Shraiman, B.I., and Alon, U. (2002) Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics. *Proc. Natl. Acad. Sci. USA*, 99 (16), 10555–10560.
- 65** Barencro, M., Tomescu, D., Brewer, D., Callard, R., Stark, J., and Hubank, M. (2006) Ranked prediction of p53 targets using hidden variable dynamic modeling. *Genome Biol.*, 7 (3), R25.
- 66** Keren, L., Zackay, O., Lotan-Pompan, M., Barenholz, U., Dekel, E., Sasson, V., Aidelberg, G., Bren, A., Zeevi, D., Weinberger, A., Alon, U., Milo, R., and Segal, E. (2013) Promoters maintain their relative activity levels under different growth conditions. *Mol. Syst. Biol.*, 9, 701.
- 67** Gerosa, L., Kochanowski, K., Heinemann, M., and Sauer, U. (2013) Dissecting specific and global transcriptional regulation of bacterial gene expression. *Mol. Syst. Biol.*, 9, 658.
- 68** Liao, J.C., Boscolo, R., Yang, Y., Tran, L.M., Sabatti, C., and Roychowdhury, V.P. (2003) Network component analysis: reconstruction of regulatory signals in biological systems. *Proc. Natl. Acad. Sci. USA*, 100 (26), 15522–15527.
- 69** Kao, K.C., Yang, Y.L., Boscolo, R., Sabatti, C., Roychowdhury, V., and Liao, J.C. (2004) Transcriptome-based determination of multiple transcription regulator activities in *Escherichia coli* by using network component analysis. *Proc. Natl. Acad. Sci. USA*, 101 (2), 641–646.
- 70** Salgado, H., Gama-Castro, S., Martínez-Antonio, A., Díaz-Pereido, E., Sánchez-Solano, F., Peralta-Gil, M., García-Alonso, D., Jiménez-Jacinto, V., Santos-Zavaleta, A., Bonavides-Martínez, C., and Collado-Vides, J. (2004) RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in *Escherichia coli* K-12. *Nucleic Acids Res.*, 32 (Database issue), D303–D306.
- 71** Lee, T.I., Rinaldi, N.J., Robert, F., Odom, D.T., Bar-Joseph, Z., Gerber, G.K., Hannett, N.M., Harbison, C.T., Thompson, C.M., Simon, I., Zeitlinger, J., Jennings, E.G., Murray, H.L., Gordon, D.B., Ren, B., Wyrick, J.J., Tagne, J.B., Volkert, T.L., Fraenkel, E., Gifford, D.K., and Young, R.A. (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, 298, 799–804.
- 72** Alter, O., Brown, P.O., and Botstein, D. (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA*, 97 (18), 10101–10106.
- 73** Liebermeister, W. (2002) Linear modes of gene expression determined by independent component analysis. *Bioinformatics*, 18, 51–60.
- 74** Buescher, J.M., Liebermeister, W., Jules, M., Uhr, M., Muntel, J., Botella, E., Hessling, B., Kleijn, R.J., Le Chat, L., Leconte, F., Mäder, U., Nicolas, P., Piersma, S., Rügheimer, F., Becher, D., Besnier, P., Bidnenko, E., Denham, E.L., Dervyn, E., Devine, K.M., Doherty, G., Drulhe, S., Felicori, L., Fogg, M.J., Goelzer, A., Hansen, A., Harwood, C.R., Hecker, M., Hubner, S., Hultschig, C., Jarmer, H., Klipp, E., Leduc, A., Lewis, P., Molina, F., Noiroit, P., Peres, S., Pigeonneau, N., Pohl, S., Rasmussen, S., Rinn, B., Schaffer, M., Schnidder, J., Schwikowski, B., van Dijl, J.M., Veiga, P., Walsh, S., Wilkinson, A.J., Stelling, J., Aymerich, S., and Sauer, U. (2012) Global network reorganization during dynamic adaptations of *Bacillus subtilis* metabolism. *Science*, 335 (6072), 1099–1103.
- 75** Holstege, F.C., Jennings, E.G., Wyrick, J.J., Lee, T.I., Hengartner, C.J., Green, M.R., Golub, T.R., Lander, E.S., and Young, R.A. (1998) Dissecting the regulatory circuitry of a eukaryotic genome. *Cell*, 95, 717–728.
- 76** Elowitz, M.B. and Leibler, S. (2000) A synthetic oscillatory network of transcriptional regulators. *Nature*, 403, 335–338.
- 77** Becskei, A. and Serrano, L. (2000) Engineering stability in gene networks by autoregulation. *Nature*, 405, 590–592.

- 78** Elowitz, M. *et al.* (2002) Stochastic gene expression in a single cell. *Science*, 297, 1183.
- 79** Pedraza, J.M. and van Oudenaarden, A. (2005) Noise propagation in gene networks. *Science*, 307, 1965–1969.
- 80** Paulsson, J. (2004) Summing up the noise in gene networks. *Nature*, 427, 415.
- 81** Thattai, M. and van Oudenaarden, A. (2001) Intrinsic noise in gene regulatory networks. *Proc. Natl. Acad. Sci. USA*, 98 (15), 8614–8619.
- 82** Ozbudak, E.M., Thattai, M., Kurtser, I., Grossman, A.D., and van Oudenaarden, A. (2002) Regulation of noise in the expression of a single gene. *Nat. Genet.*, 31, 69–73.
- 83** Swain, P.S., Elowitz, M.B., and Siggia, E.D. (2002) Intrinsic and extrinsic contributions to stochasticity in gene expression. *Proc. Natl. Acad. Sci. USA*, 99 (20), 12795.
- 84** Colman-Lerner, A. *et al.* (2005) Regulated cell-to-cell variation in a cell-fate decision system. *Nature*, 437, 699–706.
- 85** Sigal, A., Milo, R., Cohen, A., Geva-Zatorsky, N., Klein, Y., Liron, Y., Rosenfeld, N., Danon, T., Perzov, N., and Alon, U. (2006) Variability and memory of protein levels in human cells. *Nature*, 444, 643–646.

Further Reading

- Genetic regulation:** Jacob, F. and Monod, J. (1961) Genetic regulatory mechanisms in the synthesis of proteins. *J. Mol. Biol.*, 3, 318–356.
- Mechanisms of gene expression:** Ptashne, M. and Gann, A. (2002) *Genes & Signals*, Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.
- Gene regulation functions:** Bintu, L., Buchler, N.E., Garcia, H.G., Gerland, U., Hwa, T., Kondev, J., and Phillips, R. (2005) Transcriptional regulation by numbers: models. *Curr. Opin. Genet. Dev.*, 15, 116–124.
- Modeling of gene regulation functions:** Alon, U. (2006) *An Introduction to Systems Biology: Design Principles of Biological Circuits*, CRC Mathematical & Computational Biology, Chapman & Hall.
- Network component analysis:** Liao, J.C., Boscolo, R., Yang, Y., Tran, L.M., Sabatti, C., and Roychowdhury, V.P. (2003) Network component analysis: reconstruction of regulatory signals in biological systems. *Proc. Natl. Acad. Sci. USA*, 100 (26), 15522–15527.

Variability, Robustness, and Information

10

The phenotype of a cell is not fully determined by its genotype. Cells are constantly facing variability, including chemical noise, varying environments, and imperfections of their own components. Even in a clonal population, cells differ in their physiology and form subpopulations with distinctive protein expression. Population heterogeneity can be quantified by fluorescence microscopy or by fluorescence-activated cell sorting (FACS). When modeling such cells, we should also study the consequences of variability in their protein levels. Instead of modeling individual cells in a population, we can use a model ensemble: We consider a single-cell model, but draw the model parameters (e.g., protein levels or parameters influencing them) from probability distributions or add noise to the chemical reaction rates. This results in distribution of cell states, which can then be compared with distributions in real cell populations.

Using model ensembles with broad parameter distributions, we can explore which qualitative behaviors a model may show, or we can focus on specific output variables and study their variations. A large variation may indicate a strong dependence on the varied parameters, while a small variation may indicate that this variable is mostly determined by model structure and fixed parameters. Thus, model ensembles can tell us how different aspects of a model – its structure, rate laws, parameters, and initial or boundary conditions – influence its dynamic behavior.

If some dynamic property is strongly determined by network structure, it can be inferred from the model even if model parameters are unknown, uncertain, or varying in cell populations. Some system properties, such as the set of stationary fluxes on a metabolic network (see Section 3.1), are fully determined by stoichiometry. Others, such as the actual fluxes in kinetic models, depend on parameters. The control coefficients depend

10.1 Variability and Biochemical Models

- Variability and Uncertainty Analysis
- Flux Sampling
- Elasticity Sampling
- Propagation of Parameter Variability in Kinetic Models
- Models with Parameter Fluctuations

10.2 Robustness Mechanisms and Scaling Laws

- Robustness in Biochemical Systems
- Robustness by Backup Elements
- Feedback Control
- Perfect Robustness by Structure
- Scaling Laws
- Time Scaling, Summation Laws, and Robustness
- The Role of Robustness in Evolution and Modeling

10.3 Adaptation and Exploration Strategies

- Information Transmission in Signaling Pathways
- Adaptation and Fold-Change Detection
- Two Adaptation Mechanisms: Sensing and Random Switching
- Shannon Information and the Value of Information
- Metabolic Shifts and Anticipation
- Exploration Strategies

Exercises

References

Further Reading

on network structure and kinetics in a peculiar way (see Section 4.2.2): The summation theorems refer to stoichiometries only, while the connectivity theorems refer to elasticities and, thus, to rate laws and metabolic states. Together, summation and connectivity theorems determine the control coefficients completely.

Finally, instead of varying the model parameters, we can consider chemical noise (see Section 7.2) and study the temporal fluctuations resulting from it. Typically, high-frequency fluctuations will be damped, while slow

fluctuations, resembling static differences between cells, may shift steady states. At intermediate frequencies, fluctuations may propagate in the network and even lead to resonance. The quantitative details depend on parameter fluctuations, network structure, and kinetics.

In this chapter, we first study variability in biochemical models. Then, we turn to real cells and discuss the mechanisms by which they can control inevitable variability, or even create and manage variability as a part of exploration and survival strategies.

10.1 Variability and Biochemical Models

Summary

Cellular behavior is affected by varying parameters such as enzyme levels or temperature and chemical noise. Model ensembles capture the resulting variability in cell states and can be used to obtain probabilistic predictions and to study the general effects of network structure on dynamic behavior. In model ensembles, parameter values are drawn from probability distributions and output variables are characterized by their average values, variances, and correlations. Cells can adapt to randomness and employ it as a part of their search strategies.

Diversity in cell populations can arise from genetic differences, fluctuations in gene expression, changes in nutrient supply, cell cycle phases, or temperature changes, and many other factors. All these factors will influence the metabolic state of a cell, and their effects, for example, correlated changes in metabolite levels, can be observed in high-throughput data [1].

10.1.1 Variability and Uncertainty Analysis

To model heterogeneity in cell populations, we may describe all cells by one single-cell model, but with different parameter choices (e.g., differences in protein levels as seen by flow cytometry or fluorescence microscopy (see Section 14.15)), and compute the resulting spread of state variables such as metabolic concentration or fluxes. We can simulate how parameter variations will affect reaction rates, how perturbations propagate in the network, and how parameter variability translates into variability of outputs such as metabolic fluxes and concentrations. Generally, we can study the effects of defined parameter variations, parameter variability, or uncertainty about parameters (see Figure 10.1):

- 1) **Sensitivity analysis** In sensitivity analysis, we study how a defined change in model inputs (e.g., parameter values or network structure) affects the quantitative or qualitative model behavior. In *local sensitivity analysis*, one studies small parameter changes that do not affect a model's qualitative behavior: If the system response is a differentiable function $y(\mathbf{x})$, the relationship between parameters and output variables can be described by sensitivities $\partial y / \partial x_m$. In kinetic models, these sensitivities, called response coefficients, are studied by metabolic control analysis. Global sensitivity analysis addresses the effects of larger parameter variations.
- 2) **Bifurcation analysis** is concerned with the types of qualitative behavior a model can show. It studies the regions in parameter space associated with different behavior and the possible transitions from one to the other region, called bifurcations (see Section 15.2).

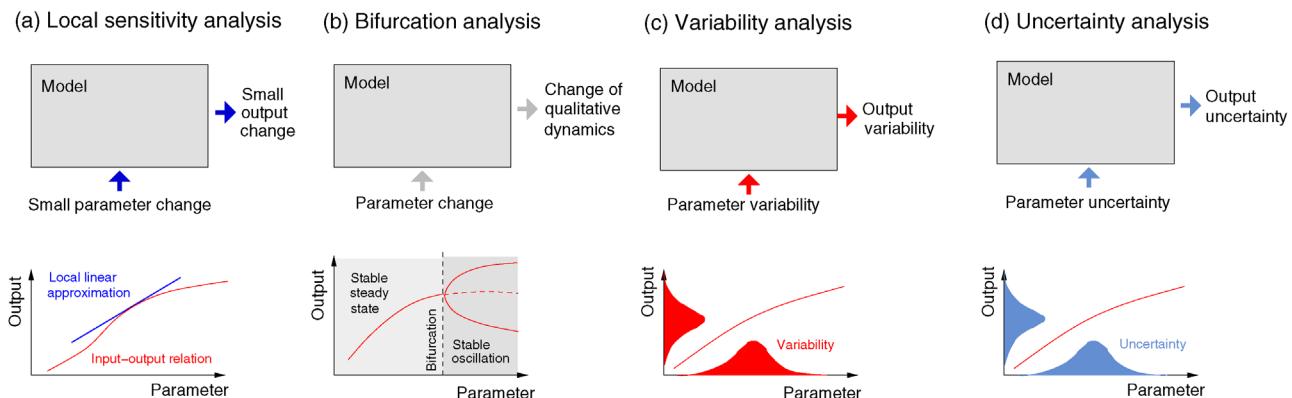


Figure 10.1 Analysis of parameter variation. (a) In local sensitivity analysis, the effects of small parameter variations on system outputs are studied in a linear approximation. (b) In bifurcation analysis, parameter space is split into regions associated with different kinds of qualitative behavior (in the Hopf bifurcation shown, a stable steady state below the bifurcation point and a stable limit cycle above). (c) Variability analysis shows what distributions of output variables will result from parameter distributions. (d) In uncertainty analysis, probability distributions are used to describe subjective uncertainties about a system, for example, uncertainty ranges caused by measurement errors.

- 3) *Variability analysis* is concerned with probability distributions. Assuming distributions of model inputs, and inferring the resulting distributions of model outputs, it can be used to simulate cell populations, to assess the effects of network structure on dynamic behavior, to study robustness properties, and to obtain probabilistic predictions when parameters are uncertain or unknown.
- 4) *Uncertainty analysis* The latter usage of variability in models is also known as *uncertainty analysis*. In *uncertainty analysis*, we try to quantify our (limited) knowledge about model outputs, given the uncertain information about model structure or parameters. Uncertainty analysis can be useful when the precise values of parameters are unknown. If a biochemical model can show a certain behavior, for example, that an inhibition of enzyme A decreases a flux B, an uncertainty analysis with broad parameter distributions can show whether this behavior obtains in general or only for certain parameter choices. If output variables or model properties, for example, the existence of stable oscillations, vary little across the model ensemble, we can attribute these properties to the model's structure.

10.1.1.1 Uncertainty Analysis and the Principle of Minimal Information

The results of a variability analysis depend critically on the parameter distribution assumed. The distribution should be chosen carefully and according to what it is supposed to represent, for example, variability in a cell population, subjective uncertainty due to measurement errors, or broad parameter variability used to explore possible qualitative model behavior. How should we choose parameter distributions in practice? In uncertainty analysis, to faithfully describe our subjective knowledge, a distribution should subsume all information available, and nothing more. For instance, if we know nothing about an enzyme level but upper and lower bounds, we may assume a uniform distribution on the range of possible values.

Generally, the *principle of minimal information* [2] states that one should choose distribution with maximal Shannon entropy, given all constraints that arise from available knowledge (see Section 15.6). If only upper and lower bounds for a parameter are available, entropy is maximized by a uniform distribution, so parameters should be drawn uniformly from this range. If mean value and variance are known, the information principle leads us to choose a Gaussian distribution. For continuous parameters, the information principle can lead to different conclusions depending on how a model is

parametrized. For instance, for a parameter u in a given range, the information principle prescribes a uniform distribution; if we apply the principle to the logarithmic parameter $\ln u$ instead, it will be $\ln u$ that is uniformly distributed, while u will be nonuniform.

Finally, different model parameters may be mutually dependent (e.g., rate constants and equilibrium constants in a kinetic model are linked by Haldane relationships). A safe way to handle such dependencies is to parametrize the model by a set of basic parameters that can be chosen independently, and from which the remaining dependent parameters can be computed. Our choice of independent parameters, when applying the information principle, will have an influence on the resulting parameter distribution. Parameter balancing (see Section 6.1) is a way to define joint distributions of dependent model parameters based on prior knowledge and data.

10.1.1.2 Variability Analysis and Model Ensembles

A model, together with a probability distribution for its parameters, defines a model ensemble. The probability distribution may describe parameter variability between cells, in time, or subjective uncertainty about parameter values. Each model realization can show a different behavior or different values of output variables. The distribution of such outputs can be studied by Monte Carlo simulation [3]: Let y be a quantitative or qualitative system output that depends on the parameter set \mathbf{x} . By drawing n random samples $\mathbf{x}_1, \dots, \mathbf{x}_n$ and computing the respective outcomes $y(\mathbf{x}_\alpha)$, one obtains a histogram of y . For large sample numbers, the empirical distribution approximates the true distribution $p(y)$. Monte Carlo simulation works for all computable models, including nondynamical models as in FBA. However, computing $y(\mathbf{x})$ for many sampled models may be costly. Using fewer samples, on the contrary, increases the statistical uncertainties. As we shall see below, the distributions of output variables can sometimes be computed, approximately, from the input distributions and the model's parameter sensitivities. Thus, variability analysis and sensitivity analysis (or, in the case of biochemical models, metabolic control analysis) are tightly linked.

Intuitively, one may expect that a spread of input parameters will cause a spread of output variables and that the average output values will remain unchanged. However, this need not be true: Parameter spread can shift the outputs systematically, in particular, if input parameters are correlated. For instance, if the substrate concentrations of a bimolecular reaction vary between cells, the average reaction rate will not follow from the mean concentrations, but will also depend on correlated variation. Consider a reaction $A + B \rightarrow C$ with a mass-action rate law $v(a, b) = kab$. If variation in a and b is uncorrelated, then

$\langle ab \rangle = \langle a \rangle \langle b \rangle$, and the average velocity $\langle v \rangle = k \langle ab \rangle$ will be given by $v(\langle a \rangle, \langle b \rangle) = k \langle a \rangle \langle b \rangle$. However, if a and b are correlated, the rate will be increased, and if they are anti-correlated, it will be lowered. Such synergisms can arise from correlated variability across cells, from molecular fluctuations, and from slowly changing correlated concentrations, for example, during day–night cycles.

In the following sections, we study four types of variability in biochemical models: flux variability, variable reaction elasticities at a given flux distribution, variability of parameters in kinetic models, and, finally, temporal fluctuations of such parameters.

10.1.2 Flux Sampling

The constraints in flux balance analysis define a polyhedron of possible stationary flux distributions (see Section 3.2). Specific flux distributions can be selected by optimality criteria, but the solutions are often nonunique. To address this problem, flux variability analysis starts from the same polyhedron and computes the range of possible flux values for each reaction (see Section 3.2). However, this does not show which flux values are actually likely and whether fluxes in different reactions are correlated. To answer these questions, one may sample flux distributions uniformly from the flux polyhedron [4]: Probability distributions of individual reaction fluxes can then be obtained from the marginal distributions.

Flux sampling also works for larger metabolic networks: High-dimensional flux polytopes can be sampled using the hit-and-run algorithm [4], and probability distributions and correlations of individual reaction fluxes can be computed from the samples. Flux sampling covers

Example 10.1 Sampling of Stationary Flux Distributions

In the model in Figure 10.2a, a metabolite X (with concentration s_x) participates in three reactions. In steady state, the fluxes have to satisfy

$$\frac{ds_x}{dt} = v_1 + v_2 + v_3 = 0. \quad (10.1)$$

We impose the additional constraints $0 \leq v_1 \leq 1$, $0 \leq v_2 \leq 1$, $0 \leq v_3 \leq 1.5$, restricting the fluxes to be non-negative and bounded by maximal values. The remaining flux combinations form a convex polyhedron in flux space (Figure 10.2b). If we assume equal probabilities for all feasible flux vectors v , the probability distributions of individual fluxes v_1 , v_2 , and v_3 follow from the polyhedron's projections to the respective dimensions in flux space (Figure 10.2c and d). These distributions are nonuniform: For instance, the maximality constraint on v_3 causes a drop in probability for high values of v_1 and v_2 . Moreover, different fluxes are correlated (e.g., v_1 and v_3) or anticorrelated (e.g., v_1 and v_2), as visible in Figure 10.2c.

all flux distributions a network can show, but it cannot tell us which fluxes are likely to appear in real cell populations. To obtain an approximate answer, Labhsetwar *et al.* [5] considered the distributions of enzyme copy numbers in bacteria, sampled enzyme levels from these distributions, and used them to define individual flux constraints for FBA. In the resulting model ensemble, the variation in predicted growth rates is large, and most of this variation is caused by variation in relatively few enzymes.

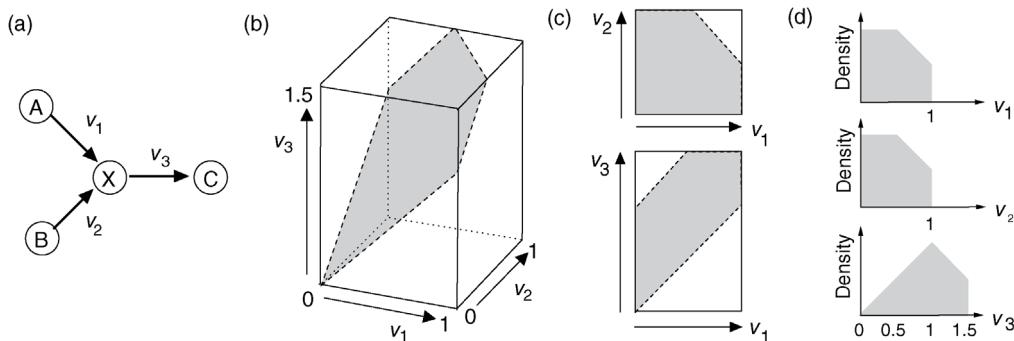


Figure 10.2 Polyhedron of stationary fluxes. (a) Metabolic branch point with three reactions and a balanced metabolite X. (b) Feasible fluxes, defined by stationarity condition and inequality constraints, form a polyhedron in flux space. (c) The same polyhedron, projected to the plane spanned by the basis vectors for v_1 and v_2 (top) or v_1 and v_3 (bottom). (d) The individual fluxes v_1 , v_2 , v_3 show nonuniform probability distributions. If the flux vectors are uniformly distributed in the polyhedron, these distributions arise from the projections of the polyhedron to the three coordinate axes. Flux variability analysis, in contrast, would only yield the ranges, that is, lower and upper bounds.

10.1.3 Elasticity Sampling

The dynamics of metabolic systems depends on their structure (reaction stoichiometries and allosteric regulation arrows) and kinetics (rate laws and rate constants). But how strong is the effect of the latter? For instance, given a metabolic network, how likely is it that a steady state in this network will be stable or unstable? In fact, to answer such questions, we need to clarify which model ensemble we are considering. In theory, one could draw all rate constants from random distributions, solve the model for its steady state, and compute the reaction elasticities. Once the stoichiometric matrix N and the elasticity matrix $\tilde{\epsilon}$ are known, the qualitative behavior follows from the eigenvalues of the Jacobian $A = N\tilde{\epsilon}$ (see Section 15.2). By repeating this procedure many times, we can assess the probabilities for different types of model dynamics and the variability of quantitative model outputs.

10.1.3.1 Elasticity Sampling

Since the calculation of steady states is numerically expensive, Steuer *et al.* [6] have proposed a type of uncertainty analysis, called structural kinetic modeling, in which a metabolic steady state is predefined and the scaled elasticities for this state are randomly sampled. The method builds on the fact that the shape of the elasticity matrix $\tilde{\epsilon}$ and the signs of its elements are mostly determined by the network structure; only the magnitudes of the matrix elements and the placement of elements for allosteric regulation depend on the rate laws. The unscaled elasticities $\tilde{\epsilon}_{ji}$ can be written in terms of scaled elasticities ϵ_{ji} as

$$\tilde{\epsilon}_{ji} = \frac{\partial v_j}{\partial s_i} = v_j \frac{\partial \ln|v_j|}{\partial \ln s_i} s_i^{-1} = v_j \epsilon_{ji} s_i^{-1} \quad (10.2)$$

or in matrix form

$$\tilde{\epsilon} = Dg(\mathbf{v})\epsilon Dg(\mathbf{s})^{-1}. \quad (10.3)$$

A scaled elasticity ϵ_{li} represents an apparent reaction order: With linear or bilinear rate laws, it has a value of 1 for substrates and -1 for products. For Michaelis–Menten kinetics and common types of allosteric activation and inhibition, the absolute value can vary between 1 (resembling mass-action kinetics) and 0 (saturation).

Equation (10.2) is well suited for variability analysis: Given stationary fluxes and concentrations, the scaled elasticities ϵ_{li} can be drawn randomly and independently from uniform distributions. Now all control properties, including the unscaled elasticities and the Jacobian matrix, are determined. The matrix $Dg(\mathbf{s})^{-1} A Dg(\mathbf{s})$, a scaled version of the Jacobian A , allows us to determine the qualitative behavior, for example, the occurrence of

bifurcations. The comparison between models will not depend on, for example, their particular steady-state concentrations. Elasticity sampling can be used to study the robustness properties of systems in which rate laws are not known precisely [7]. Instead of predefining the stationary fluxes and concentrations, we can also sample them with the fluxes satisfying stationarity and thermodynamic constraints (see Section 3.2.1), or insert fluxes or concentrations known from experiments into the model.

10.1.3.2 Elasticity Sampling under Thermodynamic Constraints

In kinetic models with reversible rate laws, elasticities are linked by thermodynamic constraints. These constraints must be respected in elasticity sampling because otherwise the sampled elasticities will not be realizable by a feasible model. For many reversible rate laws, the scaled elasticities can be split into a sum [8]

$$\epsilon_{ji} = \epsilon_{ji}^{\Theta} - \epsilon_{ji}^D, \quad (10.4)$$

with a thermodynamic contribution given by

$$\begin{aligned} \epsilon_{ji}^{\Theta} &= \frac{\zeta_j m_{ij}^{\text{sub}} - m_{ij}^{\text{prod}}}{\zeta_j - 1}, \quad \text{where } \zeta_j = \frac{v_j^+}{v_j^-} \\ &= \exp^{A_j/RT}. \end{aligned} \quad (10.5)$$

The kinetic term ϵ_{ji}^D depends on the saturation values, variables of the form $\alpha_{ji} = 1/(K_{m,ji} + s_i)$ or $\beta_{ji} = s_i/(K_{m,ji} + s_i)$ describing the saturation of enzymes with reactants or allosteric regulators. The precise formula of ϵ_{ji}^D depends on the rate law. Saturation values describe the saturation of an enzyme with reactants or allosteric regulators on a scale between 0 and 1 and can be chosen independently without violating any constraints. Their independent choice allows a thermodynamically consistent form of elasticity sampling in several steps: First, thermodynamically feasible fluxes and concentrations, together with the driving forces A_r , are determined; then saturation values are sampled and the elasticities are computed. Finally, the elasticities can be translated into corresponding rate constants. The resulting kinetic models satisfy all Wegscheider conditions and Haldane relationships. By either predefining or sampling different kinds of variables, one can flexibly build model ensembles with consistent, fixed or sampled, metabolic states. These model ensembles can be used to study possible control properties of models, their dependence on model structure, and their uncertainties [9–13].

Instead of sampling the elasticities, one may also sample rate constants directly at predefined or sampled metabolic states (see Section 6.4) [14]. Like in thermodynamically feasible elasticity sampling, a feasible steady

state (with equilibrium constants, metabolite concentrations, and fluxes) is chosen first. Then, K_m values are freely chosen and forward and backward catalytic constants are computed from the predefined fluxes and from Haldane relationships.

10.1.4 Propagation of Parameter Variability in Kinetic Models

In another type of variability analysis, we consider a kinetic model in a given steady state and study how a small variation of model parameters (e.g., enzyme levels or extracellular concentrations) would affect that metabolic state. Consider a kinetic model

$$\frac{ds(t)}{dt} = \mathbf{Nv}(s(t), \mathbf{k}), \quad (10.6)$$

with stoichiometric matrix \mathbf{N} , reaction rate vector \mathbf{v} , and a vector \mathbf{k} of parameters affecting the rates. If the model has a stable steady state and \mathbf{k} follows a random distribution, the steady-state concentrations $s^{st}(\mathbf{k})$ follow a probability distribution, which we can characterize by variances and correlations. If instead the parameters fluctuate in time, following some random process $\mathbf{k}(t)$, the metabolic state will be given by a random process and the frequency spectrum of their fluctuations will be shaped by the system's dynamics (see Section 15.4) [15].

If our model parameters, although being random variables, are assumed to be constant in time, the model dynamics can be captured by a response function $y(x)$. The output vector \mathbf{y} can comprise steady-state concentrations, time series, or time series characteristics such as maximal values or response times, while the parameter vector \mathbf{x} contains kinetic parameters and initial values, possibly in logarithmic form. In a cell population, the

parameter vector \mathbf{x} will not have a fixed value, but joint probability distribution, where probabilities represent assumed percentages in the cell population.

For positive model parameters, it is often convenient to use logarithmic values because multiplicative relationships for the original parameters (e.g., the Haldane relationship $k_+/k_- = K_{eq}$ for mass-action rate laws) can then be expressed by additive relationships for the logarithms ($\ln k_+ - \ln k_- = \ln K_{eq}$) [16,17]. Furthermore, if we assume that our parameters follow log-normal distributions, the logarithmic parameters will follow Gaussian distributions, and we can specify a multivariate Gaussian distribution for all logarithmic parameters $x_m = \ln k_m$ by a mean value $\langle \mathbf{x} \rangle$ and a covariance matrix $\text{cov}(\mathbf{x})$. Since linear combinations of Gaussian random variables are again Gaussian distributed, large parameter sets with multiplicative dependencies (e.g., all constants $\ln k_+$, $\ln k_-$, and $\ln K_{eq}$ in a larger model) can be specified by a joint Gaussian distribution of the logarithmic parameters [18]. The precise shapes of such distributions can be obtained from data with the help of parameter balancing (see Section 6.1). The probability density $p(\mathbf{x})$ of the parameters, together with the shape of the function $y(\mathbf{x})$, determines the distribution of the output variables, as shown in Figure 10.3.

10.1.4.1 Propagation of Variability

To compute output distributions caused by narrow parameter distributions, we linearize the system's input-output relation (see Figure 10.3) around a typical parameter set. With a linear response function $y(x) = Rx$ (as in Figure 10.3a), the distribution of y will resemble the distribution of x , but stretched in width by a factor $R = dy/dx$, the slope of $y(x)$. To obtain a normalized probability density $p(y)$ (i.e., a probability density with an integral value of 1), we have to divide it by the same

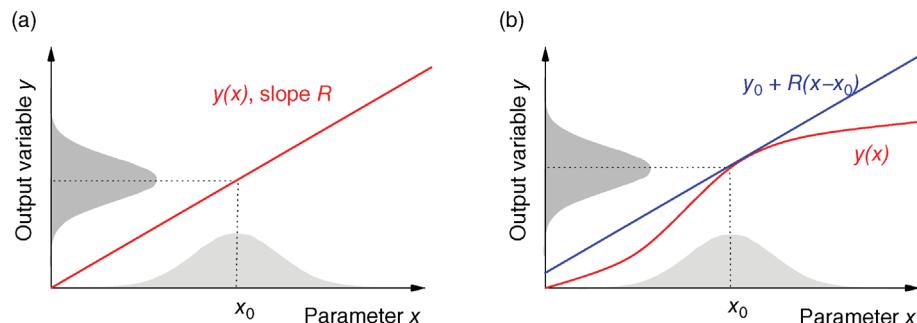


Figure 10.3 Model ensemble with uncertain parameters. (a) If an output variable y depends linearly on x , the distributions of both variables have the same shape and their relative scaling depends on the slope R of $y(x)$. (b) Typically, observables y (e.g., steady-state fluxes) in systems biology models depend nonlinearly on parameters x (e.g., enzyme levels). The response curve $y(x)$ can be approximated by its tangent in a reference point (mean parameter value). In this approximation, the width of y depends on the slope R of the tangent, called sensitivity or response coefficient.

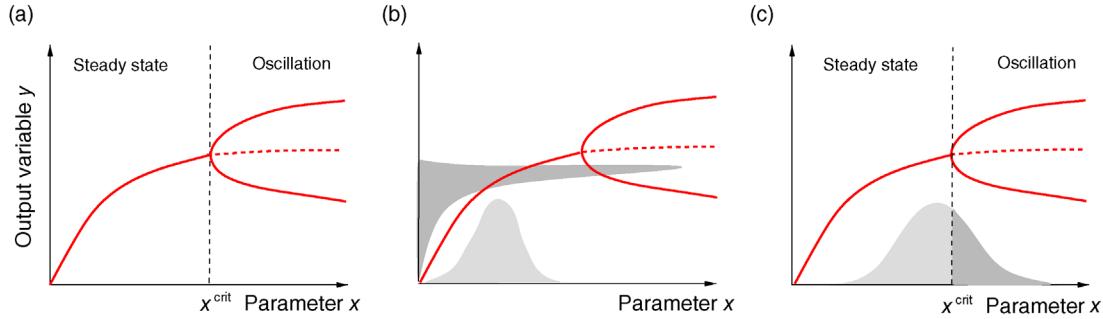


Figure 10.4 Parameter dependence in a system with a Hopf bifurcation. (a) Bifurcation analysis. Depending on the parameter x , the system shows a stable steady state (left region) or sustained oscillations (right region). A bifurcation (change of qualitative behavior, see Section 15.2) happens at a critical value x^{crit} (dashed line). (b) In variability analysis, a distribution of the parameter x (probability density shown in light gray) leads to a distribution of the steady-state value $y(x)$ (density in dark gray, attached to the y -axis). (c) Variability analysis for a qualitative property: The probability for oscillations is given by the area of the parameter density above the critical value x_{crit} (dark gray).

factor and obtain the probability density $p(y(x)) = p(x)/R$. If the original response curve $y(x)$ is nonlinear, we can approximate it by its tangent in the point x_0 :

$$y(x) \approx y(x_0) + R(x - x_0), \quad (10.7)$$

with the sensitivity $R = dy/dx|_{x=x_0}$. If the parameter x is Gaussian distributed with mean value x_0 and a small width σ_x , and if the linear approximation is valid in this parameter region, the distribution of y will be Gaussian with mean $y(x_0)$ and standard deviation $\sigma_y = R\sigma_x$. The approximation breaks down if the output behavior changes abruptly: As an example, Figure 10.4 shows a system with a bifurcation between two types of qualitative behavior. In this case, with one variable parameter and all other parameters kept fixed, the probabilities for the types of qualitative behavior can still be computed from the quantiles of the Gaussian distribution.

The same approximation also works for larger systems with many observables y_i and parameters x_m following a multivariate Gaussian distribution. If the state vector \mathbf{y} describes steady-state concentrations and \mathbf{x} describes model parameters, differentiating $y_l(\mathbf{x})$ by x_m yields the unscaled metabolic response coefficient $\tilde{R}_{x_m}^{y_l}$; if concentrations and parameters are expressed in logarithmic form, we obtain the scaled response coefficients $R_x^{y_l}$ instead. With a narrow parameter distribution and a sufficiently smooth response function $y(\mathbf{x})$, the output values can be approximated by the linear expansion:

$$y_l(\mathbf{x}_0 + \Delta\mathbf{x}) \approx y_l(\mathbf{x}_0) + \sum_m R_{lm} \Delta x_m, \quad (10.8)$$

with sensitivities R_{lm} . Accordingly, the observables y_i will follow a joint Gaussian distribution [16] with

$$\langle \mathbf{y} \rangle \approx \mathbf{y}(\mathbf{x}_0), \quad (10.9)$$

$$\text{cov}(\mathbf{y}) = \langle \mathbf{y}\mathbf{y}^T \rangle \approx \mathbf{R} \langle \mathbf{x}\mathbf{x}^T \rangle \mathbf{R}^T = \mathbf{R} \text{cov}(\mathbf{x}) \mathbf{R}^T. \quad (10.10)$$

According to Eq. (10.10), the variability of the outputs \mathbf{y} depends on two factors: the parameter's covariance matrix $\text{cov}(\mathbf{x})$ and the sensitivity matrix \mathbf{R} . Output variables that are insensitive to parameter changes will show little overall variability and will therefore be robust. Equations (10.9) and (10.10) are not restricted to steady states. If $\mathbf{s}(t; \mathbf{k})$ denotes a time-dependent solution of Eq. (10.6), the variability of this time course under a variation of parameters can be computed using the time-dependent response coefficients (see Ref. [19] and Chapter 4):

$$\tilde{R}_{k_m}^{s_l}(t) = \frac{\partial s_l(t; \mathbf{k})}{\partial k_m}. \quad (10.11)$$

The propagation of uncertainties from parameters to state variables holds also for logarithmic values: If the parameters x_m in Eq. (10.8) are logarithmic rate constants $x_m = \ln k_m$, and if the outputs denote logarithmic steady-state concentrations $y_l = \ln s_l^{\text{st}}$, the sensitivities in (10.8) are given by the scaled response coefficients $R_{k_m}^{s_l}$. With the linear approximation (10.8), and parameters x_m being Gaussian distributed, the original kinetic parameters k_m and the concentrations s_l follow log-normal distributions.

10.1.4.2 Variability Can Shift Mean Values

Most measurements in systems biology do not concern single cells, but yield averages over a cell population. When fitting a single-cell model to such data, one implicitly assumes that the model output averaged over an ensemble of cells with different parameters, $\langle y(\mathbf{x}) \rangle$, is identical to the model output $y(\langle \mathbf{x} \rangle)$ obtained from the averaged parameters. If the response function between parameters and model output is nonlinear, this identity will not exactly hold. We can see this from the second-order approximation:

$$\begin{aligned} y_l(\mathbf{x}_0 + \Delta\mathbf{x}) &\approx y_l(\mathbf{x}_0) + \sum_m R_{lm} \Delta x_m \\ &+ \frac{1}{2} \sum_{mn} R_{lmn} \Delta x_m \Delta x_n, \end{aligned} \quad (10.12)$$

with the second-order sensitivities $R_{lmn} = \partial^2 y_l / \partial x_m \partial x_n$ [16,20]. In this approximation, mean value and covariance matrix of \mathbf{y} read

$$\langle y_l \rangle \approx y_l(\mathbf{x}_0) + \frac{1}{2} \sum_{mn} R_{lmn}^{(2)} C_{mn}, \quad (10.13)$$

$$\begin{aligned} \text{cov}(y_l, y_k) &\approx \sum_{mn} R_{lm} C_{mn} R_{kn} \\ &+ \frac{1}{4} \sum_{mnr} R_{lmn} R_{krs} (C_{ms} C_{nr} + C_{mr} C_{ns}), \end{aligned} \quad (10.14)$$

where C is the covariance matrix of \mathbf{x} . We can see that the average value $\langle y_l \rangle$ deviates from the value $y_l(\langle \mathbf{x}_0 \rangle)$ obtained from the average parameter set. Moreover, even if the parameters \mathbf{x} follow a Gaussian distribution, the output variables y_l will not be Gaussian. The higher order terms can be neglected only if the response function $y(\mathbf{x})$ is weakly curved or if the parameters show negligible variability. Otherwise, when fitting a model to population-averaged data, we cannot expect that the estimated parameters will reflect the average parameters in the population.

10.1.4.3 The Value of Robustness

If cell parameters (e.g., enzyme levels) have been optimized for maximal fitness, adding random noise to their optimal values will decrease the fitness function. The fitness losses and the advantages from noise-dampening robustness mechanisms can be quantified. We assume that the fitness is a function $f(\mathbf{x})$ of the cell parameters \mathbf{x} and expand it to second order

$$f(\mathbf{x}_0 + \Delta\mathbf{x}) \approx f(\mathbf{x}_0) + R_x^f \Delta\mathbf{x} + \frac{1}{2} \Delta\mathbf{x}^T R_{xx}^f \Delta\mathbf{x}, \quad (10.15)$$

where the row vector R_x^f and the matrix R_{xx}^f contain the first- and second-order derivatives of f . If the parameters are randomly distributed with mean \mathbf{x}_0 and covariance matrix $\text{cov}(\mathbf{x})$, the average fitness can be expanded like in Eq. (10.13) as follows:

$$\langle f \rangle \approx f(\mathbf{x}_0) + \frac{1}{2} \text{Tr}[R_{xx}^f \text{cov}(\mathbf{x})]. \quad (10.16)$$

Now assume that a cell can choose its average parameters \mathbf{x}_0 to optimize f , but cannot avoid variability around them. If \mathbf{x}_0 is locally optimal, the eigenvalues of R_{xx}^f (i.e., the local fitness curvatures) will be negative, so the

second term represents a fitness loss due to parameter variability. The loss becomes large when state variables with strong negative fitness curvatures are affected by parameters with high variance. Accordingly, robustness mechanisms that keep the response coefficients between these pairs small are best capable of increasing the average fitness.

10.1.5 Models with Parameter Fluctuations

So far, we considered parameters that were variable between model instances, yet constant in time. Now we study the effects of temporally varying, periodic or stochastic, parameters. These effects will be dynamic: If a cell contains a biochemical oscillator, if it receives periodic inputs from the environment, or if the enzyme levels vary periodically, driven oscillations will spread in the biochemical networks like damped waves, evoking phase-shifted oscillations in pathway fluxes and metabolite concentrations [15,21]. The propagation of random noise can be described very similarly.

10.1.5.1 Biochemical Systems under Periodic Perturbations

The propagation of periodic, small-amplitude perturbations can be described by a variant of metabolic control analysis. To compute the amplitudes and phases of the driven metabolite and flux oscillations, we first linearize our model around a stable steady state. The linearized model $d\mathbf{y}/dt = \mathbf{Ay} + \mathbf{Bx}$ (see Section 6.3) describes deviations \mathbf{x} of metabolite concentrations from their reference values and deviations \mathbf{x} of model parameters, for example, of external concentrations or enzyme levels. Similar equations are used to study oscillations in mechanics, electrical engineering, and control engineering. In the case of biochemical systems, the matrices \mathbf{A} and \mathbf{B} depend on the stoichiometric matrix and the elasticity matrices. In the frequency domain, the dynamics of such models is described by a frequency response function $\mathbf{R}_x^y(\omega) = (\mathbf{A} - i\omega\mathbf{I})^{-1}\mathbf{B}$ (see Section 15.5). Assuming a periodic perturbation $\mathbf{x}(t) = \tilde{\mathbf{x}}e^{i\omega t}$ with complex amplitude vector $\tilde{\mathbf{x}}$ and circular frequency ω , the resulting complex amplitude vector of the state variables is given by $\tilde{\mathbf{y}}(\omega) = \mathbf{R}_x^y(\omega)\tilde{\mathbf{x}}$. The matrix $\mathbf{R}_x^y(\omega)$ can be seen as a generalized metabolic response matrix for periodic perturbations [21,22]. Under periodic perturbations, the responses show phase shifts, and the spectral response coefficients are complex numbers that depend on the driving frequency. The second-order response coefficients can capture synergy effects between harmonic perturbations [22].

10.1.5.2 Biochemical Systems under Random Fluctuations

The same response coefficients allow us to capture the propagation of small-amplitude noise. Fluctuating parameters, for example, enzyme levels, can be described by continuous random processes, which contain a spectrum of fluctuations of different frequencies (see Sections 7.2.5 and 15.4). If parameter fluctuations are very slow, the system response can be understood as a quasi-steady state and can be approximated using static, randomly distributed parameters. When parameters fluctuate fast, the system cannot adapt to these changes and will effectively respond to their mean values. Between these two extremes are parameter fluctuations that take place on the time scale of the system's own dynamics. These can propagate through the network, be damped or amplified, and possibly lead to resonance (see Section 7.2.5).

Using linearized biochemical models and specific noise processes, the effects can be computed. If parameters follow a stationary Gauss–Markov process, their variances and time correlations and the correlations between variables are encoded in the (frequency-dependent) spectral density matrix. If the original noise is described by non-logarithmic perturbation parameters \mathbf{x} with spectral density matrix $\Phi_x(\omega)$, the state variables \mathbf{y} fluctuate with spectral density matrix:

$$\Phi_y(\omega) = \tilde{\mathbf{R}}_x^y(\omega)\Phi_x(\omega)\tilde{\mathbf{R}}_x^y(\omega)^\dagger. \quad (10.17)$$

This equation comprises Eq. (10.10) for static random parameters (at frequency 0) as a special case. Extending the calculations to temporal fluctuations, we have just replaced the covariance matrix and response coefficients, respectively, by spectral density matrices and spectral response coefficients. Noise, represented by $\Phi_x(\omega)$, is translated into output fluctuations, represented by $\Phi_y(\omega)$. For white noise inputs, in particular, the fluctuation spectrum is directly determined by the spectral response coefficients. In any case, the linearized system acts as a frequency filter: Small spectral response coefficients R_{im} lead to dampening, whereas large response coefficients lead to an amplification of noise effects.

10.2 Robustness Mechanisms and Scaling Laws

Summary

Robustness, the ability to maintain biological function despite perturbations, is an essential property of cells and

organisms. It appears on all levels of organization and can be mechanistically realized in many ways. Cellular networks contain structures such as feedback loops that specifically convey robustness. Signaling pathways like the bacterial two-component systems can even show robustness against varying gene expression of their own components. Quantitative robustness and control properties can result from scaling properties of biochemical systems, for example, from the fact that output variables scale proportionally under a rescaling of time. Which robustness properties are found in cells depends on a number of factors: on the typical perturbations, on the detrimental effects of these perturbations, on the trade-offs between performance and robustness, on the costs of robustness mechanisms, and on the tradeoffs between different robustness requirements. Known robustness properties of biological systems can be a helpful guideline for model building.

Cells have to cope with varying external (e.g., temperature or nutrients) and internal conditions (e.g., gene expression noise or varying cell cycle phases). Ideally, such perturbations should not prevent cells from performing their normal physiological function. This requirement, called robustness, can lead to the evolution of specific mechanisms, for example, feedback mechanisms that stabilize substance levels, or signaling pathways that sense stress situations (e.g., osmostress) and activate the cells' defense mechanisms. Robustness against variation in external conditions, against fluctuations in the cell's internal state, and against failure of system components is an important and characteristic property of all living systems [23,24]. On an organismal level, robustness can be seen at work in embryonic development: The reliable unfolding of the body plan – visible in the symmetry in our bodies and the similarities between twins – shows that the developmental dynamics is highly robust.

Apart from being robust, cells also need to be able to respond precisely and adequately to external stimuli. Robustness and sensitivity are two sides of the same coin, and good compromises between the two are vital for cells. However, their interplay can be quite subtle: On the one hand, sensory systems must be robust against perturbations to remain sensitive to proper input signals and, on the other hand, sensitive signaling systems can enable adaptations and thereby enhance the overall robustness of the cell. In this section, we will discuss how robustness is achieved by cellular mechanisms and how knowledge about these mechanisms can be used to guide model building. An overview of types and examples of robustness is given in Figure 10.5.

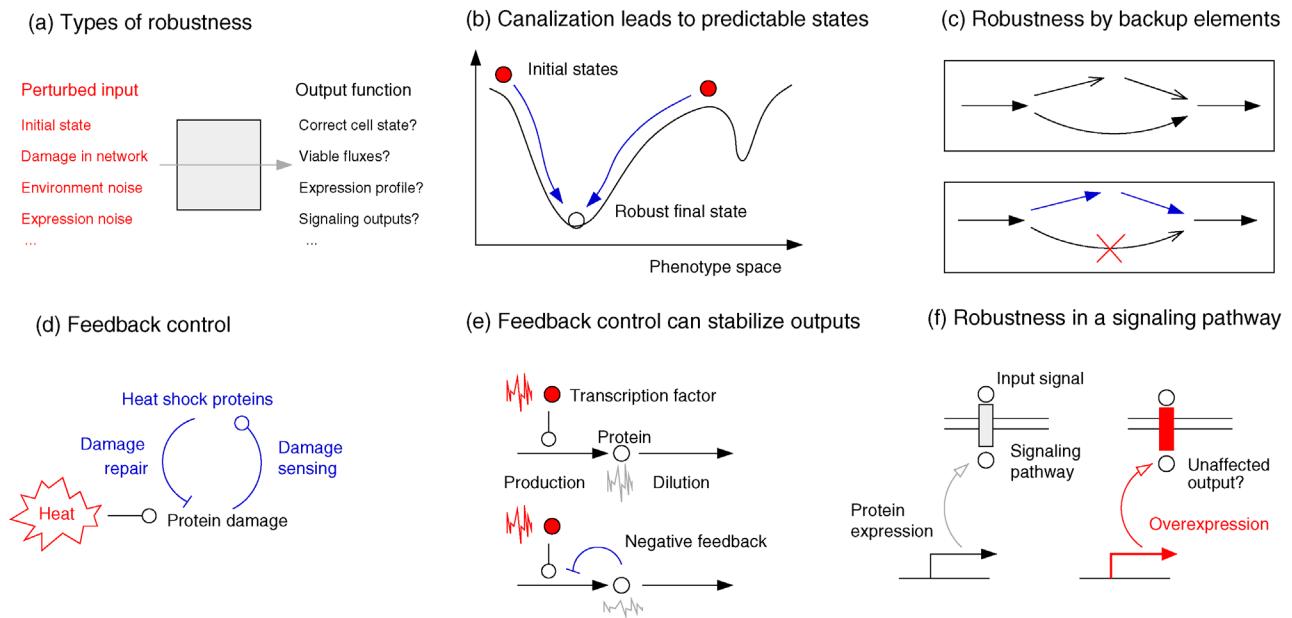


Figure 10.5 Types of robustness and robustness mechanisms. (a) Systems that maintain their function despite perturbations are called robust. Different types of robustness are shown in the following panels. (b) Canalization. Various initial conditions lead to the same final state. (c) Failure tolerance. After a reaction has been disrupted, a backup pathway is used. (d) A feedback system provides robustness against cell damages caused by high temperature. (e) Robustness by feedback regulation. By repressing its own production, a protein can stabilize its expression level against noise caused by fluctuating transcription factor levels. (f) The copy number of a signaling pathway varies due to expression noise. To make signal transduction reliable, the output signal (e.g., the total number of active kinase subunits, given a certain input signal) must be independent of the number of signaling complexes in the cell.

10.2.1 Robustness in Biochemical Systems

10.2.1.1 Biological Robustness Properties

When we speak of robustness (whether in biological systems or in models describing them), we describe a relation between two quantities: an input quantity x that is perturbed, and an output $y(x)$ that should be insensitive to those perturbations. Mathematically, an output y is robust to an input x if deviations of x from its reference value have little (or no) effect on y . In other words, many possible values of x will yield similar or identical values of y , and random variation of x causes little (or no) variability in y . Depending on the nature of x , different types of robustness can be distinguished [24]. Some network structures imply *parameter-insensitive* behavior, behavior that depends only weakly on reaction kinetics and rate constants. In *homeostasis*, a quantity is stabilized against external dynamic perturbations. *Canalization* guarantees that the system will reach a specific outcome from a wide range of initial conditions. *Failure tolerance* means that a network remains functional even when some of its parts are deleted. In kinetic models, all these scenarios can be described by parameter perturbations such as, for instance, perturbations of enzyme levels or initial

conditions: Also missing reactions can also be described in this way by setting parameters to zero.

10.2.1.2 Mathematical Robustness Criteria

To identify different types of robustness, different mathematical criteria are used. First, the metabolic response coefficients in kinetic models allow us to estimate robustness to small parameter changes. This also works for random perturbations: According to Eq. (10.10), a response matrix R_x^y determines how much variability would be caused by narrow parameter distributions. Second, to assess the effects of larger perturbations, we can quantify the range of parameter sets that lead to specific types of behavior. The robustness of a dynamical behavior such as oscillations can be quantified by the area in parameter space in which this behavior occurs [24]. Third, the robustness of a steady state (or other dynamical attractors) against initial conditions can be quantified by the size of its basin of attraction, that is, the set of points from which this final state will be reached. Finally, robustness against gene deletions can be quantified by the number of single-gene deletions that leave a certain network property intact. In all cases, the areas or numbers in question can be estimated by random sampling.

10.2.2

Robustness by Backup Elements

Many loss-of-function mutations in cells have little effect on cell viability. Only about 300 of the 4000 genes in *E. coli*, for instance, have been classified as *essential* [23]. Genes can be dispensable because of different reasons. On the one hand, many gene products are used only under special circumstances: A metabolic pathway, for instance, may be required during growth on minimal medium, but not in a medium in which the pathway product is already provided. On the other hand, some proteins (e.g., isoenzymes) and metabolic pathways can replace each other and serve as backups when one of them is knocked out. All these genes, while being dispensable under ideal standard conditions, may be needed under harder conditions or in environments that provide special opportunities.

10.2.2.1 Backup Genes and Gene Loss

Backup components can make biological as well as technical systems more robust: If a component exists twice (e.g., the two alleles of a gene in diploid cells), the system can still function when one of the components fails. Gene duplications can happen randomly and be conserved if they turn out to be beneficial. In the yeast *Saccharomyces cerevisiae*, about 60% of all genes have duplicates because of a whole-genome duplication that occurred in ascomycetes [25]. Once genes become duplicated – or once a gene copy disappears – their expression levels will change and may have to be readjusted. Quantitatively, robust protein levels can be ensured, for instance, by gene silencing (e.g., of one copy of the X chromosomes in females) or by feedback control of expression levels. Moreover, if protein levels still vary, downstream robustness mechanisms can ensure that the physiological output, for example, the performance of a signaling pathway, remains unaffected.

While gene duplications can provide advantages, they may also come at a cost, especially in cells that need to keep their genomes small. A selection pressure on small genomes, for instance, has been suggested for hummingbirds [26]: Since genome size, nucleus size, and red blood cell size are positively correlated in vertebrates and since smaller red blood cells facilitate gas exchange, a high metabolic energy demand would favor smaller genomes. In fact, hummingbirds, whose energy demand during flight is particularly high, show the smallest known genomes among birds. In general, cells that reduce their genome size during evolution may develop “anti-backup” strategies: Obligate intracellular parasites have lost many genes that are dispensable during life in host cells – not only because of the material and energy, but also because

of the relatively constant environment that host cells provide. Similarly, virus strains can lose genes and profit from gene products produced by wild-type viruses residing in the same cell [27]. By reducing their genome, these “selfish” virus mutants can replicate faster. However, they also depend on other viruses and thus become vulnerable. A similar dependence arises in cells that have lost the ability to produce certain compounds: These substances, for example, certain amino acids, must be taken up from food and thus become essential nutrients for the organism.

10.2.2.2 Backup Pathways

Backup strategies exist not only between genes but also between entire pathways. Whether a metabolic pathway is used depends upon external conditions, that is, on whether it *can* be run and whether it *needs to be* run. In the absence of oxygen, a yeast cell cannot use respiration, but it may still rely on glycolysis for energy generation. Likewise, when a pathway is blocked by a gene loss, it may be bypassed by redirecting the metabolic fluxes through alternative pathways. On the contrary, pathways may become dispensable in certain environments: For instance, if the product of an anabolic pathway can also be imported, the pathway need not be expressed [24]. In the yeast *S. cerevisiae*, such higher level robustness mechanisms turn out to be more important than single-gene backups [28].

10.2.3

Feedback Control

A basic task in control engineering is to keep output quantities of a technical system constant despite perturbations. In living systems, such a behavior is called homeostasis, and it is one of the preconditions for life.

10.2.3.1 Feedback Regulation Changes the System Dynamics

A simple and powerful way to buffer an output quantity against perturbations is *negative feedback* (see Section 8.2.3). Consider a linear model:

$$\frac{dx}{dt} = Ax + Bu, \quad (10.18)$$

with the Jacobian matrix **A** and an input vector **u(t)**. Such models can be obtained from biochemical models by linearization (see Section 6.3.4). Fluctuations of **u** will affect the output **x**, and the system’s robustness against such fluctuations depends on the eigenvalues of **A**. In technical systems, the eigenvalue spectrum can be shaped by coupling the system to a feedback controller. The controller measures the current state **x**, computes a linear function

$\mathbf{z} = \mathbf{Fx}$, and adds it to the system input. The resulting *closed-loop system* (see Figure 15.14) follows the dynamics

$$\frac{dx}{dt} = \mathbf{Ax} + \mathbf{B}(\mathbf{u} + \mathbf{z}) = (\mathbf{A} + \mathbf{BF})\mathbf{x} + \mathbf{Bu}, \quad (10.19)$$

with a new Jacobian matrix $\mathbf{A} + \mathbf{BF}$. With an appropriately chosen feedback matrix \mathbf{F} , all eigenvalues of the Jacobian may become negative and the coupled system will be stabilized against perturbations (see Section 15.5). The stabilizing effect of negative feedback has been shown, for instance, for proteins that repress their own transcription [29]. In theory, such a self-repression could also have the opposite effect: By pushing some complex eigenvalues to the right side of the complex plane, feedback control can destabilize otherwise stable systems, allowing them to show sustained oscillations (see Section 8.2). This typically happens in cases of negative feedback with long time delays. In reality, the possibilities to control systems through feedback mechanisms are limited. First, a single feedback arrow (e.g., an allosteric regulation in a metabolic network) will not specifically change a single eigenvalue, but all eigenvalues at the same time. Second, real feedback controllers cannot freely sense and manipulate the state vector \mathbf{x} , but sense some of the outputs and affect some of the inputs only. Two resulting issues, called observability and controllability, are discussed in Section 15.5.

In kinetic metabolic models, the Jacobian \mathbf{A} is obtained by multiplying the stoichiometric matrix with the elasticity matrix. For the pathway in Figure 10.6, with a fixed influx rate $v_0 > 0$, the Jacobian \mathbf{A} reads

$$\begin{aligned} \mathbf{A} = \mathbf{N}\tilde{\epsilon} &= \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{pmatrix} \begin{pmatrix} 0 & 0 \\ \tilde{\epsilon}_{11} & \tilde{\epsilon}_{12} \\ 0 & \tilde{\epsilon}_{22} \end{pmatrix} \\ &= \begin{pmatrix} -\tilde{\epsilon}_{11} & -\tilde{\epsilon}_{12} \\ \tilde{\epsilon}_{11} & \tilde{\epsilon}_{12} - \tilde{\epsilon}_{22} \end{pmatrix}. \end{aligned} \quad (10.20)$$

Its diagonal elements describe a direct self-regulation of individual compounds and are usually negative. In a reaction, an increased substrate level will lead to a higher rate (positive reaction elasticity), and therefore to a higher consumption of the substrate itself (stoichiometric coefficient -1). Likewise, a higher product level leads to a smaller rate, and thus decreases its own production. In both cases, the net effect is negative, so fluctuations of individual metabolites tend to be washed out. Despite this stabilizing effect (described by the diagonal elements of the Jacobian), the overall dynamics (described by the Jacobian's eigenvalues) may still be unstable. The dynamics is further shaped by allosteric regulation, which adds elements to the elasticity matrix or changes the existing ones, which affects the Jacobian and the dynamic behavior.

10.2.3.2 Allosteric and Transcriptional Feedback

Organisms employ feedback regulation on various levels of organization. In metabolism, the main mechanisms are transcriptional, posttranslational, and allosteric regulation of enzyme activities. While allosteric regulation via metabolites acts on the metabolic time scale of seconds (determined by metabolite diffusion), transcriptional regulation is much slower: Its characteristic time is set by the protein's effective average lifetime. In growing bacteria with stable proteins, it is given by the cell cycle period,

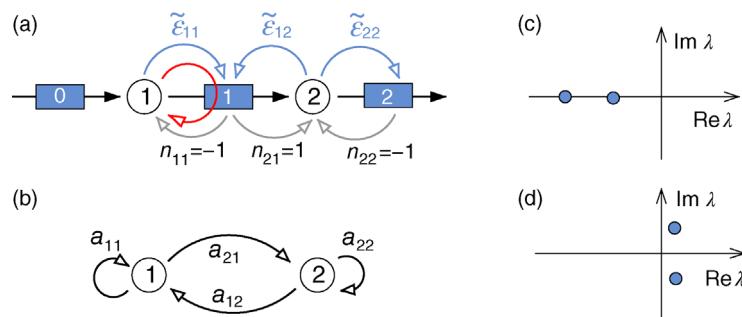
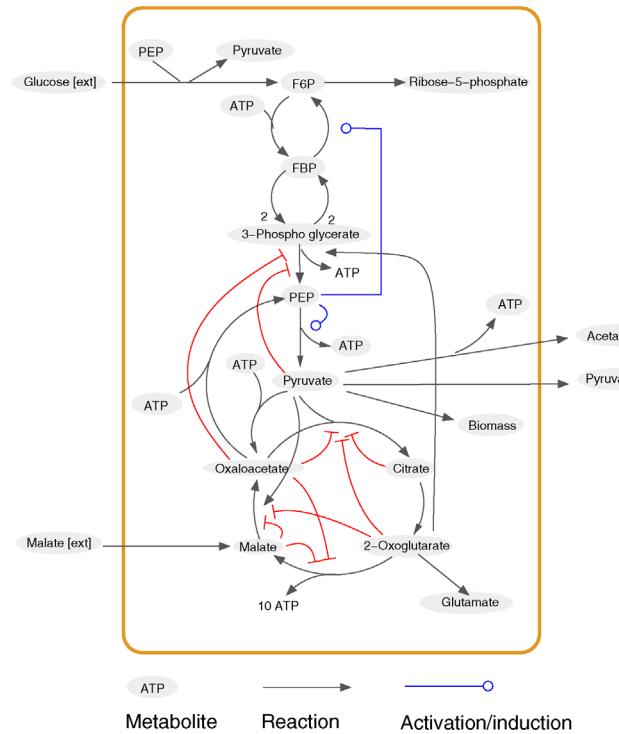


Figure 10.6 Interpretation of the Jacobian matrix. (a) A chain of metabolic reactions (boxes 0, 1, and 2) with a fixed influx v_0 and metabolites (circles). Around a steady state, concentration fluctuations affect the reaction rates via the elasticity coefficients (blue arrows). In the system equation, reaction rates act back on the concentrations via stoichiometric coefficients (gray arrows). (b) Jacobian matrix $\mathbf{A} = \mathbf{N}\tilde{\epsilon}$. Individual matrix elements arise from paths of length 2 in scheme (a): The element $a_{11} = -\tilde{\epsilon}_{11}$, for instance, is the product of $\tilde{\epsilon}_{11}$ and n_{11} on the loop between metabolite 1 and reaction 2 (red arrow). If there are several paths with the same start and end points, the resulting values are added. (c) Eigenvalues of the Jacobian matrix, shown as points in the complex plane. If all real parts are negative, the steady state is stable. (d) Changes in the elasticity matrix (e.g., an addition of allosteric regulation) can change the eigenvalues. The pair of Jacobian eigenvalues shown indicate an unstable steady state, giving rise to spontaneous oscillations.

(a) Allosteric regulation



(b) Transcriptional regulation

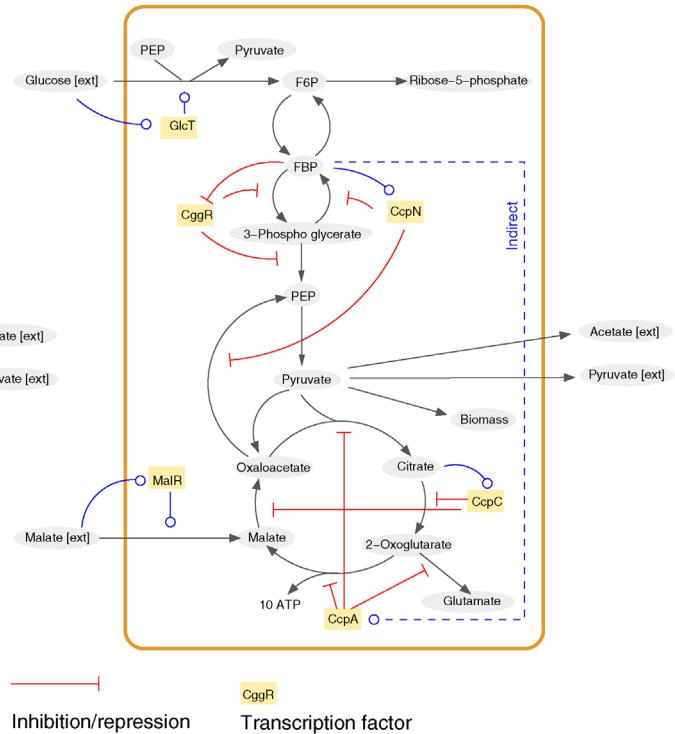


Figure 10.7 Regulation of central metabolism in *B. subtilis*. (a) Metabolic reactions and allosteric regulation in a simplified scheme of glycolysis and citric acid cycle. Only some metabolites are shown. Allosteric regulation by small molecules allows enzyme activities to adapt rapidly to changing concentrations (blue arrows: activation; red arrows: inhibition). (b) The same network, with transcriptional regulation (blue: effective activation; red: effective repression). Transcription factors (yellow boxes), regulated by small molecules, establish a number of feedback loops. The glucose and malate transporters are induced by their own substrates (feed-forward activation by supply). Fructose 1,6-bisphosphate (FBP) acts as a central regulator: It induces gluconeogenesis at low levels and glycolysis when its level is high. (Data from Ref. [31].)

that is, typically more than half an hour. Its delayed and smooth response makes transcriptional regulation less effective. On the other hand, transcriptional regulation is more cost-efficient: When enzymes are not needed, they will not be produced and do not occupy space. Allosteric and posttranslational inhibition, in contrast, decrease an enzyme's activity, but not its concentration – so the effort for producing and maintaining the protein remains (see Section 11.1). Figure 10.7 shows both types of regulation in the central carbon metabolism of the soil bacterium *Bacillus subtilis*. The metabolite fructose 1,6-bisphosphate (FBP) plays a central role in controlling the glycolytic flux: Whenever its level is high, it induces the production of glycolytic enzymes by inhibiting the transcriptional repressor CggR. At low levels, in contrast, it induces enzymes that revert the glycolytic flux and enable gluconeogenesis. This regulation can be seen at work in Figure 9.20. The same metabolite, FBP, has also been found to be a sensor of the glycolytic flux in *E. coli* [30].

10.2.3.3 Integral Feedback

Integral feedback, used to stabilize system outputs, is a standard method in control engineering. The task is to steer a system output toward a defined target value, making the system return to this value under any change of system inputs. It is achieved by continuously sensing the deviation between target value and output, integrating it over time, and feeding the resulting value back into the system as a control.

As an example, consider a linear system whose output variable $y(t) = ku(t)$ depends directly on the input variable $u(t)$. Our goal is to stabilize the output at a value y_0 despite variation in u . To achieve this, a controller continuously measures the difference $\Delta y(t) = y(t) - y_0$, computes its negative time integral $z(t)$, and adds it to the input u (see Figure 10.8). The resulting differential equations read as follows:

$$\begin{aligned} z(t) &= - \int_{t_0}^t y(t) - y_0 dt, \\ y(t) &= ku(t) + k'z(t). \end{aligned} \quad (10.21)$$

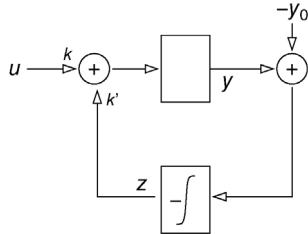


Figure 10.8 Integral feedback controller shown as a wiring diagram.

Now imagine that k and u assume arbitrary static values. In this case, the output y will be static as well and from the stationarity condition $dy/dt = 0$, we directly obtain $y = y_0$, no matter which values for u , k , and even k' we assume. After any change of u , as soon as u becomes constant again, the feedback will steer the system back to its target value y_0 . Importantly, integral feedback does not depend on parameters, but only on the structure of the equation system. Several biochemical pathways, including the yeast osmostress system [31] and the bacterial chemotaxis system [32], which we shall discuss below, implement integral feedback by their structure: This allows them to show perfect adaptation and to maintain this property even when their biochemical parameters are changing. Whether metabolic systems can also be controlled by integral feedback is not clear. For a linear synthesis pathway in which enzymes are transcriptionally controlled by the pathway product, an integral feedback mechanism would require enzymes to be degraded at a constant rate (molecules per time) independent of their own current levels [33] – a condition that cannot be satisfied by cells.

10.2.4

Perfect Robustness by Structure

Integral feedback is a way to realize *perfect robustness*: the fact that a system output y becomes completely independent of some perturbation parameter x . Even in cases where dependencies would be expected (e.g., between the expression level x of a signaling pathway and the pathway's output signal y), perfect robustness can be ensured by special network structures. To see how perfect robustness is realized in biochemical signaling pathways, let us now study models of the bacterial two-component signaling system [34] and the bacterial chemotaxis pathway [36–37].

10.2.4.1 The Two-component System

The two-component system, consisting of a sensor protein and a regulator protein, is a common type of

signaling pathway in bacteria. In the EnvZ/OmpR system in *E. coli*, the membrane-bound sensor EnvZ senses osmolarity and activates the diffusible transcription factor OmpR, which triggers the osmotic stress response. In experiments [38], the system output (in this case, the expression level of target genes of OmpR) was found to be robust against overexpression of the two signaling proteins: It changed by roughly 20% even if the EnvZ and OmpR levels were increased by factors of 10. This robustness clearly improves information transmission: The less an output responds to protein variation, the more precisely it will reflect changes in the pathway's input signal.

As shown in Figure 10.9, signal transduction in the EnvZ/OmpR system consists of three steps: (i) the sensor EnvZ (called X) is phosphorylated under consumption of ATP; (ii) its phosphate group is transferred to the regulator OmpR (called Y); and (iii) the phosphorylated regulator (called Y_P) is dephosphorylated again. A surprising feature of this mechanism, which calls for an explanation, is that the last step requires the presence of ATP, but it does *not* rely on ATP as an energy source. This requirement for ATP has been experimentally shown for the EnvZ/OmpR system, for the envelope stress system CpxA/CpxR in *E. coli*, and for the oxygen limitation system PrrB/PrrA in *Rhodobacter sphaeroides* (see Ref. [34]).

Both properties, the remarkable robustness and the unusual ATP-dependence of the last step, have been explained by a kinetic model [34]. In the model, it is not ATP itself but the first intermediate complex X·ATP that catalyzes the dephosphorylation. As shown in Figure 10.9c, the reactions are broken down into pairs of elementary reactions (1, 1'), (2, 2'), and (3, 3'). Each reaction consists of two mass-action steps: a reversible binding and an irreversible dissociation. The rates read as follows:

$$\begin{aligned}
 \text{Autophosphorylation : } & v_1 = k_1[X][\text{ATP}] - k_{-1}[X \cdot \text{ATP}], \\
 & v_{1'} = k_{1'}(u)[X \cdot \text{ATP}]; \\
 \text{Phosphotransfer : } & v_2 = k_2[X_P][Y] - k_{-2}[X_P \cdot Y], \\
 & v_{2'} = k_2[X_P \cdot Y]; \\
 \text{Dephosphorylation : } & v_3 = k_3[Y_P][X \cdot \text{ATP}] - k_{-3}[X \cdot Y_P \cdot \text{ATP}], \\
 & v_{3'} = k_{3'}[X \cdot Y_P \cdot \text{ATP}]. \\
 \end{aligned} \tag{10.22}$$

The signal is transduced as follows. The external osmolarity signal u sets the rate constant $k_{1'}(u)$, which regulates the autophosphorylation rate: Each value of $k_{1'}$ leads to a certain steady-state level of Y_P , which acts as the system output (blue arrows in Figure 10.9c).

The model allows for a steady-state flux that transfers a phosphate group from ATP to the proteins and finally converts it into inorganic phosphate (see Figure 10.9c). An external stress signal will change the autophosphorylation rate of X, which shifts the steady state and changes the output concentration [Y_P]. The steady-state output

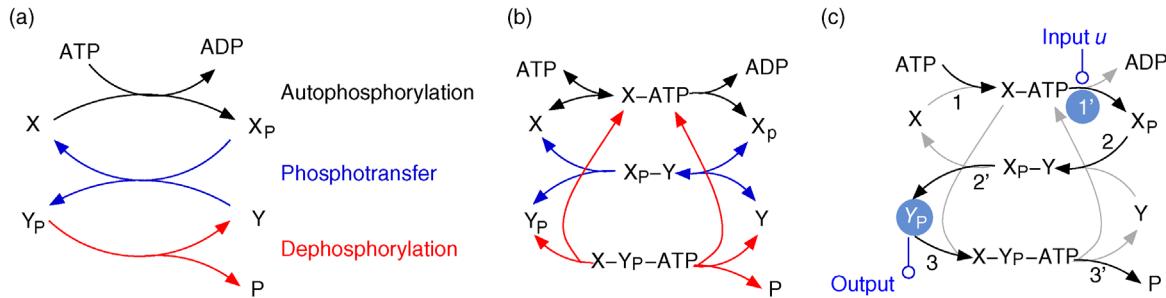


Figure 10.9 Model of the EnvZ/OmpR two-component system in *E. coli* [34]. (a) Basic scheme with three subsequent reactions: autophosphorylation, phosphotransfer, and dephosphorylation. (b) Model with elementary enzymatic steps. The complex X-ATP is needed to catalyze the dephosphorylation. The reversibility of reaction steps is shown by arrowheads. (c) The same model, with the flux of phosphate groups highlighted in black. Numbers denote reaction steps in the model; input and output variables are marked in blue.

$$[Y_P] = \frac{(k_{-3} + k_{3'}) k_1(u)}{k_{+3}} \frac{k_{1'}(u)}{k_{3'}}, \quad (10.23)$$

obtained by equating all rates in Eq. (10.22), depends on the rate constants and on the input u , but not on the total concentrations of X, Y, and ATP. It is perfectly robust with respect to these quantities! Robustness breaks down when the total concentration of Y is below the value from Eq. (10.23): In this case, the flux stops and Y is completely phosphorylated (reaching a lower level than in Eq. (10.23)).

That the model output is robust against the total concentration of X is relatively easy to see: In Eq. (10.22), each term contains exactly one of the concentrations $[X]$, $[X\text{-ATP}]$, $[X_P]$, $[X_P\text{-}Y]$, or $[X\text{-}Y_P\text{-}ATP]$. If we increase the total concentration of X by a factor λ and apply the same factor λ to all these concentrations, but leave all other concentrations (including the output concentration $[Y_P]$) unchanged, the fluxes scale by the same factor. We obtain a new steady state, but with an unchanged output value $[Y_P]$. If the last reaction did not depend on the complex X-ATP, but on ATP alone, this scaling argument would not apply. The most important point of this model is that robustness is *hard-coded* in the system structure: It does not rely on fine-tuned parameters and will therefore not be affected by parameter variation between cells.

10.2.4.2 Chemotaxis Signaling Pathway

The bacterial chemotaxis system, one of the best-studied signaling pathways, shows several levels of robustness and adaptation. Chemotaxis is a mechanism by which bacteria sense chemical attractants (or repellents) and regulate the movement of their flagellae in order to reach regions of high (or low) concentration of these chemicals. We discuss this search strategy in more detail in Section 10.3. Here, we focus on robustness properties of the pathway itself. As the pathway output, we consider the average frequency of tumbling events – moments in which bacteria, by a reverse rotation of the flagella motors, abruptly

change their direction of movement. For chemotaxis stimuli such as glucose, the average tumbling frequency for constant input stimuli is perfectly adapted, that is, independent of the stimulus value even if these values vary over several orders of magnitude. The tumbling frequency itself, however, varies between cells, indicating cell-to-cell variation in the pathway components. Chemotaxis models should reflect this behavior.

Barkai and Leibler [35] built a model of the chemotaxis pathway that implements perfect adaptation by its network structure. In the model, a receptor complex relays a phosphate group to a motile kinase that regulates the tumbling probability of the flagella motors. The receptor complex can switch between an inactive and an active form, and the balance between these forms depends on the ligand concentration. In addition, the complex can be methylated, which increases its kinase activity. By adjusting the methylation state, the sensitivity of the receptor to its ligand can be slowly adapted to typical (i.e., longer term average) ligand concentrations. Perfect adaptation (e.g., insensitivity to the values of constant ligand concentrations) is achieved by a feedback loop in which only active receptor complexes, that is, the form that determines the readout, are demethylated. This mechanism implements integral feedback [32].

This chemotaxis model displays robustness on several levels. First, the output is robust against slow changes of the stimulus. Second, perfect adaptation is hard-coded in the model structure and does not rely on specific, fine-tuned parameters. The pathway also has a third robustness property: Since flagella motors can only sense signals in a certain range, the steady-state output of the pathway (the level of phosphorylated kinase) must be kept in this range even when all pathway proteins are over- or under-expressed. Compared to other network topologies that would have all these robustness features, the network found in *E. coli* is the most simple and, maybe, the most cost-efficient [37].

10.2.5

Scaling Laws

What are the general principles behind structural robustness? Sometimes, robustness properties – for example, with respect to the overexpression of an entire pathway – are tightly related to a system's scaling behavior, for instance, to the fact that a change of time units leads to a proportional change of flux values while leaving the concentrations unchanged. To clarify how these things are related, some general words about scaling laws are in place.

Could one imagine a mouse of the size of an elephant? What if an organism lived much faster, or much more slowly than it actually does? In fact, animals' anatomies and physiologies are strongly shaped by their size and by the time scales of physiological processes. Scaling, a type of symmetry operation, plays an important role in physics. A symmetry operation describes the changes in a system under some transformation. If a rotation or mirror reflection leaves a geometric shape unchanged, the shape is *symmetric* or *invariant* with respect to that transformation. Also physical laws can be symmetric, that is, retain their form in different frames of reference (e.g., after a rotation of the coordinate system) or under a change of measurement units (e.g., the steady-state condition for biochemical systems, which is independent of the choice of time units).

Scale transformations can have two interpretations: We can imagine that a quantity either increases nominally due to a change in the physical units (e.g., 1 h → 60 s) or it is enlarged in reality (e.g., 1 s → 60 s). The change in numbers is equal in both cases. Since a change of units is always allowed, we can obtain formulas for rescaled systems (e.g., one in which all processes occur 60 times more slowly) by first considering a change of units and

then reinterpreting the resulting numbers as the result of an actual scaling.

10.2.5.1 Geometric Scaling

Geometric shapes satisfy simple scaling laws under spatial scale transformations. If L is the side length of a cube, the square areas scale as L^2 and the cube volume scales as L^3 . Geometric scaling implies integer exponents (see Figure 10.10). Shapes such as the fractal Sierpinski triangle (Figure 10.10c), which resemble their own rescaled parts, are called *self-similar*. If the side length of a Sierpinski triangle doubles, its “area” (which is not a normal two-dimensional area) is increased by a factor of 3 (larger than 2, for the length of a line; and smaller than 4, for a square area). Its scaling exponent $\log_2 3 = \log 3 / \log 2$ can be seen as a noninteger measure of dimensionality. Geometric shapes with such noninteger scaling exponents are called fractals.

10.2.5.2 Power Laws

Not only geometric sizes, but also many other quantities in nature are related by power laws:

$$y(x) = y_0(x/x_0)^\alpha. \quad (10.24)$$

Power laws, defined by reference values x_0 and y_0 and a scaling exponent α , describe simple scaling behavior. In geometric scaling, the exponent is given by an object's dimensionality. In a double-logarithmic plot $\log y$ against $\log x$, a power law translates into a linear relationship $\log y = \alpha \log x + \text{const}$. The type of logarithm (e.g., natural or decadic) can be freely chosen, and different choices will correspond to different values of the prefactor. A typical feature of power laws is their property of being scale-free: If we stretch the x -axis by a factor of λ and rescale the y -axis by λ^α , the function looks like before. This is unique to power laws. A Gaussian or an

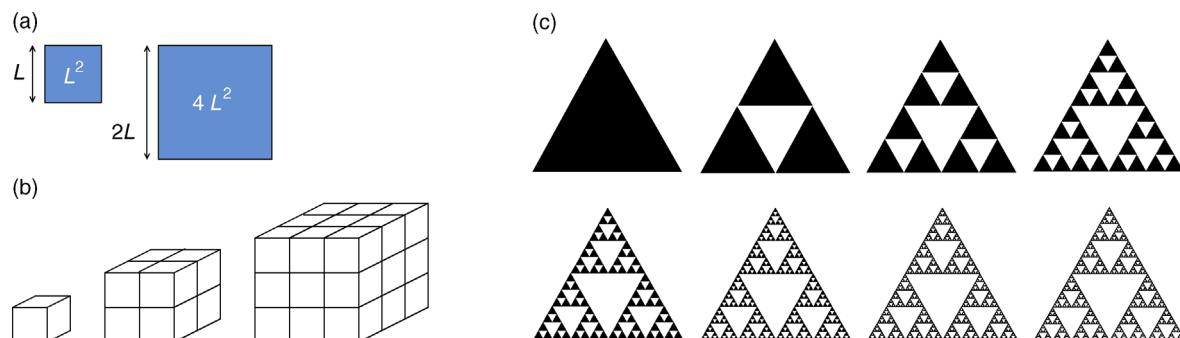


Figure 10.10 (a) The area of a square scales quadratically with its side length. (b) A cube volume V scales with the side length as $V \sim L^3$. (c) The Sierpinski triangle, a self-similar fractal, consists of three smaller copies of itself. To define it, we start from a triangle, combine three copies of the triangle, and scale the side length by one half. If we iterate this process infinitely many times, we obtain the Sierpinski triangle. Being a fractal, it shows nongeometric scaling behavior: When its side length is doubled, its area increases by a factor of 3 (more than 2, like a line, but less than 4, like a normal triangle). Its dimensionality is given by $\log_2 3$.

exponential function would have a characteristic width, which defines a scale on the x -axis. Power laws, in contrast, are scale symmetric and cannot be used to define a length scale on the x -axis.

10.2.5.3 Scale Invariance

That a quantity y is independent of some scale parameter x can be formally stated by a power law $y(x) \sim x^0$ with exponent 0. Scale-invariant quantities such as this can be constructed from quantities with a known power-law scaling: For instance, if surface area A and volume V of a geometric body scale as $A \sim L^2$ and $V \sim L^3$, the quantity $r = A^{1/2}/V^{1/3}$ is scale invariant and independent of L . Scale invariance can be inferred from physical units: Since area and volume are measured in m^2 and m^3 , respectively, r is dimensionless and therefore scale invariant. Again, this means two things. First, the numerical value of r is not affected by any change of length units. Second, it is also not affected by an *actual* enlargement or shrinkage of the body. Laws that concern dimensionless quantities (such as Mach numbers in fluid dynamics) hold universally for systems of different sizes. The same logic can be applied, for instance, to time scaling.

10.2.5.4 Allometric Scaling

Power laws play an important role in physiology. Many physiological quantities, such as the metabolic rates and lifespans of animals, are related to the body mass x by allometric laws of the form (10.24). The same laws hold for organisms of various sizes, and sometimes even for their individual cells. Most surprising, however, are the observed scaling exponents. Based on geometric scaling (with body mass being proportional to body volume, scaling geometrically as L^3), we would expect scaling exponents to be multiples of 1/3. However, the observed scaling exponents are typically multiples of 1/4. Geoffrey West has explained this nongeometric scaling by a universal model of blood vessels and of other circulation systems that transport nutrients and oxygen (in animals) or water (in plants) [39]. The self-similar, fractal geometry of such circulation systems and a few general assumptions about how they are physically operated can explain a variety of observed scaling exponents.

The allometric scaling laws have far-reaching consequences. For instance, the metabolic rates of cells do not depend on the cells themselves, but on the size of the body in which they exist. Even though small and large animals, such as mice and whales, differ strongly in their physiological properties, many of these properties can be derived, to a good approximation, by rescaling a single, ideal “standard organism.” The theory of fractal scaling exponents also predicts a universal law for the speed of

growth during an animal’s lifetime. Again, to make this law visible in data, ages and body masses of animals must be plotted in a suitable allometrically scaled form [40].

10.2.5.5 Scaling Relations within Cells: Ribosome Content and Growth Rate

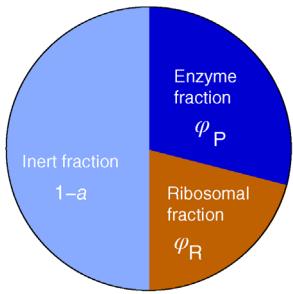
Scaling relations also exist on the cellular level, for instance, between cells’ growth rates and other physiological properties [41]. Some of these relations do not have the form of power laws. Let us have a look at a well-studied example, the cellular ribosome content. Based on some simple assumptions, we can expect that fast-growing cells should show a larger fraction of ribosomes in their proteome. Since growing cells must duplicate all of their components, each ribosome must produce, on average, one new ribosome per cell cycle. If this self-replication takes a fixed time τ , ribosomes will be busy producing ribosomal proteins during a fraction τ/T of the cell cycle T ; in the remaining fraction of time, $1 - \tau/T$, they produce other proteins. Now, if the total protein amount in cells is constant and ribosomes and other proteins are diluted, but not degraded, the mass percentage of ribosomal proteins will exactly reflect the percentage of time in which ribosomes are produced. At higher growth rates, this fraction τ/T increases (becomes T becomes smaller), and so does the ribosome content. Qualitative relations of this sort (faster growth \rightarrow higher ribosome content) have been observed experimentally [42]. However, this rule holds only when the growth rate is controlled by nutrient limitation; in contrast, when growth is reduced by translation-inhibiting drugs, the opposite relation is observed: The ribosome fraction increases at *lower* growth rates (Figure 10.11).

The two contrary relationships between growth rate and ribosome fraction are accurately described by phenomenological sector models [43]. The basic model is based on two postulates: (i) The total proteome of a cell, normalized to a mass of 1, consists of four fractions φ_Q (constant fraction of nonribosomal proteins), φ_R^{\min} (constant ribosome fraction), φ_R (variable ribosome fraction), and φ_P (variable metabolic enzyme fraction). The sum a of the two variable fractions is assumed to be constant and independent of growth. (ii) Since faster growing cells must produce more precursors and proteins per time, each of the two variable fractions, by itself, must scale proportionally with the growth rate. We thus obtain three equations:

$$\varphi_R + \varphi_P = a, \quad \lambda = \beta_R \varphi_R, \quad \lambda = \beta_P \varphi_P, \quad (10.25)$$

with constant sum $a = 1 - \varphi_Q - \varphi_R^{\min}$ and coefficients β_R and β_P called translational and nutritional capacity,

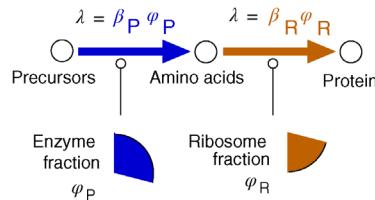
(a) Protein fractions in the sector model



Assumption 1: Enzymes and ribosomes occupy a fixed mass fraction of the proteome

$$a = \varphi_P + \varphi_R \quad (a: \text{Available proteome fraction})$$

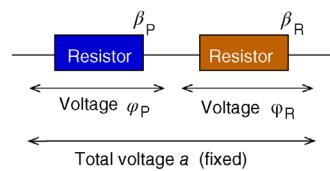
(b) Schematic model of cell growth



Assumption 2: The growth rate is proportional to each of the two proteome fractions

$$\lambda = \beta_P \varphi_P = \beta_R \varphi_R \quad \begin{matrix} \beta_P & \text{Nutrient capacity} \\ \beta_R & \text{Translation capacity} \end{matrix}$$

(c) Analogy to electric circuit



$$\varphi_P = \frac{\beta_R}{\beta_P + \beta_R} a \quad \varphi_R = \frac{\beta_P}{\beta_P + \beta_R} a$$

$$\lambda = \frac{\beta_P \beta_R}{\beta_P + \beta_R} a$$

Figure 10.11 Dependence of cells' ribosome content on the growth rate. (a) In a sector model [44], a cell's protein content is split into four fractions. Two of them, the variable metabolic enzyme fraction φ_P and the variable ribosome fraction φ_R , are assumed to have a constant sum a . The remaining two fractions φ_Q and φ_R^{\min} are depicted as one inert fraction $1 - a$. (b) The cell growth rate λ is proportional to each of the fractions φ_P and φ_R . The prefactors β_P and β_R (called capacities) depend, respectively, on nutrient quality and on the efficiency of translation. (c) From the three equations, we obtain the fractions φ_P and φ_R and the growth rate λ as functions of the β_P and β_R . Formally, the same equations describe an electric circuit with two resistors. Changes of the electric conductivities (β_P and β_R) will affect the voltages φ_P and φ_R and the current λ . The relationship between φ_R and λ (in the cell, between ribosome fraction and growth rate) depends on which of the conductivities is varied.

respectively. Solving Eq. (10.25) for the protein fractions φ_R and φ_P and the growth rate λ , we obtain

$$\begin{aligned} \varphi_R &= a \frac{\beta_P}{\beta_P + \beta_R}, \quad \varphi_P = a \frac{\beta_R}{\beta_P + \beta_R}, \\ \lambda &= a \frac{\beta_P \beta_R}{\beta_P + \beta_R}. \end{aligned} \quad (10.26)$$

Formally, the same equations could also describe an electric circuit with two resistors (where λ corresponds to the current, β_R and β_P to the resistors' conductivities, φ_R and φ_P to the voltages, and a to the total voltage, assumed to be fixed). Noting that resistances (i.e., inverse conductivities) in serial connection are additive, we immediately obtain the solution – Eq. (10.26). The formulas lead to testable predictions. Since most of the RNA in *E. coli* is ribosomal RNA, the measurable RNA/protein ratio can be seen as a proxy of the ribosome content. In the model, the ribosome content can be expressed in two ways, as $\varphi_R^{\min} + \varphi_P$ or $\varphi_R^{\max} - \varphi_P$ (where $\varphi_R^{\max} = \varphi_R^{\min} + \varphi_P$). With a proportionality factor α between RNA/protein ratio and total ribosome fraction, Eq. (10.26) leads to two laws for the RNA/protein ratio r :

$$r = \alpha \left[\varphi_R^{\min} + \frac{\lambda}{\beta_R} \right], \quad r = \alpha \left[\varphi_R^{\max} - \frac{\lambda}{\beta_P} \right]. \quad (10.27)$$

Each of these laws relates the RNA/protein ratio to the cell growth rate, but each assumes a different cause for growth rate changes. In the first law, we assume a variation of the nutritional capacity β_P (by using different

nutrients); this leads to a joint variation in $r(\beta_P)$ and $\lambda(\beta_P)$, while the translational capacity β_R remains fixed. The second law describes joint variations caused by changes in translational capacity β_R (achieved by applying a translation-inhibiting drug in different concentrations); this time, the nutritional capacity β_P remains fixed. Both laws are in good agreement with experimental data [43].

Could the observed growth laws be explained by mechanistic cell models in which gene expression and metabolism compete mechanistically for energy, ribosomes, and protein resources? Weiße *et al.* [41] presented a coarse-grained cell model that does exactly this. With effective model parameters fitted to empirical relations between growth rate and ribosomal mass fraction, and based on mechanistic competition for resources, the model captures three fundamental types of behavior observed in bacteria: the hyperbolic relation between growth rate and extracellular nutrient levels (Monod's law), and the two growth laws – the positive and negative linear relationships between growth rate and ribosomal mass fraction.

Coming back to scaling laws in general, the growth laws in Eq. (10.27) differ from power laws as in Eq. (10.24) not only by their mathematical form but also in the way they are typically used. In power laws, the focus is on scale symmetry rather than on a specific scale; it is symmetry that enables us, for instance, to use smaller systems as models of larger ones. Accordingly, one usually studies the exponents and the relations between different power laws. The coefficients and offsets in growth

laws, in contrast, define a very specific scale, which makes these laws more comparable to Ohm's law or gas laws than general scaling laws.

10.2.6

Time Scaling, Summation Laws, and Robustness

The scaling properties of a system can tell us about the system's robustness, in particular about the parameter sensitivities $\partial y / \partial x_l$ between a system output y and parameters x_l . Important summation rules for these sensitivities can be derived from a simple scaling argument. A mathematical function $f(x_1, \dots, x_n)$ that satisfies a scaling relation

$$f(\lambda x_1, \dots, \lambda x_n) = \lambda^k f(x_1, \dots, x_n) \quad (10.28)$$

is called homogeneous with degree k . The power law Eq. (10.24), for instance, is a homogeneous function with degree γ ; however, a homogeneous function need not be a power law. If we differentiate Eq. (10.28) by the scaling factor λ and then set $\lambda = 1$, we obtain a sum rule for the normalized derivatives:

$$k = \sum_l \frac{x_l}{f} \frac{\partial f}{\partial x_l} = \sum_l \frac{\partial \ln f}{\partial \ln x_l}. \quad (10.29)$$

This is *Euler's theorem* for homogeneous functions. If a quantity f is scale invariant (i.e., $k = 0$), the normalized derivatives sum to 0; for linear scaling ($k = 1$), they sum to 1. In the latter case, f can never – under no conditions! – be robust to all the x_l . Euler's theorem is a powerful tool for the study of complex systems: With its help, we can draw general conclusions about the outputs of complicated and even unknown systems whenever a relation Eq. (10.28) for time or space scaling can be established.

10.2.6.1 Time Scaling and Metabolic Control

The summation theorems of metabolic control theory are examples of the sum rule (10.29). Consider a metabolic model with rate laws $v_l = E_l v'_l(\mathbf{s})$ containing the enzyme concentrations E_l as prefactors (where \mathbf{s} contains the reactant concentrations). If all enzyme levels are scaled by a factor λ , the reaction rates increase by the same factor. By replacing $E_i \rightarrow \lambda E_i$, we obtain the system equation $d\mathbf{s}/dt = \lambda \mathbf{Nv}(\mathbf{s})$ or, equivalently, $d\mathbf{s}/d(\lambda t) = \mathbf{Nv}(\mathbf{s})$, as if time were rescaled by a factor λ . Since this time scaling argument also holds for steady states, the steady-state concentrations and fluxes, as functions of the enzyme levels, are homogeneous functions: A doubling of all enzyme levels will double the fluxes (homogeneous function with degree 1) and leave the concentrations

unchanged (homogeneous function with degree 0). This holds generally, no matter how complicated the functions are.

Since the steady-state concentrations s_i^{st} are invariant against time scaling, they must satisfy summation theorems of the form (10.29), where $f = s_i^{\text{st}}$, $x_l = E_l$, and $k = 0$. Their scaling law with exponent $k = 0$ shows that steady-state concentrations are perfectly robust against a uniform relative overexpression of all enzymes. The steady-state fluxes, in contrast, scale linearly with time, so they satisfy Eq. (10.29) with $f = J_i$, $x_l = E_l$, and $k = 1$. In summary, the normalized response coefficients, which are identical to normalized control coefficients, obey the summation theorems (see Section 4.2.2):

$$\begin{aligned} \sum_l \frac{\partial J_i}{\partial E_l} \frac{E_l}{J_i} &= \sum_l C_{v_l}^{J_i} = 1, \\ \sum_l \frac{\partial s_i}{\partial E_l} \frac{E_l}{s_i} &= \sum_l C_{v_l}^{S_i} = 0. \end{aligned} \quad (10.30)$$

The equivalence between enzyme scaling and time scaling remains valid even if we consider dynamic behavior. Consider, for instance, a signaling pathway that translates a sudden jump in its input signal into some time-dependent kinase activity. If the output curve is peak shaped, it can be characterized by features such as peak time τ or the maximal peak height. Time scaling would lead to a linear scaling of τ and leave the peak height unchanged: The resulting summation theorems for system parameters resemble, respectively, the summation theorems for fluxes and concentrations in steady states [45]. We can even go one step further. Consider a metabolic system with fixed enzyme levels and some intrinsic dynamic behavior $s_i(t)$. Now assume that all enzyme levels are proportionally varied with some *time-dependent* factor $\lambda(t)$. This variation will be equivalent to a *time-dependent rescaling* of time. Even in this case, the metabolic dynamics stays the same as before, but proceeds at a different, time-dependent speed. As long as enzyme levels vary proportionally, the metabolic state at a certain time point does not depend on the shape of the enzyme time curves, but only on the time integral of the enzyme levels before that time point. This is called the average enzyme principle [46].

10.2.6.2 Robustness against Correlated Expression Changes

A main source of variation in cells, which can make pathway outputs unreliable, is the inevitable variation in protein expression. Even if expression noise cannot be suppressed, there is an efficient way to stabilize the system output: ensuring that protein levels fluctuate only proportionally and that system outputs are robust against

such proportional fluctuations. The first condition, correlated expression changes, can be met if the proteins that constitute a pathway are encoded in one bacterial operon and are therefore produced from the same mRNA transcripts. For the second condition, suitable robustness mechanisms in the pathway must ensure that the effects of overexpressed proteins cancel out.

With y being the (presumably robust) output and x_l the protein levels, robustness against a proportional overexpression implies that the sum $\sum_l[(\partial \ln y)/(\partial \ln x_l)]$ of scaled sensitivities must vanish. In the case of metabolic pathways (where the sensitivities are called control coefficients, and where we assume fixed external metabolite concentrations), we already saw that this holds automatically whenever y is a concentration, and that it never holds if y is a flux. In some signaling systems, robustness can be guaranteed by suitable pathway structures (as in the two-component system or in the bacterial chemotaxis system discussed above). General criteria for structural robustness can be derived from both chemical reaction network theory [47] and metabolic control analysis [48].

10.2.6.3 Temperature Compensation

A special robustness task arises because the rates of chemical reactions vary with temperature. Since activation enthalpies are usually positive, an increase in temperatures tends to speed up reactions, but not necessarily proportionally. In a metabolic pathway, an abrupt temperature change would perturb the profile of stationary metabolite concentrations and a readjustment of enzyme levels would take very long. In glycolysis, however, the observed concentration profiles depend on substrate availability, but are practically unaffected by temperature changes [49]. This suggests that the glycolytic enzymes are specifically tuned to show the same temperature dependence in their catalytic constants, probably as an outcome of an evolutionary adaptation of kinetic properties.

Temperature compensation plays an important role in molecular clocks, that is, biochemical pathways supposed to produce oscillations at a fixed reliable frequency. To enable temperature compensation, the scaled response coefficient $R_T^y = \partial \ln y / \partial \ln T$ between system output (in this case, the clock frequency) and temperature must vanish: This coefficient is a sum $R_T^y = \sum_l C_l^y[(\partial \ln v_l)/(\partial \ln T)]$ over all control coefficients, weighted by the respective temperature elasticities $(\partial \ln v_l)/(\partial \ln T)$. The latter are usually positive, so temperature compensation requires that some of the control coefficients must be negative. In other words, the system must be set up in such a way that some biochemical processes have positive effects on the system output, while others have negative effects. Together, all the

effects of a changing temperature, mediated through these processes, must cancel out [50]. In a model of the *Neurospora crassa* circadian clock in Ref. [51], a proposed mechanism for temperature compensation assumes that an increasing temperature increases the translation of the FRQ protein, but simultaneously decreases its nuclear import rate. If a system's internal processes are enzyme catalyzed, the opposing control coefficients of the enzymes may look paradoxical: At first sight, the system appears like a car driver who brakes while pressing the accelerator. However, if we suspect a need for temperature compensation, these control properties find a simple explanation.

10.2.6.4 Limits of Robustness

As we saw above, biochemical systems cannot be completely robust. The summation theorems, for instance, tell us that metabolic fluxes remain sensitive to enzyme abundances no matter which regulation systems are used: The sum of control coefficients is always 1. Nevertheless, flux control can be redistributed among enzymes by a change of kinetics or additional allosteric feedbacks. Trade-offs between different robustness properties are common: Feedback loops that make a system robust against low-frequency parameter variation can make it more susceptible to high-frequency noise [23]. Robust design shifts robustness to pairs of parameters and outputs that need to be most robust (e.g., pairs that contribute most to fitness according to Eq. (10.16)). Robustness trade-offs can also occur between different levels of organization: For example, cells that achieve robustness by maintaining a constant metabolic state will provide a favorable environment to intracellular parasites, and thus become fragile against infection. These trade-offs may suggest a “summation law” for robustness: a fixed amount of robustness that is redistributed, but never changed. However, this does not seem to be the case, at least not in metabolic pathways: If we interpret the inverse flux control coefficients $1/C_{v_j}^J$ as robustness values, changes in the system leave the sum of control coefficients constant, while the sum of robustness values will change [52].

10.2.7

The Role of Robustness in Evolution and Modeling

10.2.7.1 Robustness and Evolution

We saw that many traits of organisms, and the very existence of many proteins and pathways, can be explained by a need for robustness [23]. While robustness can be explained mechanistically as a dynamic phenomenon, it also directs evolution, either by direct fitness advantages or by contributing to evolvability.

- *Direct fitness advantage* If there is an ideal behavior that maximizes biological fitness, any deviation from this behavior will cause fitness losses; such deviations should be suppressed. Which robustness properties are most important depends on the typical changes in an organism's environment, on how perturbations spread in the organism, and on the severeness of certain output deviations. Some variables in cells can vary widely without doing any harm, while others have to be controlled very carefully. If robustness mechanisms are costly (e.g., require additional protein to be expressed), they will evolve only if organisms are challenged frequently enough by critical perturbations. Intracellular parasites, which live under relatively constant conditions, are less reliant on robustness mechanisms and can live with simpler networks and smaller genomes. Finally, the trade-off between robustness and cost economy may be overlaid by trade-offs between opposing robustness properties.
- *Evolvability* Robustness in physiology can make a population or species better evolvable, that is, capable of creating inheritable phenotypic variation [53]. This can be of vital importance, especially after drastic changes in external conditions or under unfavorable environments. If an organism contains dynamically and functionally robust modules (e.g., proteins, pathways, or organs), these modules can behave, and therefore evolve, relatively independently. In this way, populations can explore variants of the modules without compromising their fitness, until at some point some new combination provides a fitness advantage. On the contrary, robustness can imply that genotypic changes have little or no phenotypic effects at all. In periods of fast evolution, this can be an obstacle.

10.2.7.2 Robustness and Modeling

The notion of robustness links three important aspects of a biological system: its environment (which parameters show strong variation?), its dynamics (how does parameter variation affect the system outputs?), and its biological function (how does variation of the output affect fitness?). Sensitivities, predicted from models, can tell us about the weak points of a system, such as potential oncogenes [54] or drug targets. Moreover, robustness and sensitivity play key roles in parameter estimation: If an observable y is insensitive to a parameter x , measurements of y will not be useful for estimation of x . At the same time, even wrong estimates of x will still allow for good predictions of y .

But apart from model fitting, robustness properties can also be important pieces of information and even a precondition for modeling. On the one hand, known or suspected robustness properties can be used as criteria in model selection and, thus, for revealing mechanistic

details of a system. Eldar *et al.* [55], for example, generated many variants of a morphogene model with sampled parameters. By selecting exactly those models that met a certain robustness criterion, they could delimit possible parameters and determine correlations between them (for more examples, see Refs [34,35,37]).

On the other hand, systems that are very nonrobust may also be hard to model. On the contrary, robustness of the real system can make our models failproof against wrong parameter choices, wrong initial conditions, and even numerical errors. The same robustness properties that allow biological modules to remain functional while varying genetically will also allow us to describe organisms *as if* they were modular. Think of parameters x that describe environmental perturbations or model details we would like to neglect: If the system outputs are robust to such parameters, many simplifications or wrong assumptions about boundary conditions will have little effect on our results. In this case, we can use our phenomenological descriptions and sweep many details of reality under the carpet.

10.3 Adaptation and Exploration Strategies

Summary

Due to limited and noisy signaling systems, cells have only relatively little information about their environment and in particular about future events. In facing this uncertainty, cells may choose actions (e.g., protein expression levels) that maximize their fitness *on average*. Diversification can improve the population's chances to withstand sudden external changes. Subpopulations that are imperfectly adapted to current conditions, but ready to survive possible catastrophes, can be seen as an "insurance policy." Such subpopulations can be stably maintained by phenotypic random switching. We compare this switching strategy with direct adaptations on the single-cell level. In both cases, Shannon information – which quantifies the information transmitted through pathways, but also the missing information about future environmental changes – plays a central role.

In the previous sections, we saw how variability can emerge in cells, how it propagates through cellular networks, and how it can be suppressed by robustness mechanisms. However, instead of avoiding randomness, cells may also profit from it. Chemical noise can enable them to take random decisions, for instance, during chemotaxis, which will be discussed below. It also allows them to establish heterogeneous subpopulations, which can limit the risk of extreme losses (e.g., the entire

population going extinct) at the price of a constant fitness loss (a decreased average growth rate). In analogy to financial investments, such strategies are called “bet hedging.” As discussed in Section 15.4, randomness can be something that is in the eye of the beholder. Here, we call processes “random” if they cannot be predicted in practice. This also applies to cells. How predictable things are (e.g., an environmental variable, given the intracellular signals received by a cell) can be quantified by Shannon’s notion of information.

10.3.1 Information Transmission in Signaling Pathways

To understand how signaling pathways function, we need to understand the role of uncertainty and noise. Imagine you had to reconstruct the input signals of a signaling pathway from its output signals. Since there is chemical noise and since the expression level of the pathway itself may vary, this reconstruction will not be precise; however, knowing the output may at least decrease the uncertainty about the input; this decrease is quantified by the Shannon information transmitted in the pathway (Section 15.6). Generally, Shannon information quantifies the information transmitted between a sender (input) and a receiver (output) [57]: We assume that input signal X and output signal Y are randomly distributed and statistically dependent. The information about X contained in Y is quantified by the Shannon information (or “mutual information”):

$$I(X, Y) = H(X) + H(Y) - H(X, Y), \quad (10.31)$$

where $H(X)$ and $H(Y)$ are the individual Shannon entropies of X and Y and $H(X, Y)$ is the Shannon entropy of their joint distribution (see Figure 10.12). If X and Y are statistically independent, Y contains no information

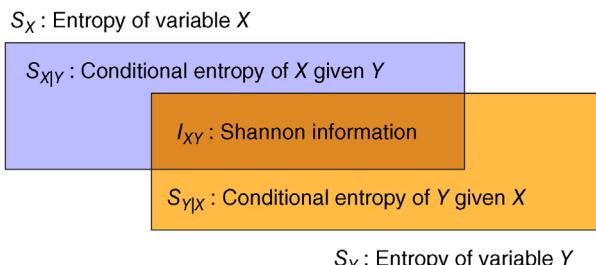


Figure 10.12 Shannon information. Two correlated random variables X and Y can be characterized by their individual entropies S_X and S_Y (rectangle areas) and their joint entropy S_{XY} (total area). The Shannon information (intersection area) is given by $I_{XY} = S_X + S_Y - S_{XY}$, and the conditional entropies are given by $S_{X|Y} = S_{XY} - S_Y$ and $S_{Y|X} = S_{XY} - S_X$.

about X : In this case, the entropies are additive ($H(X, Y) = H(X) + H(Y)$) and the Shannon information $I(X, Y)$ vanishes. On the contrary, if Y is fully dependent on X , then $H(X, Y) = H(X)$ and the Shannon information $I(X, Y) = H(Y)$ becomes maximal. As we shall see, Shannon information characterizes only the quality of a communication channel – It does not tell us whether the transmitted information is useful or not.

How can this concept be applied to signal transduction pathways? If a pathway translates an input (e.g., an extracellular ligand concentration) into an output (e.g., a transcription factor activity in the nucleus), the output, even for a fixed input, may vary widely across cells because of variation in the pathway’s expression level. In turn, cells with the same output may have experienced different input signals. The fact that a known output signal makes certain input values more likely than others can be quantified by Shannon information. For the bacterial quorum sensing pathway [58] and for the TNF-NF- κ B signaling pathway in mouse fibroblasts [59], the Shannon information has been computed from distributions of output signals at given input values. The TNF-NF- κ B pathway senses the level of extracellular tumor necrosis factor (TNF) and controls the level of nuclear NF- κ B, its measurable output. It was found that this pathway transmits enough information for accurate binary decisions and that additional feedback regulation could change its information transmission properties.

10.3.2 Adaptation and Fold-Change Detection

Adaptation to previous input signals is an important way to improve accuracy in signal transduction. Signaling systems should be sensitive to specific stimuli, but not to other, irrelevant features of their input. It is often not the absolute scaling of input signals but their temporal shape that matters. In this case, the absolute size of the input signal should be filtered out by adapting the system to the signal’s baseline level. Another scope for adaptation is in the fact that sensory systems have limited dynamical ranges: Our retina, for instance, can distinguish a certain range of light intensities. When light is turned on, we are blinded and cannot see clearly. However, our pupils contract and adjust the incoming light to our retina’s sensitive range. Brightness differences between objects remain visible, while the overall brightness of our visual field is filtered out. In a similar way, sensory systems in cells can become habituated to recurrent or slowly changing stimuli and thereby more sensitive to new or fast changing stimuli. Possible mechanisms, such as the adaptation

motif and the incoherent feed-forward loop, have been described in Section 8.2.

A special type of adaptation performed by sensory systems is *fold-change detection* [60]: The system responds only to *relative changes*, while the absolute scaling of the signals (which is supposed to change rather slowly) does not matter and is filtered out. Systems that perform fold-change detection show perfect adaptation (i.e., a fixed output value for stationary inputs) and satisfy Weber's law (the maximal response to a signal change is inversely proportional to the background signal). Fold-change detection can be implemented by a number of regulation mechanisms, including incoherent feed-forward loops, nonlinear integral feedback, and logarithmic input functions followed by linear feedback systems.

10.3.3

Two Adaptation Mechanisms: Sensing and Random Switching

Microbes face various challenges (e.g., by antibiotics, changing nutrient supplies, and attacks by immune cells), which necessitate the expression of specific proteins. Since proteins are costly, they should be expressed only when necessary, but the challenges are largely unpredictable. Therefore, cells must adapt. Different adaptation mechanisms would be conceivable: Single cells may sense cues from the environment and trigger responses, or cell populations may maintain subpopulations of cells prepared for possible changes. Examples of these two strategies are, respectively, bacterial quorum sensing [58] and bacterial persistence [61].

10.3.3.1 Random Switching in Cell Populations

Let us say a bit more about the second strategy. Two genes that inhibit each other can form a bistable switch (see Example 6.9 and Figure 2.2), and noise in gene expression can induce spontaneous transitions between the two states. Controlling other processes, genetic switches can trigger "decisions" between different types of behavior, for example, between the lysogenic and the lytic state in the λ phage [62]. Such spontaneous switching can generate diversity in cell populations. Clonal populations of *E. coli*, for instance, contain small subpopulations of *persister* cells that are less susceptible to antibiotics but grow more slowly [63]. If persisters were genetically different from other cells, they could be outcompeted and would disappear from the population. This would make the population vulnerable in the long run. A mechanism based on random switching rather than genetic differences can avoid this problem: All cells share the same genotype, so persister cells cannot be genetically outcompeted [61]. We can think of the

Example 10.2 Population Model for Bacterial Persistence

Kussell *et al.* studied the fitness advantage of phenotypic random switching, considering long-term growth in an alternating environment as the evolutionary selection criterion [64]. In the model, a bacteria population is described by cell numbers n (normal cells) and p (persisters). Cells switch randomly between the two states with constant transition rates a (normal \rightarrow persister) and b (persister \rightarrow normal), and replicate with effective rates μ_n and μ_p . The subpopulation sizes follow the equation

$$\frac{d}{dt} \begin{pmatrix} n \\ p \end{pmatrix} = \begin{pmatrix} \mu_n - a & b \\ a & \mu_p - b \end{pmatrix} \begin{pmatrix} n \\ p \end{pmatrix} \quad (10.32)$$

or, in vector notation,

$$\frac{dx(t)}{dt} = Ex(t). \quad (10.33)$$

The environment alternates between periods of normal growth and periods with antibiotics treatment. The presence of the antibiotic affects the reproduction rates: μ_n is usually positive, but negative under the effect of antibiotics; μ_p is always positive, but small. By inserting the reproduction rates for both cases into the matrix E , we obtain the matrices G (for normal growth) and S (for antibiotics conditions). During a growth period, the population vector x evolves as

$$x(t) = e^{t_g G} x(0). \quad (10.34)$$

This linear equation can be solved by matrix exponentials. A matrix exponential e^M is a matrix with the same eigenvectors as the matrix M , but with eigenvalues e^{λ_i} , where the λ_i are the eigenvalues of M . Under alternating conditions with durations t_g and t_s , the population vector x evolves as

$$x(t) = \dots e^{t_s S} e^{t_g G} e^{t_s S} e^{t_g G} x(0). \quad (10.35)$$

Through repeated multiplication with the matrix $Q = e^{t_s S} e^{t_g G}$, the cell population will grow, overall, at a rate given by the maximal eigenvalue of Q , which can be easily computed. Taking this rate as an objective to be maximized, optimal switching rates a and b can be determined. Under a number of approximations, the optimal values read $a \approx 1/t_g$ and $b \approx 1/t_s$. These rates match exactly the switching rates of the environment [64].

persisters as an insurance policy taken out by the cell population.

In the model of Kussell *et al.*, phenotypic random switching can provide a long-term advantage. However, how can the population avoid that the switching mechanism gets lost in phases without antibiotics treatment? In

the absence of antibiotics, a mutant that loses the switching mechanism and does not produce persisters would have a selection advantage. However, the fraction of persister cells in a switching cell population is so small that it has almost no effect on a population's average growth rate. Accordingly, the selection advantage of nonswitching mutants will be negligible, and bacterial populations can maintain the switching mechanism even if they are not challenged with drugs for a long time. Very rare phases of antibiotics challenge will suffice for the mechanism to be conserved.

10.3.3.2 Phenotypic or Responsive Switching

Which of the two adaptation mechanisms – sensing and responding or phenotypic random switching – will lead to a faster long-term growth of a cell population? In general, sensory systems come at a cost, that is, a decrease of the average growth rate. Pure random switching, on the contrary, yields slow adaptations and nonadapted subpopulations. To compare the two strategies directly, Kussell and Leibler [65] extended the model from Example 10.2 to environments with random alternations between a larger number of states. For each environment state, there is an optimal cell state, and cells can switch between all these states according to one or the other strategy. In responsive switching, cells switch to the currently optimal cell state with only a short time delay. The other strategy is spontaneous stochastic switching. In both cases, cells grow faster if they are in the adapted state. The success of a strategy is measured by the long-term growth rate of an infinitely growing population, but the qualitative results also apply to models with bounded population sizes. Finally, it is assumed that environmental changes are rare, that is, cell populations typically have enough time to reach their stationary composition (i.e., the relative fractions of cell states arising from state switching and growth) before the environment changes again.

Given all these assumptions, long-term growth rates for the two strategies – random switching and signal response – can be directly computed. The growth rates differ mainly by two effective cost terms: the sensing strategy implies a constant cost, that is, a reduction in growth rate, because the sensing system must be maintained; the switching strategy entails a “diversity cost” caused by the fact that parts of the population remain maladapted and grow more slowly. To study the relative advantages of the two strategies, one can compute a break-even cost for signal response: This is the maximal cost value at which signal response is preferred over random switching. With a number of model assumptions, the break-even cost is given by $c^* = (1/\tau)[1 + \ln(\Delta f \tau/2) - \ln(1 + \Delta f/H_m) + I_{\text{env}}]$, where τ is the average duration of environment states, Δf is the

fitness difference between optimally adapted and nonoptimally adapted cell states, and H_m is the switching rate in responsive switching (assumed to be equal for all environments). The entropy $I_{\text{env}} = -\sum_{ij} p_i b_{ij} \ln b_{ij}$ of the environment, computed from the transition probabilities b_{ij} for environment states $j \rightarrow i$ and from the occurrence probabilities p_i of environment states i , states how difficult it is, on average, to predict the next environment state. Thus, responsive switching is preferred when responsive adaptations are fast (large H_m) and the uncertainty about environmental changes is high (large I_{env}).

Like in Example 10.2, optimal rates for stochastic switching can be computed. The optimal switching rate from phenotype j to phenotype i (i.e., the optimal phenotypes for environments j and i) is given by b_{ij}/τ_j , where τ_j is the average duration of environment j . Like in the previous model, optimal switching rates reflect the statistics of environmental changes. If this theory holds for real cells, observed switching rates can tell us about the environment statistics experienced by cells during evolution.

10.3.4

Shannon Information and the Value of Information

Shannon information plays a central role in each of the two adaptation strategies: in signaling pathways, Shannon information describes the information transmitted, and in phenotypic switching, it describes the uncertainty (i.e., lack of information) about future environmental changes. But how can information be turned into fitness? Shannon's notion of information refers to statistical dependencies between variables, independent of what these variables stand for. It does not imply that cells “know” or “represent” their environment – It just tells us *how much they could know* given the stimuli they receive. However, which signals matter to cells is not encoded in Shannon information. This fitness value of signals is quantified by the *value of information*, a measure of information used in Bayesian decision theory (see Section 8.2) [66]. To define it, we consider an agent that can choose between actions and is rewarded for them. An information source is considered valuable if it enables the agent to take better decisions. Specifically, the value of an information source is defined as the maximal average advantage that can be obtained by using this information source when taking decisions.

In the case of cells, the information sources are the intracellular levels, modifications, and localization of signaling molecules that contain Shannon information about the environment. A cell's fitness f depends on the current environment state E (e.g., the presence or absence of antibiotics) and on the cell's actions A (e.g., expressing an antibiotics resistance protein). Adopting

the perspective of Bayesian decision theory, we assume that environment states occur randomly with probabilities $p(E)$, and ask how cells should act *ideally* to maximize their fitness. Without signals, a cell would choose the action that maximizes its expected fitness over the typical distribution of environment states: $\bar{A} = \operatorname{argmax}_A \sum_E p(E)f(A, E)$, and the average cell fitness reads $\langle \bar{f} \rangle = \sum_E p(E)f(\bar{A}, E)$. Now we assume that the cell can sense signals S , which are correlated with the current environment state. Given a signal S , the cell can assume the conditional distribution $p(E|S) = p(E, S)/p(S)$ for the current environment and choose its action accordingly. The optimal signal-based action reads $\hat{A}(S) = \operatorname{argmax}_A \sum_{E,S} p(E|S)f(A, E)$, and the average fitness becomes $\langle \hat{f} \rangle = \sum_{E,S} p(E, S)f(\hat{A}(S), E)$. By using the signal, the average fitness can only become higher, and the difference $\langle \hat{f} \rangle - \langle \bar{f} \rangle$ is called the value of information. It is the maximal price a cell should pay for maintaining the signaling pathway. Note that external stimuli can be valuable signals even if they seem unrelated to a cell's following actions. Here is an example. For bacteria that enter and leave a host, lower oxygen levels are correlated with higher temperature in the cell's typical environment. If this is the case, then not only temperature itself, but also low oxygen levels should trigger adaptations to higher temperatures [67]. Crosstalk of this sort, which makes perfect sense in the light of information value, has been observed in gene regulation networks. We come back to this point below.

If the fitness advantage from signals, as described by the value of information, is what counts for cells, why would Shannon information still matter? Shannon information, in itself, is not about fitness: It describes uncertainties, and the cell's fitness function appears nowhere in the formula. However, in the adaptation strategies discussed above – random switching and signal response, it is mostly uncertainty that limits cells' fitness. In the random switching model, this can be shown explicitly: After optimizing the rates for random switching, the formula for the long-term growth rate contains, as a negative term, the Shannon entropy of the environment, that is, the average uncertainty about the coming environmental state. This directly shows that the fitness of the population is limited by missing information.

10.3.5 Metabolic Shifts and Anticipation

10.3.5.1 Metabolic Shifts

An important type of adaptation is metabolic shifts, the rearrangements of metabolic fluxes that enable cells to use new nutrients. The adaptation process involves metabolic and transcriptional adjustments on different time

scales and has been monitored in *B. subtilis* [68] and *S. cerevisiae* [69] by multi-omics measurements. If we think of the shift as an optimally controlled program (as discussed in Section 11.2), we realize that the enzyme levels should not just be adjusted from moment to moment, but should anticipate the entire process. In experiments, the shifts in individual cells may be obscured by population dynamics. An observed growth curve may suggest a very slow metabolic adaptation within cells. It may actually arise from the fact that a small subpopulation, after quickly shifting to the well-adapted state, is slowly taking over the population. For the adaptation of glucose-grown *E. coli* bacteria to the gluconeogenic substrates acetate, fumarate, malate, and succinate, it has been shown that the observed growth curve does not reflect a slow change in expression, but the growth of a small, immediately adapted subpopulation [70]. This phenomenon brings us back to the topic of phenotypic random switching. Random switching, as a deliberate strategy, may be realized by gene circuits that establish and manage switches that would otherwise not exist. However, switches can also arise as side effects of normal cellular dynamics: Whenever a pathway has a bistable dynamics, cells will fall into one or the other state, and a heterogeneous population emerges. In this case, the task of regulation systems is not to generate two states, but to manage the cells' probabilities to enter existing states and to transit between them.

10.3.5.2 Management of a Transient State

Let us consider an example of a bistable metabolic system. Yeast cells face the problem that glycolysis, instead of running with a normal balanced flux, can jump into a second, unbalanced state in which the flux in upper glycolysis is higher than in lower glycolysis, causing an accumulation of intermediates [71]. This imbalance leads to a growth arrest. A phosphate release, achieved by ATP hydrolysis in the trehalose cycle, can counter this effect. During an adaptation to higher glucose levels, yeast cells run the risk of falling into the unbalanced state; however, a transient activation of trehalose cycling allows most cells to avoid this. Mutants deficient of the trehalose cycle cannot use this safety measure and are much more likely to get into the unbalanced state.

How can we tell whether an individual cell would profit from using the trehalose cycle? Which of the states a cell will fall into depends upon details of the previous cell state, where cells reside in one of the other basins of attraction at the moment of the nutrient change. However, since cells cannot anticipate whether glucose or other nutrients will soon be available, since they cannot be prepared for all possible changes to come, and since they cannot even fully control their internal processes, the risk of getting into unfavorable or maladapted states cannot

be fully avoided. However, actions such as the usage of the trehalose cycle can change the probabilities for such states and lead, on average, to more favorable outcomes. On the single-cell level, an optimal strategy should trade the risk of an unbalanced state against the cost of the trehalose cycle. On the population level, the probabilities for different cell states (visible as subpopulation sizes) could be dictated by bet hedging strategies.

10.3.5.3 Adaptation Based on Indirect Cues

Cells use environmental cues to trigger adaptations. Since they cannot anticipate future changes precisely, they need to resort to expression behavior that is most likely to be favorable, given the current information available. For instance, *E. coli* bacteria, when entering or leaving the body of their host, will experience changes in temperature and adapt to the new environment. According to the idea of homeostasis, bacteria, when sensing a higher temperature, should respond with temperature-related adaptations (e.g., expression of heat shock proteins). However, whether a cue directly reflects an environmental challenge or whether it is only statistically correlated to it does not matter. In fact, in *E. coli* bacteria, temperature and oxygen levels trigger strongly overlapping expression responses; both signals are taken as cues indicating a general change in environments [67]. Such a predictive expression can be an advantageous strategy, as has been shown by artificial evolution experiments: After evolving in an environment in which temperature and oxygen levels varied independently, the expression program changed, suggesting that an independent adaptation to both variables became then more favorable. Analogous crosstalk was observed between sugar utilization pathways in *E. coli* (where first lactose and then maltose are predominant during the bacteria's passage through the intestinal tract) and for heat shock, ethanol and oxidative stress in *S. cerevisiae*, which typically experience these stresses, in this order, during brewing [72]. The fitness advantage of a "Pavlovian conditioning" can be described by a simple mathematical model of predictive expression [73]. In turn, an observed predictive gene expression may give clues about cells' natural environments [67]. Of course, anticipation may matter at any point in time, even if there is no clear sign of a shift to come. This can explain cellular behavior such as preemptive enzyme expression that would appear nonoptimal at first sight [74].

10.3.6 Exploration Strategies

Subpopulations of bacterial populations explore different types of behavior, in the same way as organisms

explore a habitat in space. Also, evolution can be pictured as an exploration of a phenotype space. In all these cases, exploring means searching for good places and, at the same time, collecting information about the landscape. In an unknown landscape, random exploration with a controllable degree of randomness can be a very efficient search strategy. We shall now discuss some examples of exploration strategies in which randomness plays a role.

10.3.6.1 Stress-Induced Mutagenesis

By actively increasing their noise level, cells may increase their chances to evade challenging situations. An example is stress-induced mutagenesis [75]: Bacteria under stress can increase their error rate in DNA replication. By doing so, a cell population can create more inheritable phenotypic variation and is therefore more likely to evade the stress situation instead of going extinct. If we see a cell population as a cloud of points in genotype space, a higher mutation rate makes the cloud spread faster, allowing a faster successful adaptation. This is an example of non-Darwinian evolution, that is, an evolution in which organisms can tune the very process of mutation in response to their current environment.

10.3.6.2 Chemotaxis

Another prominent example of controllable random behavior is *bacterial chemotaxis*. As already mentioned, chemotaxis allows bacteria to move toward sources of attractants (e.g., sugar, aspartate, and serine) and away from sources of repellents (e.g., metal ions and leucine) in their growth medium. Since bacteria are too small to sense the gradients directly, they explore the concentration landscape through random movements consisting of straight "runs," interrupted by "tumbling," that is, random changes of direction [76]. By sensing temporal changes during their movement, bacteria effectively collect information about spatial gradients. During chemotaxis, bacteria monitor the attractant level and adjust the frequency of tumbling events: As long as the concentration increases, the tumbling frequency remains low and the bacterium is likely to stay on track. When a cell moves down a gradient, the tumbling frequency increases, making a change of direction more likely. This biased random walk results, on average, in a movement toward higher attractant concentrations.

Within cells, chemotaxis is enabled and controlled by a signaling pathway that senses the current level of attractants and repellents, "computes" the right tumbling frequency, and transmits the signal to the flagella motors. As a control system, the pathway maps inputs (sensed

time profiles of attractant or repellent concentrations) to outputs (time-dependent probabilities for tumbling events). The response is linear and its impulse response function (see Section 15.5) can be reconstructed from an analysis of bacterial trajectories [77]. Different chemical stimuli evoke different responses. The response to the attractant glucose shows perfect adaptation, that is, the tumbling frequency after a pulse returns exactly to its original value. This means that the integral over the impulse response function is exactly zero. Other responses, for example, to the attractant serine or to the repellent leucine, do not show perfect adaptation.

Why would perfect adaptation matter for successful chemotaxis? As long as the attractant concentration increases in time (i.e., as long as bacteria move up the gradient), cells should lower their tumbling frequencies to stay on track. If the sensed attractant level decreases, they should increase the tumbling frequency. In both cases, the attractant's baseline level, to which the gradient is added, should not matter, so the chemotaxis pathway should be insensitive to it. While this sounds reasonable in general, specific attractant profiles (e.g., strictly linear gradients) might still favor other, less than perfectly adapted chemotaxis strategies. Simulations show that this is indeed true. However, responses with perfect adaptation – and only such responses – can be a maximin strategy, that is, perform best in a worst-case scenario. What does this mean? Given a chemotaxis strategy, we may consider a variety of possible attractant profiles and compute, for each of them, the average chemoattractant uptake. The minimum value over all attractant profiles will be the guaranteed, worst-case uptake for this strategy. Now we compare different strategies by their worst-case uptakes. As proven in Ref. [78], those that maximize this worst-case uptake must show perfect adaptation. In particular, only strategies with perfect adaptation can outcompete motile, nonchemotactic bacteria under all circumstances. Since attractant profiles experienced by bacteria vary widely, perfect adaptation is a safe choice.

10.3.6.3 Infotaxis

Search strategies in complex environments suffer from the *exploration versus exploitation trade-off*, the dilemma whereby individuals, in order to find an optimal point (e.g., the source of an attractant), must decide between exploring their environment or moving straight toward the suspected optimal point. A possible solution to this problem is provided by a hypothetical exploration strategy called infotaxis. Here, individuals actively move in directions in which they can expect to obtain, on average, the maximal amount of relevant information [79]. In

some cases, for instance, if attractants do not diffuse, but are carried by turbulent air movements, the signal-to-noise ratio is very low, and strategies such as chemotaxis would not work unless data are collected for unrealistically long times. Infotaxis remains viable even in such difficult cases.

How does infotaxis work in detail? A hypothetical organism that adopts the infotaxis strategy develops an uncertain mental picture of its environment, for example, a probability distribution describing the position of an attractant source. The uncertainty of the current picture can be quantified by Shannon entropy. Whenever the individual senses a signal, it refines the picture. The distribution will become more narrow and the Shannon entropy of the picture decreases by the amount of Shannon information received. In the infotaxis strategy, every movement would now be chosen such that the Shannon information expected to be obtained in the next moment – that is, the average Shannon information obtained, as expected based on the current environment model – will be maximized. The search for new information resembles active learning methods in machine learning and optimal experimental design in statistics: In all three cases, the task is to ask questions that lead, on average, to maximally informative answers.

Infotaxis is computation-intensive, but it functions efficiently even in fluctuating environments and tolerates errors in the assumed environment model [79]. In simulations, infotaxis in turbulent airflows lead to spiraling movements, resembling the trajectories of insects approaching the source of a scent.

We do not know how other organisms see the world. For unicellular organisms, we may not even ask that question: We would rather ask how cells' movements are implemented by cellular network dynamics, for example, how signaling systems manage to map stimuli to the right responses. Evolution may have “trained” these networks, just like artificial neural networks, to do their mechanistic mapping in favorable ways. In this picture, there is no representation of the environment, just ways to get along with it. But even if we describe living beings in this way, it can be insightful to compare an observed, explainable mechanism, for example, chemotaxis, with infotaxis as a potentially optimal strategy based on representation. By showing how missing information limits the success of search strategies, the comparison highlights the role of information. In other words, successful strategies, no matter how they are implemented, have to involve information collection and must therefore resemble, at least in some ways, infotaxis.

Exercises

Section 10.1

- 1) *Principle of minimal information.* (a) All that is known about a parameter value x is that it lies in the interval $[a, b]$. Show that the principle of minimal information would suggest a uniform probability distribution for x in this interval. (b) For another parameter, the expected value and variance are known. Show that the information principle leads to a Gaussian distribution. (c) Show that applying the information principle to a parameter itself or to its logarithm yields different results.
- 2) *Statistical error of Monte Carlo sampling.* (a) Implement the following Monte Carlo sampling procedure. A circle (diameter L) is surrounded by a square (side length L); to estimate the circle area relative to the square area (true value $\pi/4$), sample N points \mathbf{x}_i uniformly and randomly from the square and count how many of them (number n) are inside the circle (i.e., $|\mathbf{x}_i - \mathbf{x}_{\text{center}}| \leq L/2$). Use n/N as an estimate of $\pi/4$ and plot its empirical value for increasing sample numbers N . (b) Repeat the numerical experiment several times. Which probability distribution of the ratio n/N arises from the random sampling? Compute its average value and standard deviation. (c) Describe how you would use Monte Carlo sampling to compute the probability of oscillations in a given kinetic model. (d) Will the error estimate for Monte Carlo sampling also hold for non-uniform parameter distributions?
- 3) *Variability and uncertainty.* Discuss similarities and differences between biological variability, on the one hand, and subjective uncertainty, on the other. Explain how the two aspects can be captured in kinetic or stoichiometric cell models.
- 4) *Sensitivity analysis and variability analysis.* Describe the difference between sensitivity analysis and variability analysis. Under what circumstances will a local sensitivity analysis yield misleading results?
- 5) *Log-normal distribution.* Show that $Y = \alpha \prod_i X_i^{\beta_i}$ is a log-normal random variable if α and all β_i are constants and the X_i are log-normal random variables.
- 6) *Probabilities for signs or order relations.* (a) Assume that the logarithm of x is a Gaussian-distributed random variable with mean $\langle \ln x \rangle$ and variance $\text{var}(\ln x)$. Consider the probability that x exceeds a given threshold $a > 0$ and express it using the cumulative density $\Phi(\cdot)$ of a standard

Gaussian distribution. (b) Let $\ln x_1$ and $\ln x_2$ be Gaussian-distributed variables with mean value vector $\langle \ln x \rangle$ and covariance matrix $\text{cov}(\ln x)$. What is the probability that the ratio x_1/x_2 exceeds a given threshold $a > 0$?

Section 10.2

- 7) *Effects of feedback.* Describe the potential effects of negative and positive feedback. For each type, give two examples of relevant dynamic effects in cells.
- 8) *Positive feedback in gene expression.* Consider a protein X with concentration x , production rate $a_0 + a x$, and degradation constant b . Write down a kinetic rate equation for x . How does x behave after a_0 is switched from its initial value $a_0 = 0$ to some constant positive value? What is the response time? Which biological functions could such a positive feedback have?
- 9) *Negative feedback in gene expression.* A protein X inhibits its own production and is degraded linearly. The system equation

$$\frac{dx}{dt} = \frac{a}{1+x/k} - bx \quad (10.36)$$

is replaced by an approximation

$$\frac{dx}{dt} = \frac{ak}{x} - bx. \quad (10.37)$$

(a) Compute the steady-state concentration x^{st} for Eq. (10.37). (b) Show that the time profile $x(t) = x^{\text{st}} \sqrt{1 - e^{-2bt}}$ solves Eq. (10.37) with initial condition $x(0) = 0$, and draw its graph schematically. Hint: Insert the time profile into Eq. (10.37). (c) In which phases of the time course will Eq. (10.37) be a good approximation of Eq. (10.36)?

- 10) *Integral feedback.* Consider the model (10.21) and prove that for time-constant u and k , integral feedback stabilizes the output y at the intended steady-state value y_0 .
- 11) *Two-component system.* Derive the steady-state output Eq. (10.23) of the two-component system.
- 12) *Dimensional analysis.* Consider a metabolite concentration x following the rate equation $dx/dt = E_a a - E_b b x$ with enzyme concentrations E_a and E_b and rate constants a and b . Use dimensionality analysis (not an explicit calculation) to show that the steady-state value of x remains unchanged if both enzyme levels change by the same percentage.
- 13) *Rate-limiting steps.* Control in biochemical systems can be widely distributed over or mostly

- concentrated in one rate-limiting step. Speculate about the biological advantages or disadvantages of both cases. Which kinds of enzyme mutations could cause a rate-limiting step in a pathway? How are rate-limiting steps related to reaction thermodynamics?
- 14) *Summation theorems for a signaling cascade.* Consider a signaling pathway that responds to a step-like input stimulus. The transient response $y_i(t)$ of the i th component is characterized by its maximal amplitude y_i^* and by the time point t_i^* at which this maximum is reached. Derive summation theorems for both quantities based on time scaling considerations.
- 15) *Time scaling and summation theorems.* Consider a biochemical system with a supercritical Hopf

bifurcation. Below the bifurcation point, the Jacobian shows a pair of complex eigenvalues $-a \pm ib$ with a small negative real part $-a$ and a positive value of b . (a) How will changes in a and b affect the dynamic behavior? (b) Derive summation theorems for a and b , assuming that the reactions are catalyzed by specific enzymes. (c) How will the dynamics change if all enzyme levels are multiplied by a common factor?

- 16) *Chemotaxis system.* Describe the chemotaxis system in *E. coli*. Why is perfect adaptation necessary for chemotaxis and how is it achieved? Describe the difference between structural robustness and robustness by fine-tuned parameters.

References

- 1 Steuer, R., Kurths, J., Fiehn, O., and Weckwerth, W. (2003) Observing and interpreting correlations in metabolomics networks. *Bioinformatics*, 19 (8), 1019–1026.
- 2 Jaynes, E.T. (1957) Information theory and statistical mechanics. *Phys. Rev.*, 106, 620–630.
- 3 Trigg, G.L. (2005) *Mathematical Tools for Physicists*, Wiley-VCH Verlag GmbH, Weinheim.
- 4 Price, N.D., Schellenberger, J., and Palsson, B.Ø. (2004) Uniform sampling of steady-state flux spaces: means to design experiments and to interpret enzymopathies. *Biophys. J.*, 87, 2172–2186.
- 5 Labhsetwar, P., Cole, J.A., Roberts, E., Price, N.D., and Luthey-Schulten, Z.A. (2013) Heterogeneity in protein expression induces metabolic variability in a modeled *Escherichia coli* population. *Proc. Natl. Acad. Sci. USA*, 111. doi: 10.1073/pnas.1222569110
- 6 Steuer, R., Gross, T., Selbig, J., and Blasius, B. (2006) Structural kinetic modeling of metabolic networks. *Proc. Natl. Acad. Sci. USA*, 103 (32), 11868–11873.
- 7 Grimsb, S., Selbig, J., Bulik, S., Holzhüttter, H.-G., and Steuer, R. (2007) The stability and robustness of metabolic states: identifying stabilizing sites in metabolic networks. *Mol. Syst. Biol.*, 3, 146.
- 8 Liebermeister, W., Uhlendorf, J., and Klipp, E. (2010) Modular rate laws for enzymatic reactions: thermodynamics, elasticities, and implementation. *Bioinformatics*, 26 (12), 1528–1534.
- 9 Wang, L. and Hatzimanikatis, V. (2006) Metabolic engineering under uncertainty: I. Framework development. *Metab. Eng.*, 8, 133–141.
- 10 Wang, L. and Hatzimanikatis, V. (2006) Metabolic engineering under uncertainty. II: analysis of yeast metabolism. *Metab. Eng.*, 8, 142–159.
- 11 Mišković, L. and Hatzimanikatis, V. (2011) Modeling of uncertainties in biochemical reactions. *Biotechnol. Bioeng.*, 108 (2), 413–423.
- 12 Soh, K.C., Miskovic, L., and Hatzimanikatis, V. (2012) From network models to network responses: integration of thermodynamic and kinetic properties of yeast genome-scale metabolic networks. *FEMS Yeast Res.*, 12, 129–143.
- 13 Soh, K.C., Chakrabarti, A., Miskovic, L., and Hatzimanikatis, V. (2013) Towards kinetic modeling of genome-scale metabolic networks without sacrificing stoichiometric, thermodynamic and physiological constraints. *Biotechnol. J.*, 8 (9), 1043–1057.
- 14 Stanford, N.J., Lubitz, T., Smallbone, K., Klipp, E., Mendes, P., and Liebermeister, W. (2013) Systematic construction of kinetic models from genome-scale metabolic networks. *PLoS One*, 8 (11), e79195.
- 15 Liebermeister, W. (2005) Predicting physiological concentrations of metabolites from their molecular structure. *J. Comput. Biol.*, 12 (10), 1307–1315.
- 16 Liebermeister, W. and Klipp, E. (2005) Biochemical networks with uncertain parameters. *IEE Proc. Syst. Biol.*, 152 (3), 97–107.
- 17 Liebermeister, W. and Klipp, E. (2006) Bringing metabolic networks to life: convenience rate law and thermodynamic constraints. *Theor. Biol. Med. Model.*, 3, 41.
- 18 Liebermeister, W. and Klipp, E. (2006) Bringing metabolic networks to life: integration of kinetic, metabolic, and proteomic data. *Theor. Biol. Med. Model.*, 3, 42.
- 19 Ingalls, B.P. and Sauro, H.M. (2003) Sensitivity analysis of stoichiometric networks: an extension of metabolic control analysis to non-steady state trajectories. *J. Theor. Biol.*, 222 (1), 23–36.
- 20 Höfer, T. and Heinrich, R. (1993) A second-order approach to metabolic control analysis. *J. Theor. Biol.*, 164, 85–102.
- 21 Ingalls, B.P. (2004) A frequency domain approach to sensitivity analysis of biochemical systems. *J. Phys. Chem. B*, 108, 1143–1152.
- 22 Liebermeister, W. (2005) Response to temporal parameter fluctuations in biochemical networks. *J. Theor. Biol.*, 234 (3), 423–438.
- 23 Csete, M.E. and Doyle, J.C. (2002) Reverse engineering of biological complexity. *Science*, 295 (5560), 1664–1669.
- 24 Stelling, J., Sauer, U., Szallasi, Z., Doyle, F.J., and Doyle, J. (2004) Robustness of cellular functions. *Cell*, 118, 675–685.
- 25 Kellis, M., Birren, B.W., and Lander, E.S. (2004) Proof and evolutionary analysis of ancient genome duplication in the yeast *Saccharomyces cerevisiae*. *Nature*, 428 (6983), 617–624.
- 26 Gregory, T.R., Andrews, C.B., McGuire, J.A., and Witt, C.C. (2009) The smallest avian genomes are found in hummingbirds. *Proc. Biol. Sci.*, 276 (1674), 3753–3757.

- 27 Turner, P.E. (2005) Cheating viruses and game theory. *Am. Sci.*, 93, 428–435.
- 28 Wagner, A. (2000) Robustness against mutations in genetic networks of yeast. *Nat. Genet.*, 24, 355–361.
- 29 Becskei, A. and Serrano, L. (2000) Engineering stability in gene networks by autoregulation. *Nature*, 405, 590–592.
- 30 Kochanowski, K., Volkmer, B., Gerosa, L., Haverkorn van Rijswijka, B.R., Schmidt, A., and Heinemann, M. (2013) Functioning of a metabolic flux sensor in *Escherichia coli*. *Proc. Natl. Acad. Sci. USA*, 110 (3), 1130–1135.
- 31 Muzzey, D., Gómez-Uribe, C.A., Mettetal, J.T., and van Oudenaarden, A. (2009) A systems-level analysis of perfect adaptation in yeast osmoregulation. *Cell*, 138, 160–171.
- 32 Yi, T., Huang, Y., Simon, M.I., and Doyle, J. (2000) Robust perfect adaptation in bacterial chemotaxis through integral feedback control. *Proc. Natl. Acad. Sci. USA*, 97 (9), 4649–4653.
- 33 He, F., Fromion, V., and Westerhoff, H.V. (2013) (Im)Perfect robustness and adaptation of metabolic networks subject to metabolic and gene-expression regulation: marrying control engineering with metabolic control analysis. *BMC Syst. Biol.*, 7, 131.
- 34 Shinar, G., Milo, R., Martinez, M.R., and Alon, U. (2007) Input-output robustness in simple bacterial signaling systems. *Proc. Natl. Acad. Sci. USA*, 104 (50), 19931–19935.
- 35 Barkai, N. and Leibler, S. (1997) Robustness in simple biochemical networks. *Nature*, 387, 913–917.
- 36 Alon, U., Surette, M.G., Barkai, N., and Leibler, S. (1999) Robustness in bacterial chemotaxis. *Nature*, 397, 168–171.
- 37 Kollmann, M., Løvdok, L., Bartholome, K., Timmer, J., and Sourjik, V. (2005) Design principles of a bacterial chemotaxis network. *Nature*, 438, 504–507.
- 38 Batchelor, E., Silhavy, T.J., and Goulian, M. (2004) Continuous control in bacterial regulatory circuits. *J. Bacteriol.*, 186 (22), 7618–7625.
- 39 West, G.B., Brown, J.H., and Enquist, B.J. (1997) A general model for the origin of allometric scaling laws in biology. *Science*, 276, 122–126.
- 40 West, G.B. and Brown, J.H. (2005) The origin of allometric scaling laws in biology from genomes to ecosystems: towards a quantitative unifying theory of biological structure and organization. *J. Exp. Biol.*, 208, 1575–1592.
- 41 Bremer, H. and Dennis, P.D. (1996) Modulation of chemical composition and other parameters of the cell by growth rate, in *Escherichia coli and Salmonella typhimurium: Cellular and Molecular Biology* (eds. F.C. Neidhardt *et al.*), American Society for Microbiology Press, pp. 1553–1569.
- 42 Valgepea, K., Adamberg, K., Seiman, A., and Vilu, R. (2013) *Escherichia coli* achieves faster growth by increasing catalytic and translation rates of proteins. *Mol. Biosyst.*, 9 (9), 2344–2358.
- 43 Scott, M., Gunderson, C.W., Mateescu, E.M., Zhang, Z., and Hwa, T. (2010) Interdependence of cell growth and gene expression: origins and consequences. *Science*, 330, 1099.
- 44 Weiße, A.Y., Oyarzún, D.A., Danos, V., and Swain, P.S. (2015) Mechanistic links between cellular trade-offs, gene expression, and growth. *Proc. Natl. Acad. Sci. USA*, 112 (9), E1038–E1047.
- 45 Hornberg, J.J., Binder, B., Bruggeman, F.J., Schoeberl, B., Heinrich, R., and Westerhoff, H.V. (2005) Control of MAPK signalling: from complexity to what really matters. *Oncogene*, 24, 5533–5542.
- 46 Reznik, E., Chaudhary, O., and Segrè, D. (2013) The average enzyme principle. *FEBS Lett.*, 587 (17), 2891–2894.
- 47 Shinar, G. and Feinberg, M. (2010) Structural sources of robustness in biochemical reaction networks. *Science*, 327 (5971), 1389–1391.
- 48 Steuer, R., Waldherr, S., Sourjik, V., and Kollmann, M. (2011) Robust signal processing in living cells. *PLoS Comput. Biol.*, 7 (11), e1002218.
- 49 Cruz, A.L.B., Hebly, M., Duong, G.-H., Wahl, S.A., Pronk, J.T., Heijnen, J.J., Daran-Lapujade, P., and van Gulik, W.M. (2012) Similar temperature dependencies of glycolytic enzymes: an evolutionary adaptation to temperature dynamics? *BMC Syst. Biol.*, 6, 151.
- 50 Ruoff, P., Zakhartsev, M., and Westerhoff, H.W. (2007) Temperature compensation through systems biology. *FEBS J.*, 274, 940–950.
- 51 Tseng, Y.-Y., Hunt, S.M., Heintzen, C., Crosthwaite, S.K., and Schwartz, J.-M. (2012) Comprehensive modelling of the *Neurospora* circadian clock and its temperature compensation. *PLoS Comput. Biol.*, 8 (3), e1002437.
- 52 Quinton-Tulloch, M.J., Bruggeman, F.J., Snoep, J.L., and Westerhoff, H.V. (2013) Trade-off of dynamic fragility but not of robustness in metabolic pathways *in silico*. *FEBS J.*, 280, 160–173.
- 53 Kirschner, M. and Gerhart, J. (1998) Evolvability. *Proc. Natl. Acad. Sci. USA*, 95 (15), 8420–8427.
- 54 Lee, E., Salic, A., Krüger, R., Heinrich, R., and Kirschner, M.W. (2003) The roles of APC and Axin derived from experimental and theoretical analysis of the Wnt pathway. *PLoS Biol.*, 1 (1), 116–132.
- 55 Eldar, A., Rosin, D., Shilo, B.Z., and Barkai, N. (2003) Self-enhanced ligand degradation underlies robustness of morphogen gradients. *Dev. Cell*, 5, 635–646.
- 56 Goelzer, A., Brikci, F.B., Martin-Verstraete, I., Noirot, P., Bessières, P., Aymerich, S., and Fromion, V. (2008) Reconstruction and analysis of the genetic and metabolic regulatory networks of the central metabolism of *Bacillus subtilis*. *BMC Syst. Biol.*, 2, 20.
- 57 Shannon, C.E. (1948) A mathematical theory of communication. *AT&T TECH. J.*, 27, 379–423.
- 58 Mehta, P., Goyal, S., Long, T., Bassler, B.L., and Wingreen, N.S. (2009) Information processing and signal integration in bacterial quorum sensing. *Mol. Syst. Biol.*, 5, 325.
- 59 Cheong, R., Rhee, A., Wang, C.J., Nemenman, I., and Levchenko, A. (2011) Information transduction capacity of noisy biochemical signaling networks. *Science*, 334, 354.
- 60 Shoval, O., Goentoro, L., Hart, Y., Mayo, A., Sontag, E., and Alon, U. (2010) Fold-change detection and scalar symmetry of sensory input fields. *Proc. Natl. Acad. Sci. USA*, 107 (36), 15995–16000.
- 61 Balaban, N.Q., Merrin, J., Chait, R., Kowalik, L., and Leibler, S. (2004) Bacterial persistence as a phenotypic switch. *Science*, 305, 1622–1625.
- 62 Arkin, A., Ross, J., and McAdams, H.H. (1998) Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected *Escherichia coli* cells. *Genetics*, 149 (4), 1633–1648.
- 63 Bigger, W.B. (1944) Treatment of staphylococcal infections with penicillin. *Lancet*, 144 (6320), 497–500.
- 64 Kussell, E., Kishony, R., Balaban, N.Q., and Leibler, S. (2005) Bacterial persistence: a model of survival in changing environments. *Genetics*, 169 (4), 1807–1814.
- 65 Kussell, E. and Leibler, S. (2005) Phenotypic diversity, population growth, and information in fluctuating environments. *Science*, 309 (5743), 2075–2078.
- 66 Duda, R.O., Hart, P.E., and Stork, D.G. (2000) *Pattern Classification*, 2nd edn, John Wiley & Sons, Inc., New York.
- 67 Tagkopoulos, I., Liu, Y.C., and Tavazoie, S. (2008) Predictive behavior within microbial genetic networks. *Science*, 320 (5881), 1313–1317.
- 68 Buescher, J.M., Liebermeister, W., Jules, M., Uhr, M., Muntel, J., Botella, E., Hessling, B., Kleijn, R.J., Chat, L.L., Lecointe, F., Mäder,

- U., Nicolas, P., Piersma, S., Rügheimer, F., Becher, D., Bessières, P., Bidnenko, E., Denham, E.L., Dervyn, E., Devine, K.M., Doherty, G., Drulhe, S., Felicori, L., Fogg, M.J., Goelzer, A., Hansen, A., Harwood, C.R., Hecker, M., Hubner, S., Hultschig, C., Jarmer, H., Klipp, E., Leduc, A., Lewis, P., Molina, F., Noirot, P., Peres, S., Pigeonneau, N., Pohl, S., Rasmussen, S., Rinn, B., Schaffer, M., Schnidder, J., Schwikowski, B., van Dijl, J.M., Veiga, P., Walsh, S., Wilkinson, A.J., Stelling, J., Aymerich, S., and Sauer, U. (2012) Global network reorganization during dynamic adaptations of *Bacillus subtilis* metabolism. *Science*, 335 (6072), 1099–1103.
- 69 Zampar, G.G., Kümmel, A., Ewald, J., Jol, S., Bastian, B., Picotti, P., Aebersold, R., Sauer, U., Zamboni, N., and Heinemann, M. (2013) Temporal system-level organization of the switch from glycolytic to gluconeogenic operation in yeast. *Mol. Syst. Biol.*, 9, 651.
- 70 Kotte, O., Volkmer, B., Radzikowski, J.L., and Heinemann, M. (2014) Phenotypic bistability in *Escherichia coli*'s central carbon metabolism. *Mol. Syst. Biol.*, 10, 736.
- 71 van Heerden, J.H., Wortel, M.T., Bruggeman, F.J., Heijnen, J.J., Bollen, Y.J.M., Planqué, R., Hulshof, J., O'Toole, T.G., Wahl, S.A., and Teusink, B. (2014) Lost in transition: startup of glycolysis yields subpopulations of nongrowing cells. *Science*, 343 (6174), 1245114.
- 72 Mitchell, A., Romano, G.H., Groisman, B., Yona, A., Dekel, E., Kupiec, M., Dahan, O., and Pilpel, Y. (2009) Adaptive prediction of environmental changes by microorganisms. *Nature*, 460 (7252), 220–204.
- 73 Mitchell, A. and Pilpel, Y. (2011) A mathematical model for adaptive prediction of environmental changes by microorganisms. *Proc. Natl. Acad. Sci. USA*, 108 (17), 7271–7276.
- 74 Wessely, F., Bart, M., Guthke, R., Li, P., Schuster, S., and Kaleta, C. (2011) Optimal regulatory strategies for metabolic pathways in *Escherichia coli* depending on protein costs. *Mol. Syst. Biol.*, 7, 515.
- 75 Bjedov, I., Tenaillon, O., Gérard, B., Souza, V., Denamur, E., Radman, M., Taddei, F., and Matic, I. (2003) Stress-induced mutagenesis in bacteria. *Science*, 300, 1404–1409.
- 76 Berg, H.C. and Brown, D.A. (1972) Chemotaxis in *Escherichia coli* analysed by three-dimensional tracking. *Nature*, 239, 500–504.
- 77 Massona, J.-B., Voisinne, G., Wong-Nga, J., Celania, A., and Vergassola, M. (2012) Noninvasive inference of the molecular chemotactic response using bacterial trajectories. *Proc. Natl. Acad. Sci. USA*, 109 (5), 1802–1807.
- 78 Celani, A. and Vergassola, M. (2010) Bacterial strategies for chemotaxis response. *Proc. Natl. Acad. Sci. USA*, 107 (4), 1391–1396.
- 79 Vergassola, M., Villermaux, E., and Shraiman, B.I. (2007) 'Infotaxis' as a strategy for searching without gradient. *Nature*, 445, 406–409.

Further Reading

- Structural robustness:** Barkai, N. and Leibler, S. (1997) Robustness in simple biochemical networks. *Nature*, 387, 913–917.
- Principle of minimal information:** Jaynes, E.T. (1957) Information theory and statistical mechanics. *Phys. Rev.*, 106, 620–630.
- Uncertain parameters in systems biology models:** Liebermeister, W. and Klipp, E. (2005) Biochemical networks with uncertain parameters. *IEE Proc. Syst. Biol.*, 152 (3), 97–107.
- Flux sampling:** Price, N.D., Schellenberger, J., and Palsson, B.Å. (2004) Uniform sampling of steady-state flux spaces: means to design experiments and to interpret enzymopathies. *Biophys. J.*, 87, 2172–2186.
- Value of information and evolution:** Rivoire, O. and Leibler, S. (2011) The value of information for populations in varying environments. *J. Stat. Phys.*, 142 (6), 1124–1166.
- Cell growth laws:** Scott, M., Gunderson, C.W., Mateescu, E.M., Zhang, Z., and Hwa, T. (2010) Interdependence of cell growth and gene expression: origins and consequences. *Science*, 330, 1099.
- Shannon information:** Shannon, C.E. (1948) A mathematical theory of communication. *AT&T Tech. J.*, 27, 379–423.
- Structural robustness:** Shinar, G., Milo, R., Martinez, M.R., and Alon, U. (2007) Input–output robustness in simple bacterial signaling systems. *Proc. Natl. Acad. Sci. USA*, 104 (50), 19931–19935.
- Robustness:** Stelling, J., Sauer, U., Szallasi, Z., Doyle, F.J., and Doyle, J. (2004) Robustness of cellular functions. *Cell*, 118, 675–685.
- Structural kinetic modeling:** Steuer, R., Gross, T., Selbig, J., and Blasius, B. (2006) Structural kinetic modeling of metabolic networks. *Proc. Natl. Acad. Sci. USA*, 103 (32), 11868–11873.
- Uncertain parameters in systems biology models:** Wang, L., Hatzimanikatis, V., Miskovic, L., and Hatzimanikatis, V. (2011) Modeling of uncertainties in biochemical reactions. *Biotechnol. Bioeng.*, 108 (2), 413–423.
- Scaling laws:** West, G.B. and Brown, J.H. (2005) The origin of allometric scaling laws in biology from genomes to ecosystems: towards a quantitative unifying theory of biological structure and organization. *J. Exp. Biol.*, 208, 1575–1592.

Optimality and Evolution

11

Evolution is an open process without any goals. However, there is a general trend toward increasing complexity, and phenotypes arising from mutation and selection often look as if they were optimized for specific biological functions. The evolution toward optimal solutions can be studied experimentally: in a competition among bacteria, faster growing mutants are likely to outcompete the wild-type population until bacterial genomes, after many such changes, become enriched with features needed for fast growth. Dekel and Alon [1] grew *Escherichia coli* bacteria under different combinations of lactose levels and under artificially induced expression levels of the Lac operon (which includes the gene lacZ). From the measured growth rates, they predicted that there exists, for each lactose level, an optimal LacZ expression level that maximizes growth and, therefore, the fitness advantage in a direct competition. The prediction was tested by having bacteria evolve in serial dilution experiments at given lactose levels. After a few hundred generations, the wild-type population was in fact replaced by mutants with the predicted optimal LacZ expression levels.

To picture the mechanism of evolution, we can imagine individuals as points in a phenotype space (see Figure 11.1), and a population forming a cloud (or distribution) of points. Individuals' chances to replicate (or their expected number of descendants) depend on the phenotype and can be described by a fitness function. Due to mutations and recombination of genes, offspring may show new phenotypes and may spread in this fitness landscape. Due to a selection for high fitness, a population of individuals will change from generation to generation, tending to move toward a fitness maximum. The fitness landscape can have local optima, each surrounded by a basin of attraction. If rare mutations are needed to jump into another basin of attraction, the population may be confined to a current local optimum even if better optima exist elsewhere. The fitness function can also be

11.1 Optimality in Systems Biology Models

- Mathematical Concepts for Optimality and Compromise
- Metabolism Is Shaped by Optimality
- Optimality Approaches in Metabolic Modeling
- Metabolic Strategies
- Metabolic Adaptation

11.2 Optimal Enzyme Concentrations

- Optimization of Catalytic Properties of Single Enzymes
- Optimal Distribution of Enzyme Concentrations in a Metabolic Pathway
- Temporal Transcription Programs

11.3 Evolution and Self-Organization

- Introduction
- Selection Equations for Biological Macromolecules
- The Quasispecies Model: Self-Replication with Mutations
- The Hypercycle
- Other Mathematical Models of Evolution: Spin Glass Model
- The Neutral Theory of Molecular Evolution

11.4 Evolutionary Game Theory

- Social Interactions
- Game Theory
- Evolutionary Game Theory
- Replicator Equation for Population Dynamics
- Evolutionarily Stable Strategies
- Dynamical Behavior in the Rock–Scissors–Paper Game
- Evolution of Cooperative Behavior
- Compromises between Metabolic Yield and Efficiency

Exercises

References

Further Reading

drawn directly as a landscape in *genotype* space. Since the same phenotype, on which selection primarily acts, can emerge from various genotypes, movements in genotype space will often be fitness-neutral. Moreover, analogous traits can evolve, under similar selective pressures, in organisms that are genetically very distant.

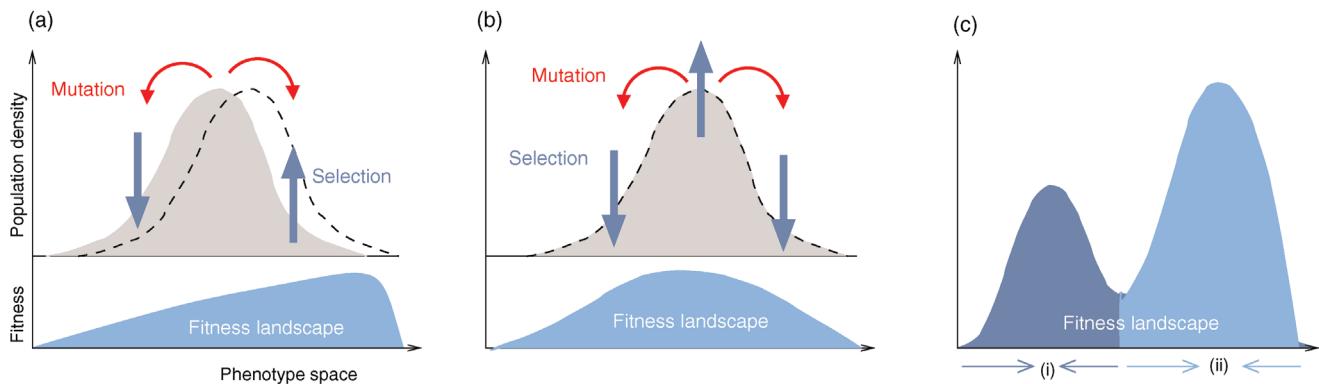


Figure 11.1 Evolution in a fitness landscape. (a) Evolution by mutation and selection. A population, shown as a frequency density in phenotype space (gray), changes by mutation and selection. A higher fitness value (shown at the bottom) represents a selection advantage and leads to an increase in frequency. As the different phenotypes change in their frequencies, the distribution gradually moves along the fitness gradient (dashed line). (b) Stationary population around a local fitness optimum. Mutation and selection are balanced and the population is stable. (c) Fitness landscape with two optima and their basins of attraction (marked by (i) and (ii)).

Evolution as a Search Strategy

Evolution by mutation and selection is an efficient search strategy for optima in complex fitness landscapes and can lead to relatively predictable outcomes. This may sound surprising because mutations are random events. Any innovation emerges from individuals in which certain mutations first occurred and which had to survive and pass it on. However, the number of possible favorable mutations is usually large, and so is the number of individuals in which these mutations can occur: thus, even if a certain change is very unlikely to occur in a single individual, it may be very likely to occur in a species. This is how evolution, the “tinkerer,” can act as an engineer [2].

Biomechanical problems solved by evolution are numerous and complex, and the solutions are often so good that engineers try to learn from them. Evolution has even inspired the development of very successful algorithms for numerical optimization [3]. Genetic algorithms (see Section 6.1) can tackle general, not necessarily biological optimization problems: metaphorically, candidate solutions to a problem are described as “genotypes” (usually encoded by numbers). A population of such potential solutions is then optimized in a series of mutation and selection steps. Since genetic algorithms simulate populations of individuals and not single individuals (as in, for example, simulated annealing; see Section 6.1), they allow for recombination of advantageous “partial solutions” from different individuals. This can considerably speed up the convergence toward optimal solutions.

What makes mutation and selection such a powerful search strategy? One reason is that the search for alternatives happens in a massively parallel way. A bacterial population produces huge numbers of mutants in every generation, and there is a constant selection for

advantageous mutations that can be integrated into the complex physiology and genetics of existing cells. In constant environments, well-adapted genotypes are continuously tested and advantageous traits are actively conserved. At the same time, the population remains genetically flexible: when conditions are changing, it can move along a fitness gradient toward the next local optimum (see Figure 11.1a).

During evolution, there were long periods of relatively stable conditions, in which species could adapt to specific ecological niches. These periods, however, were interrupted by sudden environmental changes in which specialists went extinct and new species arose from the remaining generalists. Thus, to succeed in evolution, species should be not only well adapted, but also *evolvable*, that is, able to further adapt genetically [4]. As we saw in Sections 8.3 and 10.2, evolvability can be improved by the existence of robust physiological modules. Simulations suggest that varying external conditions can speed up evolution toward modular, well-adapted phenotypes [5]. Evolvability is further facilitated by exaptation – the reuse of existing traits to perform new functions. An example of exaptation are crystallin proteins, which were originally enzymes, but adopted a second unrelated function, namely, to increase the refraction index of the eye lens. Also in metabolism, potentially useful traits can emerge as by-products: for instance, the metabolic networks that enable cells to grow on one carbon source will, typically, also allow them to live on many other carbon sources. Even if a carbon source has never been present during evolution, the evolutionary step to using it as a nutrient can be relatively small [6].

Finally, we should remember that genotypes and phenotypes are not linked by abstract rules, but through very concrete physiological processes – the processes we

attempt to understand in systems biology. Accordingly, what appears to be a selection for phenotypic traits may actually be a selection for *regulation mechanisms* that shape and control these traits and adapt them to an individual's needs and environment. An example is given by the way in which bones grow to their functional shapes, adapt to typical mechanical stresses, and re-enforce their structures after a fracture. This adaptation is based on feedback mechanisms. Higher stresses will cause bone growth and a strengthening of internal structures, while stresses below average cause a remodeling, that is, a removal of material [7]. The result of this regulation, described by the mechanostat model [8], are well-adapted bone shapes and microstructures. The physiological processes underlying this adaptation have been studied in detail [9] and the resulting microstructures of bones can be simulated [10]. Similar principles hold for other structures that emerge from self-organized processes, for example, cell organelles [11]: an evolutionary selection for good shapes may, effectively, consist in a selection for regulation mechanisms that produce the right shapes under the right circumstances.

Control of Evolution Processes

Evolution can proceed quite predictably. We saw an example in the artificial evolution of Lac operon expression levels, where the expression level after evolution could be predicted from previous experiments. Another known example is the evolution from C₃ to C₄ carbon fixation systems in plants, which occurred independently more than 60 times in evolution. The evolution toward C₄ metabolism involves a number of changes in metabolism. In theory, these changes could take place in various orders. However, simulations of the evolutionary process, based on fitness evaluation in metabolic models, showed that the evolutionary trajectory (i.e., the best sequence of adaptation steps) is almost uniquely determined [12]. Even if individual gene mutations cannot be foreseen, the order of phenotypic changes seems to be quite predictable.

Can evolutionary dynamics, for example, the evolution of microbes in experiments, be steered by controlling the environment or by applying genetic modifications that constrain further evolution? Avoiding unintended evolution is important in biotechnology: if genetic changes, meant to increase the production of chemicals, burden cells excessively, the cells' growth rate will be severely reduced and the engineered cells may be outcompeted by faster growing mutants. This can be avoided, for instance, by applying genetic changes that stoichiometrically couple biomass production to production of the desired product.

Another case in which controlling evolution would matter is the prevention of bacterial resistances. Resistant

bacteria carry mutations that make them less sensitive to antibiotic drugs. The resistant mutants have better chances to survive, so their resistance genes will be selected for. Chait *et al.* [13] have developed a method to prevent this. A combined administration of antibiotics can either increase or reduce antibiotics' effects on cell proliferation (*synergism* or *antagonism*). In extreme cases of antagonism, drug combinations can even have a weaker effect than either of the drugs alone. This phenomenon, known as *Suppressive* drug combination, has a paradoxical consequence: mutants that have become resistant to one drug will suffer more strongly from the other one (because the first drug cannot exert its suppressive effect) and will be selected against. Thus, a suppressive combination of antibiotics traps bacteria in a local fitness optimum where resistances cannot spread. In experiments, this was shown to prevent the emergence of resistances [13,14].

There is, however, another lesson to be learned from this: if both drugs are applied, their effect decreases. For somebody who does not care about resistances spreading, taking only one of the drugs would be more effective. The fight against bacterial resistance – from which we all benefit – can thus be undermined by everyone's desire to get the most effective drug treatment *now*. This means that we're also trapped in a dilemma called "tragedy of the commons," which we encounter again in Section 11.4.

11.1

Optimality in Systems Biology Models

Summary

Evolution shapes the phenotypes of cells, including pathway structures, typical enzyme levels, and even biochemical rate constants. Phenotypes resulting from evolutionary selection are expected to show a maximal fitness in typical environments. This hypothesis can be tested by optimality studies, in which variants of a model are rated by a presumable objective and the best model variant is chosen. The objective function is meant to represent relevant selective pressures in an evolution scenario in question. In optimality studies, we do not simply describe *how* systems work; we ask *why* they work the way they do, and which fitness requirements and constraints may have shaped them. Optimality considerations can explain the structure and regulation of metabolic pathways and may help engineer microorganisms in biotechnology.

Optimality principles play a central role in many fields of research and development, including engineering, economics, and physics. Whenever we build a machine, steer a technical process, or produce goods, we want to do this

effectively (i.e., realize some objective, which we define) and efficiently (i.e., do so with the least possible effort). Often, we need to deal with trade-offs, for instance, if machines that are built to be more durable and energy-efficient become more costly. Moreover, what is optimal for one person may be detrimental for society. In such cases, the solution to a problem depends crucially on what the optimality problem is considered exactly.

In physics, optimality principles are central, but they are used in a more abstract sense and without referring to human intentions. Many physical laws can be formulated in a way *as if* nature maximized or minimized certain functions. In classical mechanics, laws as basic as Newton's laws of motion can be derived from variational principles [15]: in Lagrangian mechanics, instead of integrating the system equations in time, we consider the system's entire trajectory and compare it with other conceivable (but unphysical) trajectories. Each trajectory is scored by a so-called *action functional*, and the real, physical trajectory is a trajectory for which this functional shows an extremal value. Variational principles exist for many physical laws, and they sometimes connect different theories of the same phenomenon. For instance, Fermat's principle states that light rays from point A to point B follow paths of extremal duration: small deviations from the path would always increase (or, in some cases, always decrease) the time to reach point B. Fermat's principle entails not only straight light rays, but also the law of diffraction. An explanation comes from wave optics: light waves following extremal paths and the paths surrounding them show constructive interference, allowing for light to be detected at the end point; in quantum electrodynamics, the same principle holds for the wave function of photons [16]. Another variational principle, defining thermodynamically feasible flux distributions [17], is described in Section 15.6.

The idea of optimality is also common in biology and bioengineering. On the one hand, we can assume that phenotypic traits of organisms, for example, the body shapes of animals, are optimized during evolution. On the other hand, engineered microorganisms are supposed to maximize, for instance, the yield of some chemical product. In both cases, optimality assumptions can be translated into a mathematical model. Optimality-based (or "teleological") models follow a common scheme: we consider different possible variants x of a biological system, score them by a fitness function $f(x)$, and select variants of maximal fitness. In studies of cellular networks, model variants considered can differ in their structure, kinetic parameters, or regulation and may be constrained by physical limitations. The fitness function can represent either the expected evolutionary success or our goals in metabolic engineering. In models, the fitness

is expressed as a function of the system's behavior and, possibly, of costs caused by the system.

Teleological models not only describe *how* a system works, but also explore *why* it works in this way and why it shows its specific dynamics. Thus, these models incorporate mechanisms, but their main focus is on how a system may have been optimized, according to what principles, and under what constraints. Optimality studies can also tell us about potential limits of a system. We may ask: if a known fitness function had to be maximized in evolution or biotechnology, what fitness value could be maximally achieved, and how? Turning this around, if a system is close to its theoretical optimum, we may suspect that natural selection, with fitness criteria similar to the ones assumed, has acted upon the system. Finally, optimality studies force us to think about the typical environment to which systems are adapted. A general assumption is that species are optimized for routine situations, but also rare severe events can exert considerable selective pressures. Generally, variable environments entail different types of selective pressure (e.g., toward robustness or versatility) than constant environments (adaptation to minimal effort and specialization).

Teleological Modeling Approaches

How does the concept of optimality fit into the mechanistic framework of molecular biology? As noted in Section 8.1.5, observations can be explained not only by cause and effect (Aristotle's *causa efficiens*), but also by objectives (Aristotle's *causa finalis*). In our languages, we do this quite naturally, for instance, when saying "*E. coli* bacteria produce flagella *in order to swim*." The sentence not only describes a physical process, but also relates it to a task. This does not mean that we ascribe intentions to cells or to evolution – it simply expresses the fact that flagella can increase cells' chances to proliferate, and this is a reason why *E. coli* bacteria – which have outcompeted many mutants in the past – can express flagella.

Teleological statements are statements about biological function. But then, what exactly is meant by "function," and why do we need such a concept? Can't we simply describe what is happening inside cells? In fact, cell biology combines two very different levels of description: on the one hand, the physical dynamics within cells and, on the other hand, the evolutionary mechanisms that lead to these cells and can be seen at work in evolution experiments. The two perspectives inform each other: the cellular dynamics we observe today is the one that has succeeded in evolution, and changes in this dynamics will affect the species' further evolution. The entanglement of long-term selection and short-term cell physiology is subsumed in the concept of biological function.

Thus, “cells express flagella *in order to swim*” can be read as a short form of “cells that cease to produce flagella may be less vital due to shortage of nutrients” or, one step further, “without producing flagella in the right moment, cells would have been outcompeted by other cells (maybe, by cells producing flagella).” Optimality-based models translate such hypotheses into mathematical formulas. Here, “*in order to*” is represented by scoring a system by an objective function, and optimizing it. Dynamic models relate cause and mechanistic effect; optimality models, in contrast, relate mechanisms and fitness effects.

Epistemologically, the concept of “function” has a similar justification as the concept of randomness. Random noise can be used as a proxy for processes that we omit from our models, either because they are too difficult to describe or because their details are unknown. Since we cannot neglect them completely, we account for them in a simple way – by random noise. Similarly, a functional objective can be a proxy for evolutionary selection, which by itself has no place in biochemical models. Since we cannot neglect evolution completely, we may account for it in a simple way – by a hypothetical optimization with some presumable objective. The biological justification lies in the assumption that evolution would already have selected out system variants that are clearly suboptimal. How much nonoptimality can be tolerated depends on many factors, including the strictness of selection (reflected by the steepness of the fitness function in models), population size, and the evolutionary time span considered.

Optimality and Model Fitting

Another point in systems biology modeling where optimization matters is parameter estimation. As mathematical problems, estimation and optimization of biochemical

parameters are very similar: to find the most likely model, parameters are optimized for goodness of fit, and to find the most performant one, they are optimized for some fitness function. Biochemical constraints play a central role in both cases. Algorithms for numerical optimization are described in Section 6.1. They range from an optimization of structural model variants to linear flux optimization (as in flux balance analysis (FBA)) or a nonlinear optimization of enzyme levels or rate constants.

11.1.1

Mathematical Concepts for Optimality and Compromise

In an optimality perspective, all biological traits result from compromise. In the case of the Lac operon, an increased expression level increases the cell’s energy supply (if lactose is present), but also demands resources (e.g., energy and ribosomes) and causes a major physiological burden due to side effects of the Lac permease activity [18]. To capture such trade-offs, optimality models typically combine four components: a system to be optimized, a fitness function (which may comprise costs and benefits), ways in which the system can be varied, and constraints under which these variations take place. We will now see how such trade-offs can be treated mathematically (Figure 11.2 gives an overview).

Cost–Benefit Models

In cost–benefit calculations for metabolic systems [1,19–21], a difference

$$f(x) = g(x) - h(x) \quad (11.1)$$

between a benefit g and a cost h is assumed as a fitness function (see Figure 11.2a). The variable x is a vector

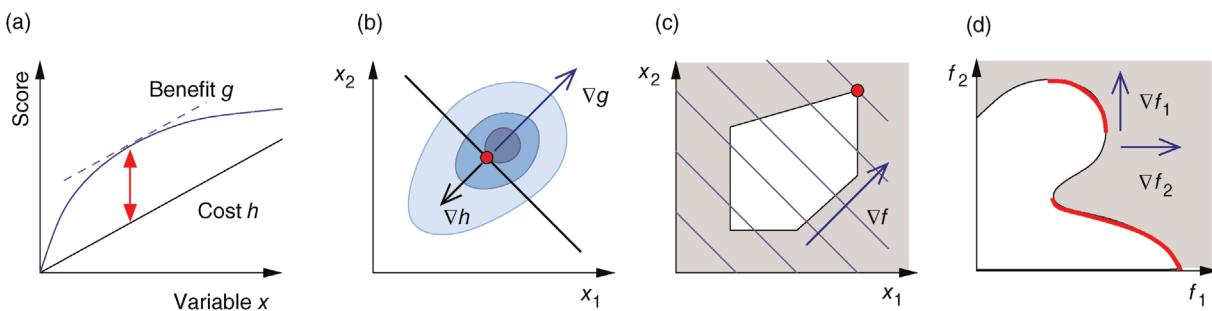


Figure 11.2 Optimization under constraints. (a) Cost–benefit optimization. A fitness function, defined as the difference $f(x) = g(x) - h(x)$ between benefit and cost, becomes maximal where both curves have the same slope (dashed). (b) Local optimum under an equality constraint (black line). A fitness function $g(x)$ in two dimensions is shown by contour lines. The constraint is defined by an equality $h(x) = c$. In the constrained optimum point (red circle), contour lines and constraint line are parallel, so the gradients must satisfy an equation $\nabla g + \lambda \nabla h = 0$ with a Lagrange multiplier λ . (c) Linear programming: a feasible region (white) is defined by linear inequality constraints for the arguments x_1 and x_2 . The linear fitness function is shown by contour lines and gradient (arrow). (d) Pareto (multi-objective) optimization with two objectives f_1 and f_2 (shown on the axes). Feasible combinations (f_1, f_2) lie within the white region. States in which none of the objectives can be increased without decreasing the other objective are Pareto-optimal (red). Pareto optimization yields a continuous set of solutions, the so-called Pareto front.

describing quantitative traits to be varied, for example, enzyme abundances in a kinetic model. In the model for Lac expression [1], bacterial growth is treated as a function of two variables: external lactose concentration (an external parameter) and expression of LacZ (regarded as a control parameter, optimized by evolution). The function was first fitted to measured growth rates and then used to predict the optimal LacZ expression levels for different lactose concentrations. A local optimum of the variable x – in this case, the enzyme level – implies a vanishing fitness slope $df/dx = dg/dx - dh/dx$. The *marginal benefit* dg/dx and the *marginal cost* dh/dx must therefore be equal. In an optimal state, the benefit obtained from expressing an additional enzyme molecule and the cost for producing this molecule must exactly be balanced. Figure 11.2a illustrates this: since the benefit saturates at high x values while the cost keeps increasing, the fitness function shows a local maximum. If the cost function were much steeper, the marginal cost would exceed the marginal benefit already at the level $x = 0$. Since enzyme levels cannot be negative, this point would be a boundary optimum, and in this case the enzyme should not be expressed.

Inequality Constraints

Which constraints need to be considered in optimization? This depends very much on our models and questions. Typically, there are nonnegativity constraints (e.g., for enzyme levels) and physical restrictions (e.g., upper bounds on substance concentrations, bounds on rate constants due to diffusion limitation, or Haldane relationships between rate constants to be optimized). However, inequality constraints can also represent biological costs. In fact, various important constraints on cell function, including limits on cell sizes and growth rates, the dense packing of proteins in cells, and cell death at higher temperatures, can be derived from basic information such as protein sizes, stabilities, and rates of folding and diffusion, as well as the known protein length distribution [22].

Mathematically, costs and constraints are, to an extent, interchangeable, which provides some freedom in how to formulate a model. As an example, consider a cost–benefit problem as in Eq. (11.1), but with several variables x_i . In a unique global optimum $\mathbf{x}^{\text{opt}} = \text{argmax}_{\mathbf{x}} g(\mathbf{x}) - h(\mathbf{x})$, the cost function has a certain value $h(\mathbf{x}^{\text{opt}}) = h^{\text{opt}}$. We can now reformulate the model: we drop the cost terms and require that $g(\mathbf{x})$ be maximized under the constraint $h(\mathbf{x}) \leq h^{\text{opt}}$. Our state \mathbf{x}^{opt} will be a global optimum of the new problem. Likewise, if we start from a *local* optimum \mathbf{x}^{opt} with cost h^{opt} and introduce the equality constraint $h(\mathbf{x}) = h^{\text{opt}}$, the optimum \mathbf{x}^{opt} will also be a local optimum to the new problem. In this way, a quantitative cost term for enzyme levels, for instance, can formally be

replaced by a bound on enzyme levels. The advantage a cost constraint over an additive cost term is that benefits and costs can be measured on different scales and in different units. There is no need to quantify their “relative importance.”

In some case, however, it can be practical to do exactly the opposite: to replace a constraint by a cost term. If our system state hits some upper bound $h(\mathbf{x}) \leq h^{\text{max}}$, we can express this by an equality constraint $h(\mathbf{x}) = h^{\text{max}}$. This constraint can be treated by the method of *Lagrange multipliers*: instead of maximizing $g(\mathbf{x})$ under the constraint $h(\mathbf{x}) = h^{\text{opt}}$, we search for a number λ , called the Lagrange multiplier, such that $f'(\mathbf{x}) = g(\mathbf{x}) - \lambda h(\mathbf{x})$ becomes maximal with respect to \mathbf{x} (where the constraint $h(\mathbf{x}) = h^{\text{opt}}$ must still hold). The necessary optimality condition reads $0 = \partial f'(\mathbf{x}) / \partial x_i = \partial g(\mathbf{x}) / \partial x_i - \lambda \partial h(\mathbf{x}) / \partial x_i$. Effectively, the original constraint is replaced by a virtual cost term that is subtracted from the objective function. The relative importance of costs and benefits is determined by the value of the Lagrange multiplier in the original optimal state.

Pareto Optimality

If an organism is subject to multiple objectives, its potential phenotypes can be depicted as points in a space spanned by these objective functions (see Figure 11.3a). If there are trade-offs between the objectives, only a limited region of this space will be accessible. Boundary points of this region in which none of the objectives can be improved without compromising the others form the Pareto front. We will see an example – a Pareto front of metabolic fluxes in bacterial metabolism – further below. Pareto optimality problems allow for many solutions, and unlike local minima in single-objective problems, these solutions are not directly comparable: one solution is not better or worse than another one – it is just different. Moreover, each objective can be scored on a different scale – there is no need to make them directly comparable.

Pareto optimality can be depicted very intuitively with the concept of *archetypes*, introduced by Shoval *et al.* to refer to phenotypes that optimize, individually, one of the different objectives [23,24]. The role of archetypes becomes clear when phenotypes are displayed in another space, the space of phenotype variables (Figure 11.3b). In a problem with three objectives and two phenotypical traits, the three archetypes would define a triangle in a plane. If the objectives, as functions in this space, have circular contour lines, as we assume as an approximation, the possible Pareto optima will fill this triangle. Each of them will thus be a convex combination of the archetype vectors. The same concept applies also to higher dimensions (i.e., more objectives and traits).

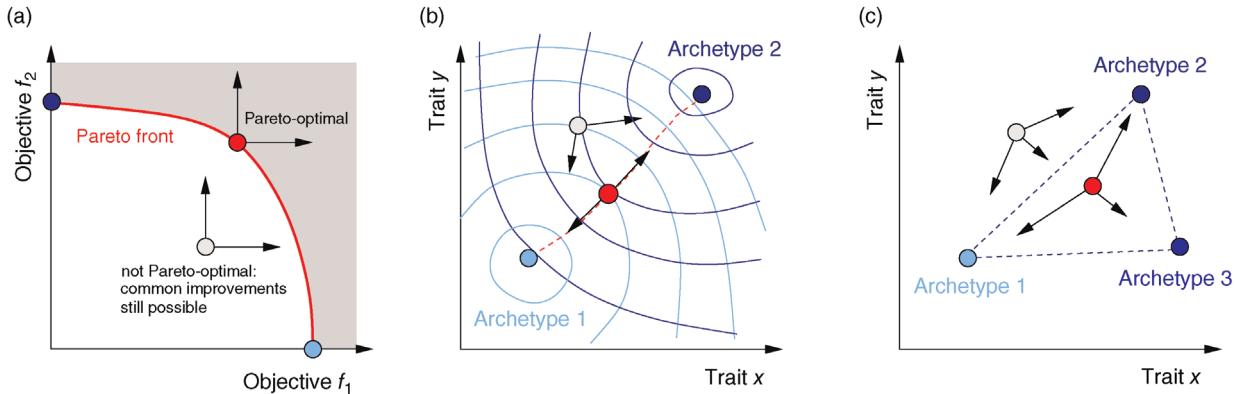


Figure 11.3 Pareto optimality for a system with multiple objectives. (a) Variants of a system, shown as points in the space of objective functions. (b) The same system variants, shown in phenotype space (axes correspond to phenotypical traits) [23]. The two objective functions are shown by contour lines; their optimum points are called archetypes. In a Pareto-optimal point (red), the gradients of both objective functions are parallel, but with opposite directions: none of the objectives can be simultaneously improved without compromising the other one. In a nonoptimum point (gray), both objectives could still be improved. (c) In a problem with three objectives, the archetypes form a triangle in phenotype space. The criterion for a Pareto optimum (red) is that the gradients, weighted by positive prefactors, must sum to zero. This resembles the balance between objective and constraint in Lagrange optimization (Figure 11.2b). With simple objective functions (circular contour lines, not shown), this condition is satisfied exactly within the triangle, so the Pareto optima are also convex combinations of the archetypes.

Shoval *et al.* applied this approach to different kinds of data, for instance, Darwin's data on the body shapes of ground finches. When plotting body size against a quantitative measure of beak shape, they obtained a triangle of data points. The three archetypes, with their extreme combinations of the traits, could be explained as optimal adaptations to particular diets. All other species, located within the triangle, represent less specialized adaptations, possibly adaptations to mixed diets. In the theoretical model, triangles are expected to appear for simple objective functions with circular contour lines. With more general objective functions, the triangles will be distorted [24]. In general, whether a cloud of data points forms a triangle (or, more generally, an m -dimensional convex polytope in n -dimensional space) needs to be tested statistically.

Compromises in the Choice of Catalytic Constants

As an example of optimality considerations, including compromises and possibly nonoptimality, let us consider the choice of k^{cat} values in metabolic systems. Empirically, k^{cat} values tend to be higher in central metabolism, where fluxes are large and enzyme levels are high [25]. This suggests that k^{cat} values in other regions stay below their biochemically possible maximum, contradicting the optimality assumption whereby each k^{cat} value should be as high as possible. Thus, what prevents some k^{cat} values from increasing? In some cases, enzymes with higher k^{cat} values become larger and therefore more costly. In other cases, high k^{cat} values could compromise substrate specificity (which may be more relevant in secondary than in

central metabolism), allosteric regulation (as in the case of phosphofructokinase, a highly regulated enzyme with a rather low catalytic constant), or a favorable temperature dependence of kinetic properties. Finally, enzymes can also exert additional functions (e.g., metabolize alternative substrates or act as a signaling, structural, or cell cycle protein), which may compromise their k^{cat} values.

However, we may also assume an evolutionary balance between a selection for high k^{cat} values and a tendency toward lower k^{cat} values caused by random mutations. Formally, this nonoptimal balance can be treated in an optimality framework by framing the effect of mutations as an effective *evolutionary cost*. This cost appears as a fitness term, but it actually reflects the fact that random mutations of an enzyme are much more likely to decrease than to increase its k^{cat} value. The prior probability $\rho(k^{\text{cat}})$, that is, the number of gene sequences that realize a certain k^{cat} value, defines the evolutionary cost $h^{\text{evo}}(k^{\text{cat}}) = -\theta \ln \rho(k^{\text{cat}})$. The constant parameter θ describes the strictness of selection [25]. The most likely k^{cat} value after an evolution is thus not expected to maximize an organism's actual fitness $f(k^{\text{cat}})$, but the apparent fitness $f^{\text{evo}} = f - h^{\text{evo}}$.

The concept of apparent fitness is analogous to the concept of free energy in thermodynamics. To describe systems at given temperature T , the principle of minimal energy E (a stability condition from classical mechanics) is replaced by a principle of minimal free energy $E + TS$, where the entropy $S = k_B \ln W$ refers to the number W of microstates showing the energy E . The term TS appears formally as an energy, but it actually reflects the

Table 11.1 Some optimality approaches used in metabolic modeling.

Name	Formalism	Objective	Control variables	Main constraints	Reference
FBA	Stoichiometric	Flux benefit	Fluxes	Stationary fluxes	[27]
FBA with minimal fluxes	Stoichiometric	Flux cost	Fluxes	Stationary, flux benefit	[28]
Resource balance analysis	Stoichiometric	Growth rate	Fluxes	Stationary, whole cell	[29]
Enzyme rate constants	Reaction kinetics	Different objectives	Rate constants	Haldane relations	[30]
Enzyme allocation	Kinetic	Pathway flux	Enzymes (static)	Total enzyme	[31]
Temporal enzyme allocation	Kinetic	Substrate conversion	Enzymes (dynamic)	Total enzyme	[32]
Transcriptional regulation	Kinetic	Substrate conversion	Rate constants	Feedback structure	[33]
Enzyme adaptation	Kinetic	General fitness	Enzymes (changes)	Model structure	[20]
Growth rate optimization	Kinetic, growing cell	Growth rate	Protein allocation	Total protein	[34]

fact that macrostates of higher energy contain much more microstates (Section 15.6).

Of course, the notion of evolutionary costs is not restricted to k^{cat} values, but applies very generally. A similar type of effective cost is the *evolutionary effort*, quantified by counting the number of mutations or events necessary to attain a given state [26].

11.1.2 Metabolism Is Shaped by Optimality

A field within systems biology in which optimality considerations are central is metabolic modeling. Aside from standard methods such as FBA, which take optimality as one of their basic assumptions, there are dedicated optimality-based studies on pathway architecture, choices between pathways, metabolic flux distributions, the enzyme levels supporting them, and the regulatory systems controlling these enzyme levels (Table 11.1). In the forthcoming sections, we shall see a number of examples of these approaches.

When describing metabolic systems as optimized, we first need to specify an objective function. The objective does not reflect an absolute truth, but is related to the evolution scenario we consider. In natural environments, organisms will experience times of plenty, where fast growth is possible and maybe crucial for evolutionary success, and times of limitation, where efficient, sustainable usage of resources is most important. Both situations can be mimicked by artificial evolution experiments and studied with models. If bacteria are experimentally selected for fast growth – and nothing else – their growth rate will be a meaningful fitness function: mutants that grow faster will be those that survive, and those are also the ones we are going to model as optimized. However, setting up an experiment that selects for growth *and nothing else* is difficult: in a bioreactor with continuous dilution, cells could evolve to stick to the walls of the

bioreactor, which would allow them not to be washed out. Even if these cells replicate slowly, their strategy provides a selection advantage in this setting. In serial dilution experiments, this drawback is avoided by transferring the microbial population between different flasks every day. However, if the cultures are grown as batch cultures, the cells will experience different conditions during the course of each day and may switch between growth and stationary phases. Again, many more traits, not just for a fast growth under constant conditions, will be selected for in the experiment.

Thus, it is the experimental setup that determines what is selected for and what objective should be used in our models. There is also a version of serial dilution that can select for nutrient efficiency instead of fast inefficient growth [35]. Unlike normal serial dilution experiments, this experiment requires – and allows for – models in which nutrient efficiency is the objective function. Not surprisingly, selection processes in natural environments are even more complicated: a species' long-term survival depends on many factors, including the populating of new habitats, defenses against pathogens, and complex social behavior. How evolution depends on interactions between organisms and environment can be studied through evolutionary game theory, which we discuss in Section 11.4.

Given a fitness function for an entire cell, how can we derive from it objectives for individual pathways? Metabolic pathways contribute to a cell's fitness by performing some local function (e.g., production of fatty acids), which contributes to more general cell functions (e.g., maintenance of cell membranes) and thereby to cell viability in general. Compromises between opposing subobjectives may be inevitable, for example, between high production fluxes, large product yields, and low enzyme investment. The relative importance of the different objectives may depend on the cell's current environment [36]. Once we have identified an objective, we can

ask how pathway structure, flux distribution, or enzyme profile should be chosen, and also how regulation systems can ensure optimal behavior under various physiological conditions.

What Functional Reasons Can Explain Metabolic Network Structures?

If the laws of chemistry (e.g., conservation of atom numbers) were the only restriction, an enormous number of metabolic network structures could exist (see Section 8.1). The actual network structures, however, are determined by the set of enzymes encoded in cells' genomes or, more specifically, by the set of enzymes expressed by cells. The sizes and capabilities of metabolic networks vary widely, but their structures follow some general principles. Important precursors (such as nucleotides and amino acids in growing cells) form the central nodes that are connected by pathways. Many reactions involve cofactors, which connect these pathways very tightly. A relatively small set of cofactors has probably been used since the early times of metabolic network evolution [37].

By which chemical reactions will two substances be connected? In fact, the number of possibilities is vast: already for the route from chorismate to tyrosine, realized by three reactions in *E. coli*, a computational screen yielded more than 350 000 potential pathway variants [38]. Out of numerous possible pathways, only very few are actually used by organisms, and we may wonder what determines this selection. Some of the choices may be due to specific functional advantages [39]: for instance, glucose molecules are phosphorylated at the very beginning of glycolysis; this prevents them from diffusing through membranes, thus keeping them inside the cell. But there are also more general reasons informing pathway structure. We shall discuss three important principles: the preferences for yield-efficient, short, and thermodynamically efficient pathways [38,40].

Yield Efficiency

Alternative pathways may produce different amounts of product from the same substrate amount. At given substrate uptake rates, high-yield pathways produce more product per time and should therefore be preferable in evolution. However, there may be a trade-off between a high product yield and a high product production *per amount of enzyme* needed to catalyze the pathway flux. The reason is that yield-efficient pathways tend to operate closer to equilibrium, where enzymes are used inefficiently. At equal enzyme investment, yield-inefficient pathways may thus provide larger fluxes, enabling faster cell growth. This is why some metabolic pathways exist in alternative versions with different yields,

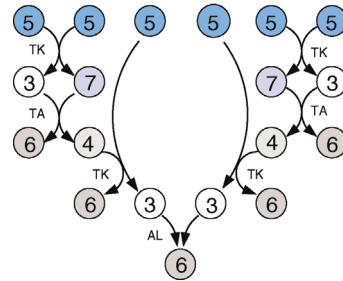


Figure 11.4 A metabolic pathway of minimal size. The non-oxidative phase pentose phosphate pathway converts six pentose molecules into five hexose molecules in a minimal number of reactions. Metabolites shown (numbers of carbon atoms in parentheses): D-ribose 5-phosphate and D-xylulose 5-phosphate (5); D-glyceraldehyde 3-phosphate (3); D-sedoheptulose 7-phosphate (7); D-fructose 6-phosphate and D-fructose 1,6-bisphosphate (6); D-erythrose 4-phosphate (4). Enzymes: transketolase (TK); transaldolase (TA); aldolase (AL).

even in the same organism. We will come back to this point, using glycolysis as an example.

Preference for Short Pathways

The preference for short pathways is exemplified by the non-oxidative part of the pentose phosphate pathway, a pathway with the intricate task of converting six pentose molecules into five hexose molecules. If we additionally require that intermediates have at least three carbon atoms and that enzymes can transfer two or three carbon atoms, the pentose phosphate pathway solves this task in a minimal number of enzymatic steps [41] (Figure 11.4). A trend toward minimal size is also visible more generally in central metabolism: Noor *et al.* showed that glycolysis, pentose phosphate pathway, and TCA cycle are composed of reaction modules that connect a number of central precursor metabolites and that, in each case, consist of minimal possible numbers of enzymatic steps [42].

Pathways Must Be Thermodynamically Feasible

A third general factor that affects pathway structure is thermodynamic efficiency. The metabolic flux directions, in each single reaction, are determined by the negative Gibbs energies of reactions (also called reaction affinity or thermodynamic driving force; see Section 15.6). To allow for a forward flux, the driving force must be positive. This holds not only for an entire pathway, but also for each single reaction within a pathway. However, the driving force has also a quantitative meaning: it determines the percentage of enzyme capacity that is wasted by the backward flux within that reaction. If the force approaches zero (i.e., a chemical equilibrium), the enzyme demand for realizing a given flux rises fast. Therefore, a sufficiently large positive driving force must exist in every reaction (see Section 15.6).

Glycolysis exemplifies the role of thermodynamics in determining pathway structure. The typical chemical potential difference between glucose and lactate would allow for producing four molecules of ATP per glucose molecule; however, this would leave practically no energy to be dissipated and thus to drive a flux in the forward direction. Assuming that the flux is proportional to the thermodynamic force ΔG along the pathway – which, far from equilibrium, is only a rough approximation – a maximal rate of ATP production will be reached by glycolytic pathways that produce only half of the maximal number of ATP molecules [43]. In glycolytic pathways found in nature, this is indeed the typical case. The two ATP molecules, however, are not produced directly. First, two ATP molecules are consumed in upper glycolysis, and then four ATP molecules are produced in lower glycolysis. A comparison with other potential pathway structures shows that this procedure allows for a maximal glycolytic flux at given enzyme investment [40,44]. A systematic way to assess the thermodynamic efficiency of a pathway is the max–min driving force (MDF) method [45]. Based on equilibrium constants and bounds on metabolite concentrations, it distributes the reaction driving forces in such a way (by choosing the metabolite concentrations) that small forces are avoided. Specifically, concentrations are arranged in such a way that the lowest reaction driving force in the pathway is as high as possible. The minimal driving force can be used as a quality criterion to compare pathway structures. If the minimal driving force is low, a pathway is unfavorable in terms of enzyme demand. Like shorter pathways, also pathways with favorable thermodynamics allow cells to reduce their enzyme investments. Also the ATP investment in upper glycolysis can be explained with this concept: without the early phosphorylation by ATP, a sufficient driving force in the pathway would require very high concentrations of the first intermediates. With the phosphorylation, the chemical potentials of the first intermediates can be high, but their concentrations will be much lower!

Enzyme Investments

The preferences for short pathways and the avoidance of small driving forces may be explained by a single requirement: the need to realize flux distributions at a minimal enzyme cost. If enzyme investments are a main factor driving the choice of pathways, how can we quantify them? Typically, the notion of protein investment captures marginal biological costs, that is, a fitness reduction caused by an additional expression of proteins. For growing bacteria, the costs can be defined as the measured reduction in growth rate caused by artificially induced proteins. To ensure that the measurement concerns only

costs, and not benefits caused by the proteins, one studies proteins without a normal physiological function (e.g., fluorescent marker proteins) or enzymes without available substrate [1,46].

For use in models, various proxies for enzyme cost have been proposed, including the total amount of enzyme in cells or specific pathways [19] or the total energy utilization [47]. To define simple cost functions, we can assume that protein cost increases linearly with the protein production rate, that is, with the amount of amino acids invested per time in translation [48]. With these definitions, protein cost will increase with protein abundance, protein chain length, and the effective protein turnover rate (which may include dilution in growing cells) [49].

11.1.3

Optimality Approaches in Metabolic Modeling

Based on known requirements for efficient metabolism (high product yields, short pathways, sufficient thermodynamic forces, and low enzyme investment), one may attempt to predict metabolic flux distributions as well as the enzyme levels driving them.

Flux Optimization

Flux prediction has many possible applications: for instance, as a sanity check in network reconstruction or for scoring possible pathway variants in metabolic engineering [50,51]. Understanding how flux distributions are chosen in nature can also shed light on the allocation of protein resources, for instance, on the enzyme investments in different metabolic pathways (see Figure 8.15). Flux distributions in metabolic networks can be predicted by FBA (see Chapter 3). In a typical FBA, flux distributions are predicted by maximizing a linear flux benefit $b(\mathbf{v}) = \mathbf{z} \cdot \mathbf{v}$ under the stationarity constraint $\mathbf{N}\mathbf{v} = 0$ and bounds on individual fluxes. FBA can predict optimal yields and the viability of cells after enzyme deletions. The flux distribution itself, however, is usually underdetermined, and flux predictions are often unreliable. One reason is that FBA, by placing a bound on the substrate supply, effectively maximizes the product yield *per substrate* invested. When cells need to grow fast, however, it is more plausible to assume that cells maximize the product yield *per enzyme investment*.

To account for this fact, some variants of FBA suppress unnecessary pathway fluxes in their solutions. The *principle of minimal fluxes* [28] states that a flux distribution \mathbf{v} must achieve a given benefit $b = \mathbf{z} \cdot \mathbf{v}$ at a minimal sum of absolute fluxes $\sum |v_j|$. In this sum, the fluxes can also be individually weighted (see Chapter 3). This flux cost may be seen as a proxy for the (unknown) enzyme costs

required to realize the fluxes. In contrast to standard FBA, FBA with flux minimization resolves many of the underdetermined fluxes and yields more realistic flux predictions [36,52]. FBA with molecular crowding [53], an alternative method, pursues a similar, yet contrary approach: it translates absolute fluxes into approximate enzyme demands and puts upper bounds on the latter while maximizing the linear flux benefit. Resource balance analysis (RBA), finally, not only covers metabolism, but also brings macromolecule production into the picture [29]. It assumes a growing cell in which macromolecules, including enzymes, are diluted. Therefore, metabolic reactions depend on a continuous production of enzymes and, indirectly, on ribosome production. Macromolecule synthesis, in turn, depends on precursors and energy to be supplied by metabolism. RBA checks, for different possible growth rates, whether a consistent cell state can be sustained, and determines the maximal possible growth rate and the corresponding metabolic fluxes and macromolecule concentrations.

All these methods assume, in one way or another, a trade-off between flux benefits and flux costs, which sometimes appear in the form of constraints. However, this trade-off can be studied more directly by adopting a multi-objective approach. Instead of being optimized in only one way, flux distributions may be chosen to optimize different objectives under different growth conditions [36] or compromises between different objectives. For flux distributions in the central metabolism of *E. coli*, measured by isotope labeling of carbon atoms, this has been demonstrated [52]. Each flux distribution can be characterized by three objectives: ATP yield and biomass yield (both to be maximized), and the sum of absolute fluxes (to be minimized). By assessing all possible flux distributions in an FBA model, the Pareto surface for these objectives can be computed. As shown in Figure 11.5, the measured flux distributions are close to the Pareto surface; each of them represents a different compromise under a different experimental condition. The fact that the flux distributions are not exactly *on* the Pareto surface can mean two things: either that these are not optimized or that other optimality criteria, aside from those considered in the analysis, play a role. One such possibility could be anticipation behavior: by deviating from the momentary optimum, flux distributions could facilitate future flux changes and thus the adaptation to new environmental conditions.

Optimization of Enzyme Levels

If flux distributions are thought to be optimized, optimality should also hold for enzyme levels and the regulation systems behind them. In models, the question of optimal enzyme levels can be addressed in two ways. Either we

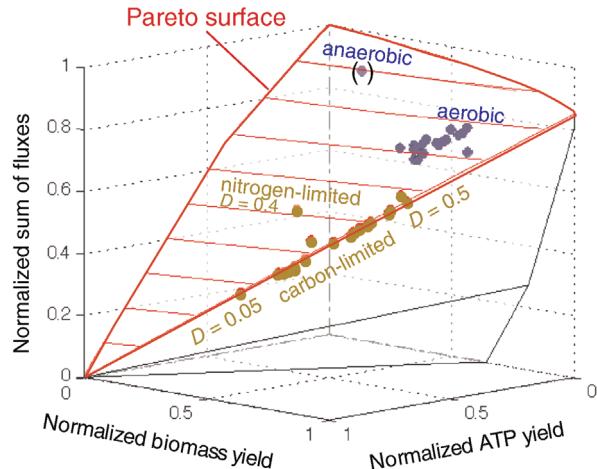


Figure 11.5 Metabolic flux distributions in *E. coli* central metabolism are close to being Pareto-optimal. Axes correspond to three metabolic objectives: biomass yield, ATP yield, and the sum of absolute fluxes. Dots represent measured flux distributions under different growth conditions. The polyhedron comprises all possible combinations of objective values, determined from a stoichiometric model. The Pareto surface is marked in red. The measured flux distributions are close to, but not exactly on the Pareto surface. (From Ref. [52].)

assume that some predefined flux distribution must be realized or we directly optimize cell fitness as a function of enzyme levels. In both cases, fluxes and enzyme investments are linked by enzyme kinetics.

In the first approach, we use flux analysis to determine a flux distribution and then search for enzyme profiles (and corresponding metabolite profiles) that can realize this flux distribution with a given choice of rate laws [54,55]. Since the fluxes do not uniquely determine the enzyme profile, an additional optimality criterion is needed to pick a solution, for instance, minimization of total enzyme cost. Since reactions with low driving forces would automatically imply larger enzyme levels, they will automatically be avoided: either by shutting down the enzyme or by an arrangement of enzyme profiles that ensures sufficient reaction affinities in all reactions.

Effectively, this approach translates a given flux distribution into an optimal enzyme profile and thereby into an overall enzyme cost. In theory, cost functions of this sort could also be used to define nonlinear flux costs for FBA, but these would be hard to compute in practice. It can be generally proven, though, that such flux cost functions, at equal flux benefits, are minimized by elementary flux modes [56,57].

In the second approach, the metabolic objective is directly written as a function $g(E)$ of the enzyme levels E_i , mediated through enzyme-dependent steady-state

concentrations and fluxes. The enzyme levels are optimized for a high fitness value, either at a given total enzyme investment or with a cost term for enzymes (following Eq. (11.1), where g is a metabolic objective scoring the stationary fluxes and concentrations and h scores the enzyme investments). Given the fitness function, we can compute optimal protein profiles and study how they should be adapted after perturbations. Modeling approaches of this sort will be discussed further below.

Measurements of Enzyme Fitness Functions

The fitness as a function of enzyme levels can also be explored by experiments. If the fitness function is given by the cell growth rate, the fitness landscape can be experimentally screened by varying the expression of a protein and recording the resulting growth rates [1]. The protein levels in *E. coli*, for instance, can be varied by manipulating the ribosome binding sites. By modifying the ribosome binding sites of several genes combinatorially, one can produce a library of bacterial clones with different expression profiles of these genes. In this way, suitable expression profiles for a recombinant carotenoid synthesis pathway in *E. coli* have been determined [58]. A similar approach was applied to constitutive gene promoters in *Saccharomyces cerevisiae*. A large number of variants of the engineered violacein biosynthetic pathway, differing in their combinations of promoter strengths, were obtained by combinatorial assembly. Based on measured production rates, a statistical model could be trained to predict the preferential production of different pathway products depending on the promoter strengths [59].

Another way to measure the benefits from single genes is the genetic tug-of-war (gTOW) method [60], which allows for systematic measurements of maximal tolerable gene copy numbers, that is, copy numbers at which the selective pressure against an even higher expression begins to be strong. The method, established for the yeast *S. cerevisiae*, uses a plasmid vector whose copy number in cells ranges typically from 10 to 40. In a yeast strain with a deletion of the gene leu2 (coding for a leucine biosynthesis enzyme needed for growth on leucine-free media) and with plasmids containing a low-expression variant of the same gene, larger copy numbers of the plasmid lead to a growth advantage. Due to a selection for higher copy numbers, the typical numbers rise to more than 100. Now a second gene, the target gene under study, is inserted into the plasmid, and thus expressed depending on the plasmid's copy number. The gene's expression can have beneficial or (especially at very high copy numbers) deleterious fitness effects, which are overlaid with the (mostly beneficial) effect of leu expression.

Eventually, the cells arrive at an optimal copy number, the number at which a small further change in copy number has no net effect because marginal advantage (due to increased leu expression) and marginal disadvantage (from increased target gene expression) cancel out. By measuring this optimal copy number, one can compare the fitness effects (or, a bit simplified, the maximal tolerable gene copy numbers) of different target genes.

11.1.4 Metabolic Strategies

Fermentation and Respiration as Examples of High-Rate and High-Yield Strategies

Cell choices between high-flux and high-yield strategies are exemplified by the choice between two metabolic strategies in central metabolism, fermentation and respiration. Glycolysis realizes fermentation, producing incompletely oxidized substances such as lactate or ethanol. In terms of yield, this is rather inefficient: only two ATP molecules are produced per glucose molecule. Exactly because of this, however, large amounts of Gibbs energy are dissipated, which suppresses backward fluxes and thereby increases the metabolic flux. Respiration, performed by the TCA cycle and oxidative phosphorylation, uses oxygen to oxidize carbohydrates completely into CO₂. It has a much higher yield than glycolysis, producing up to 36 molecules of ATP per glucose molecule. However, this leaves little Gibbs energy for driving the reactions, so more enzyme may need to be invested to obtain the same ATP production flux. The details depend on the concentrations of extracellular compounds (e.g., oxygen and carbon dioxide), which affect the overall Gibbs free energy of the pathway.

Cells tend to use high-yield strategies under nutrient limitation and in multicellular organisms (although microbes such as *E. coli* can also be engineered to use high-yield strategies [61]). Low-yield strategies, such as additional fermentation on top of respiration in yeast (Crabtree effect) [62] and the Warburg effect in cancer cells [63], are preferred when nutrients are abundant and in situations of strong competition. Low-yield strategies may be profitable when enzyme levels are costly or constrained, or when the capacity for respiration is limited. For instance, when the maximal capacity of oxidative phosphorylation has been reached (because of limited space on mitochondrial membranes), additional fermentation could increase the ATP production flux [64]. Moreover, a higher thermodynamic driving force may lead to a higher ATP production per time at the same total enzyme investment, which in turn would allow for faster growth [53,65].

Paradoxically, low-yield strategies can outperform high-yield strategies even if their performance is worse. Grown separately on equal substrates, a high-yield strategy leads to higher cell densities (in continuous cultures) and longer survival (in batch cultures). However, when grown on a shared food source, microbes using a low-yield strategy initiate a competition in which they use their waste of nutrients as a weapon to outperform yield-efficient microbes. Therefore, inefficient metabolism can provide a selection advantage for individual cells that is paid by a permanent loss of the cell population. But this is not the end of the story: excreted fermentation products, such as lactate or ethanol, can serve as nutrients for other cells or for later times. Thus, depending on the frame of description, fermentation can also appear as a cooperative strategy. In Section 11.4, we will see how the social behavior of microbes is framed by evolutionary game theory.

Trade-off between Product Yield and Enzyme Cost

Aside from the most typically studied variant of glycolysis, called Embden–Meyerhof–Parnas (EMP) pathway, there are other variants of glycolysis with different ATP yields. EMP glycolysis produces two ATP molecules from one molecule of glucose and is common in eukaryotes. An alternative variant, the Entner–Doudoroff (ED) pathway, produces one ATP molecule only (see Figure 11.6). If the two variants are compared at equal ATP production rates, the ED pathway consumes twice as much glucose. Nevertheless, many bacteria use the ED pathway, either as an alternative or in addition to the EMP pathway.

We already saw that pathways with lower yields may be justified by their larger energy dissipation, which can lower the enzyme demand. By modeling the energetics of both pathways, Flamholz *et al.* [55] showed that the enzyme demand of the EMP pathway, at equal ATP production rates, is about twice as high. Thus, the choice between the two pathways will depend on what is more critical in a given environment – an efficient usage of glucose or a low enzyme demand. If ATP production matters less (e.g., because other ATP sources, such as respiration or photosynthesis, exist), or if glucose can be assumed to be abundant (as in the case of *Zymomonas mobilis*, which is adapted to very large glucose concentrations), the ED pathway will be preferable. This prediction has been confirmed by a broad comparison of microbes with different lifestyles [55].

Molenaar *et al.* modeled the choice between high-yield and low-yield strategies as a problem of optimal resource allocation [34]. In their models, cells can choose between two pathways for biomass precursor production: one with

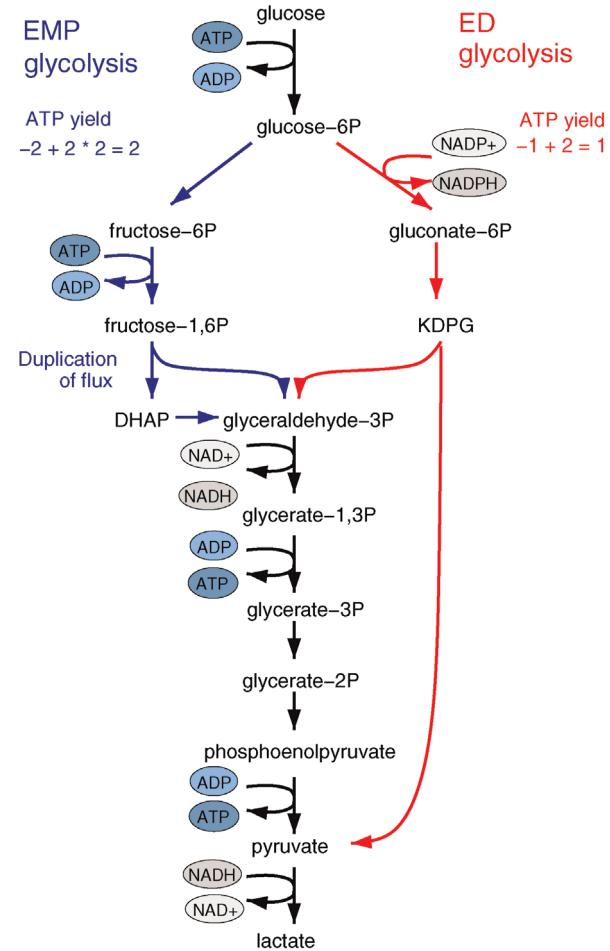


Figure 11.6 Comparison between two variants of glycolysis. The EMP pathway shows a higher ATP yield (two ATP molecules per molecule of glucose) than the ED pathway (one ATP molecule per glucose molecule). However, its enzyme demand per ATP flux is more than twice as high [45].

high cost and high yield and another one with low cost and low yield. Given a fixed substrate concentration and aiming at maximal growth, cells can allocate a fixed total amount of protein resources between the two pathways. After numerical optimization, the simulated cells employ the high-yield pathway when substrate is scarce, and the low-yield pathway when the substrate level increases. The two cases correspond, respectively, to slow-growth and fast-growth conditions.

11.1.5 Optimal Metabolic Adaptation

Metabolic enzymes show characteristic expression patterns across the metabolic network, but also characteristic temporal patterns during adaptation. Mechanistically,

expression patterns follow, for instance, from the regulation functions encoded in the genes' promoter sequences (see Section 9.3). If a pathway is controlled by a single transcription factor, the enzymes in that pathway will tend to show a coherent expression. However, we may still ask why the regulation system works in this way: why is it that transcription factor regulons and metabolic pathway structures match, even if the metabolic network and its transcriptional regulation evolve independently? In terms of function, coherent expression patterns may simply be the best way to regulate metabolism. This hypothesis can be studied by models in which enzyme profiles control the metabolic state and are chosen to optimize the metabolic performance.

Optimal Adaptation of Enzyme Activities

Using such models, we can address a number of questions. If a metabolic state is perturbed, how should cells adapt their enzyme levels? And, in particular, if an enzyme level itself is perturbed, how should the other enzymes respond? To predict such adaptations, we may consider a kinetic pathway model and determine its enzyme levels from an optimality principle as in Eq. (11.1). We start in an initially optimal state. When an external perturbation is applied, this will change the fitness landscape in enzyme space and shift the optimum point. Similarly, when a knockdown forces one enzyme level to a lower value, this will lead to a new, constrained optimum for the other enzymes. In both cases, the cell should adapt its enzyme activities to reach the new optimum. Given a model, the necessary enzyme adaptations can be computationally optimized.

A mathematical approach for such scenarios was developed in Ref. [20]: under the assumption of small perturbations, the optimal adaptations can be directly computed from the metabolic response coefficients and the local curvatures of the fitness landscape (i.e., the Hessian matrix). Consider a kinetic model with steady-state metabolite concentrations and fluxes (in a state vector $\mathbf{y}(\mathbf{x}, \mathbf{a})$) depending on control variables (e.g., enzyme activities, in a vector \mathbf{x}) and environment variables (in a vector \mathbf{a}). The possible control profiles \mathbf{x} are scored by a fitness function

$$f(\mathbf{x}, \mathbf{a}) = g(\mathbf{y}(\mathbf{x}, \mathbf{a})) - h(\mathbf{x}), \quad (11.2)$$

which describes the trade-off between a metabolic objective g and a cost function h . The optimality principle requires that, given the environment \mathbf{a} , the control profile \mathbf{x} must be chosen to maximize f . As a condition for a local optimum, the gradient $\mathbf{f}_x = (\partial f / \partial x_i)$ must vanish

and the Hessian matrix $\mathbf{F}_{xx} = (\partial^2 x / \partial x_i \partial x_j)$ must be negative definite. Now we assume a change $\Delta \mathbf{a}$ of the environment parameters, causing a shift in the optimal enzyme profile. The adaptation $\Delta \mathbf{x}$ that brings the cell from the old to the new optimal state can be determined as follows. To provide an optimal adaptation, the change Δf_x of the fitness gradient during the adaptation must vanish. We approximate the fitness landscape to second order around the initial optimum and obtain the optimality condition

$$0 = \Delta \mathbf{f}_x \approx \mathbf{F}_{xx} \Delta \mathbf{x} + \mathbf{F}_{xa} \Delta \mathbf{a} \quad (11.3)$$

and thus

$$\Delta \mathbf{x}^{\text{opt}} = -\mathbf{F}_{xx}^{-1} \mathbf{F}_{xa} \Delta \mathbf{a}. \quad (11.4)$$

The Hessian matrices \mathbf{F}_{xx} and \mathbf{F}_{xa} can be obtained from gradients and curvatures of g and h and from the first- and second-order response coefficients of the state variables in \mathbf{y} with respect to \mathbf{x} or \mathbf{a} . In the second adaptation scenario, we assume that one control variable x_i is constrained to some higher or lower value (in the case of enzyme levels, by overexpression or knockdown). How should the other control variables be adapted? If a change $\Delta \hat{x}_j$ is enforced on the j th enzyme, the optimal adaptation profile of all enzymes can be written as

$$\Delta x_i^{\text{opt}} = \frac{(\mathbf{F}_{xx}^{-1})_{ij}}{(\mathbf{F}_{xx}^{-1})_{jj}} \Delta \hat{x}_j, \quad (11.5)$$

where Δx_i^{opt} automatically assumes its perturbed value $\Delta \hat{x}_i$. Even if the Hessian matrix \mathbf{F}_{xx} is unknown, its symmetry implies a general symmetry between perturbation and response. We can see this as follows: in the initial optimum state, the matrix \mathbf{F}_{xx} must be negative definite, so the elements $(\mathbf{F}_{xx}^{-1})_{jj}$ will all be negative. Since \mathbf{F}_{xx} itself is symmetric, the coefficients $(\mathbf{F}_{xx}^{-1})_{ij}/(\mathbf{F}_{xx}^{-1})_{jj}$ must form a matrix with a symmetric sign pattern. This means that whenever a knockdown of enzyme A necessitates an adaptive upregulation of enzyme B, a knockdown of B should also necessitate an adaptive upregulation of A. An analogous relationship holds for cases of adaptive downregulation. Symmetric behavior, as predicted, has been found in expression data.

Optimal Control Profiles Reflect the Objective and the System Controlled

What can we learn about proteins' function by observing their expression profiles? For instance, is there a reason

why correlations in expression should reflect metabolic pathway structures? Or, turning this question around, when a metabolic pathway is induced, should all enzyme levels be upregulated, or only some of them? In protein complexes, subunits often come in fixed proportions, and should therefore be expressed as such. The ribosomal subunits, for instance, are combined in a fixed stoichiometry; if the expression ratios did not match this stoichiometry, some subunits would remain unused and protein resources would be wasted. This argument can be framed mathematically as an optimality principle (e.g., minimal protein production at a fixed number of functional ribosomes). The stoichiometric expression is confirmed by proteomics and ribosome profiling [66] and visible, for instance, in Figure 8.15. A proportional expression can also make sense for forming a metabolic pathway – even if the enzymes do not form a complex. In theory, the same predefined pathway flux could be achieved by various different enzyme profiles. Accordingly, a small increase in flux could be achieved either by a proportional induction of all enzymes or by a selective induction of one enzyme, for instance, the one with the highest control. Thus, the question is not how the flux change *can* be achieved, but how it can be achieved *most economically*.

Here, the answer depends on details. In a simple case – if the external metabolite concentrations are fixed and if enzyme cost depends linearly on the enzyme levels – we can employ a scaling argument. We imagine a proportional scaling of fluxes and enzyme levels at constant metabolite levels. If we start with optimal expression ratios along the pathway, the fact that these ratios are optimal should be unaffected by scaling (see Section 10.2.6). In other words, if the flux needs to be increased, the best way of doing so is a proportional induction of all enzymes. In other cases, for instance, after a change of substrate, product, or cofactor levels, a nonproportional induction may be more favorable. According to Eq. (11.4), the optimal enzyme adaptations depend on how enzyme levels affect the steady-state variables, how these variables affect fitness, and how enzyme levels affect the fitness directly via their costs. The Hessian matrices, which encode this information, are partially based on the metabolic control coefficients. Accordingly, as a rough approximation, enzymes with a similar control over fitness-relevant variables can be expected to show similar differential expression. Loosely speaking, we can see optimal control problems as an inverted form of MCA. Instead of going from cause to effect (i.e., from enzyme changes to the resulting metabolic state changes), we go backward and ask, teleologically, how enzyme levels should be adjusted *in order to*

achieve a given metabolic change. Thus, what the differential activity patterns can be expected to portray – if any – is not the shape of the metabolic network, but the enzymes' control coefficients, which, to an extent, reflect the network structure [20].

11.2 Optimal Enzyme Concentrations

Summary

We exemplify how metabolic systems should be designed if they were designed according to optimality principles. We investigate the consequences of a demand for rapid conversion of substrate into product on the catalytic properties of single enzymes and on the appropriate amount of enzymes in a metabolic pathway. In the first two sections, we determine conditions on enzyme parameters and enzyme concentrations that yield maximal steady-state fluxes. The third section studies how temporal regulation of a metabolic pathway can support fast conversion of a substrate into a product.

11.2.1 Optimization of Catalytic Properties of Single Enzymes

An important function of enzymes is to increase the rate of a reaction. Therefore, evolutionary pressure should lead toward a maximization of the reaction rate $v \rightarrow \max$ [67–70]. High reaction rates may only be achieved if the kinetic properties of the enzymes are suitably adapted. We identify the optimal kinetic parameters that maximize the rate for the reversible conversion of substrate S into product P [71].

Two constraints must be considered. First, the action of an enzyme cannot alter the thermodynamic equilibrium constant for the conversion of S to P (Section 4.1, Eqs. (4.5) and (4.10)). Changes of kinetic properties must obey the thermodynamic constraint. Second, the values of the kinetic parameters are limited by physical constraints even for the best enzymes, such as diffusion limits or maximal velocity of intramolecular rearrangements. In the following, the maximal possible values are denoted by k_{\max} and all rate constants are normalized by their respective k_{\max} , such that the normalized kinetic constants have a maximal value of 1. Likewise, concentrations and rates are normalized to yield dimensionless quantities. For a simple reaction



that can be described with mass action kinetics with the thermodynamic equilibrium constant $q = K_{\text{eq}} = k_1/k_{-1}$, the rate equation reads

$$\begin{aligned} v &= E_{\text{total}} \cdot (S \cdot k_1 - P \cdot k_{-1}) \\ &= E_{\text{total}} \cdot k_{-1} \cdot (S \cdot q - P) = E_{\text{total}} \cdot k_1 \cdot \left(S - \frac{P}{q} \right). \end{aligned} \quad (11.7)$$

It is easy to see that v becomes maximal for fixed values of E_{total} , S , P , and q , if k_1 or k_{-1} becomes maximal. This is mathematically equivalent to a minimal transition time $\tau = (k_1 + k_{-1})^{-1}$. Note that usually only one of the two rate constants may attain its maximal value. The value of the other, submaximal constant is given by the relation to the equilibrium constant.

For a reversible reaction obeying Michaelis–Menten kinetics

$$v = E_{\text{total}} \cdot (Sq - P)k_{-1}k_{-2}/(Sk_1 + k_{-1} + k_2 + Pk_{-2}) \quad (11.8)$$

with $q = k_1k_2/(k_{-1}k_{-2})$, the optimal result depends on the value of P . For $q \geq 1$ and $P \leq 1/q$, the rate becomes maximal if k_1 , k_2 , and k_{-2} assume maximal values and k_{-1} is submaximal (region R_1 in Figure 11.7). For $P \geq q$, we obtain submaximal values of only k_{-2} (region R_2). For $1/q < P < q$, the optimal solution is characterized by submaximal values of k_{-1} and k_{-2} with $k_{-1} = \sqrt{P/q}$ and $k_{-2} = \sqrt{1/(P \cdot q)}$ (region R_3).

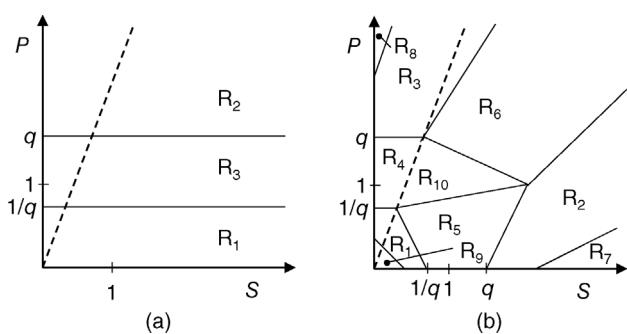


Figure 11.7 Subdivision of the plane of substrate and product concentrations (S, P) into regions of different solutions for the optimal microscopic rate constants (schematic representation). The dashed lines indicate the function $S \cdot q = P$. (a) Solution regions for the two-step mechanism. (b) Solution regions for the three-step mechanism.

Comparison of the optimal state with a reference state can assess the effect of the optimization. One simple choice for a reference state is $k_1 = k_2 = 1$ and $k_{-1} = k_{-2} = 1/\sqrt{q}$, that is, equal distribution of the free energy difference (see Section 4.1) represented by the equilibrium constant in the first and the second step. The respective reference rate reads

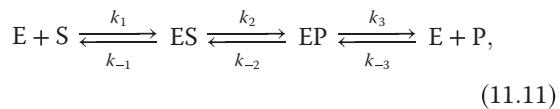
$$v^{\text{ref}} = (S \cdot q - P)/((S + 1) \cdot q + (P + 1) \cdot \sqrt{q}) \quad (11.9)$$

and the optimal rates in regions R_1 , R_2 , and R_3 are

$$\begin{aligned} v^{\text{opt},R_1} &= (S \cdot q - P)/((S + 1) \cdot q + 1 + P \cdot q), \\ v^{\text{opt},R_2} &= (S \cdot q - P)/((S + 1) \cdot q + q + P), \\ v^{\text{opt},R_3} &= (S \cdot q - P)/((S + 1) \cdot q + 2\sqrt{P \cdot q}). \end{aligned} \quad (11.10)$$

For example, in the case $P = q$ and $q = 100$, the maximal rate for optimal kinetic constants is $v^{\text{max}} = v^{\text{opt},R_3} = (S - 1)/(S + 3)$ and the reference rate is calculated as $v^{\text{ref}} = (S - 1)/(S + 11.1)$, which is lower than the maximal rate.

For the reversible three-step mechanism involving the binding of the substrate to the enzyme, the isomerization of the ES complex to an EP complex, and the release of product from the enzyme,



the reaction rate is given as

$$v = E_{\text{total}} \cdot \frac{S \cdot k_1 k_2 k_3 - P \cdot k_{-1} k_{-2} k_{-3}}{k_2 k_3 + k_{-1} k_3 + k_{-1} k_{-2} + S \cdot k_1 (k_2 + k_3 + k_{-2}) + P \cdot k_{-3} (k_2 + k_{-1} + k_{-2})}. \quad (11.12)$$

It turns out that the optimal solution for this mechanism depends on the values of both S and P . There are 10 different solutions, shown in Table 11.2 and Figure 11.7.

Among the 10 solutions, there are three solutions with a submaximal value of one backward rate constant, three solutions with submaximal values of two backward rate constants, three solutions with submaximal values of one backward and one forward rate constant, and one solution with submaximal values of all three backward rate constants. The constraint imposed by the thermodynamic equilibrium constant leads to the following effects. At very low substrate and product concentrations, a

Table 11.2 Optimal solutions for the rate constants of the three-step enzymatic reaction as functions of the concentrations of substrate and product for $q \geq 1$.

Solution	k_1	k_{-1}	k_2	k_{-2}	k_3	k_{-3}
R ₁	1	$1/q$	1	1	1	1
R ₂	1	1	1	$1/q$	1	1
R ₃	1	1	1	1	1	$1/q$
R ₄	1	$\sqrt{P/q}$	1	1	1	$\sqrt{1/Pq}$
R ₅	1	$\sqrt{\frac{S+P}{q(1+P)}}$	1	$\sqrt{\frac{1+P}{q(S+P)}}$	1	1
R ₆	1	1	1	$\sqrt{\frac{2P}{q(1+S)}}$	1	$\sqrt{\frac{1+S}{2Pq}}$
R ₇	$\sqrt{\frac{2q(1+P)}{S}}$	1	1	$\sqrt{\frac{2(1+P)}{qS}}$	1	1
R ₈	1	1	$\sqrt{\frac{2q(1+S)}{P}}$	1	1	$\sqrt{\frac{2(1+S)}{qP}}$
R ₉	1	$\sqrt{\frac{2(S+P)}{q}}$	1	1	$\sqrt{2q(S+P)}$	1
R ₁₀	1	^a	1	$\frac{P}{qk_{-1}^2}$	1	$\frac{1}{qk_{-1}k_{-2}}$

^a k_{-1} is solution of the equation $k_{-1}^4 + k_{-1}^3 - k_{-1}(P/q) - [(S \cdot P)/q] = 0$.

maximal rate is achieved by enhancing the binding of S and P to the enzyme (so-called high (S, P)-affinity solution). If S or P is present in very high concentrations, they should be weakly bound (low S- or P-affinity solutions). For intermediate values of S and P, only backward constants assume submaximal values. For concentrations of S and P equal to unity, the optimal solution reads

$$k_{-1} = k_{-2} = k_{-3} = q^{-1/3} \quad \text{and} \quad k_1 = k_2 = k_3 = 1. \quad (11.13)$$

This case represents an equal distribution of the drop in free energy in all three elementary steps.

In summary, we find that when normalized substrate concentrations are close to 1, then the maximal rates are favored by maximal forward rate constants and equal distribution of the thermodynamic constraints to the backward rate constants. If S or P deviates from the mean value, optimal rate constants compensate for the thermodynamic burden.

11.2.2 Optimal Distribution of Enzyme Concentrations in a Metabolic Pathway

By means of regulated gene expression and protein degradation, cells can adjust the amount of enzyme

allocated to the reactions of a metabolic pathway according to the current metabolic supply or demand. In many cases, the individual amounts of enzymes are regulated such that the metabolic fluxes necessary to maintain cell functions are achieved while the total enzyme amount is low. A reason for keeping enzyme concentrations this low is that proteins are osmotically active substances. One strategy to achieve osmotic balance is, therefore, to keep the total amount of enzyme constrained. Furthermore, enzyme synthesis is expensive for the cell, with respect to both energy and material. Therefore, it is sensible to assume that various pathways or even individual reactions compete for the available resources.

We can study theoretically how a maximal steady-state flux through a pathway is achieved with a given fixed total amount of enzyme [72]. The optimization problem is to distribute the total protein concentration $E_{\text{total}} = \sum_{i=1}^r E_i$ optimally among the r reactions. We will exemplify this for the simple unbranched pathway presented in Chapter 4, Eq. (4.100). To assess the effect of optimization we will again compare the optimal state to a reference state where the given total concentration of enzymes is distributed uniformly such that $E_i = E_{\text{total}}/r$.

The optimal enzyme concentrations E_i^{opt} in states of maximal steady-state flux can be determined by the

variational equation

$$\frac{\partial}{\partial E_i} \left(J - \lambda \left(\sum_{j=1}^r E_j - E_{\text{total}} \right) \right) = 0 \quad (11.14)$$

$$(i = 1, \dots, r),$$

where λ denotes the Lagrange multiplier. From this equation, it follows that

$$\frac{\partial J}{\partial E_i} = \lambda \quad (i = 1, \dots, r). \quad (11.15)$$

Equation (11.15) indicates that all nonnormalized flux-response coefficients with respect to enzyme concentrations (Chapter 4) have the same value. By multiplication with E_i^{opt}/J , we find

$$\frac{E_i^{\text{opt}}}{J} \left(\frac{\partial J}{\partial E_i} \right)_{E_j=E_j^{\text{opt}}} = \frac{E_i^{\text{opt}}}{J} \lambda. \quad (11.16)$$

The left-hand term of Eq. (11.16) represents the normalized flux control coefficient (Eq. (4.65)) $\left(C_{v_i}^J \right)_{E_j=E_j^{\text{opt}}} = C_i^{\text{opt}}$ of reaction i over steady-state flux J in optimal states. Since the sum of the flux control coefficients over all reactions equals unity (summation theorem, Eq. (4.70)), it follows that

$$1 = \sum_{i=1}^r \frac{E_i^{\text{opt}}}{J} \lambda = \frac{E_{\text{total}}}{J} \lambda. \quad (11.17)$$

Therefore,

$$C_i^{\text{opt}} = \frac{E_i^{\text{opt}}}{E_{\text{total}}}. \quad (11.18)$$

This means that the allocation of flux control coefficient in optimal states (here, states of maximal steady-state fluxes), C_i^{opt} , is equal to the allocation of the relative enzyme concentrations along the pathway: the flux only becomes maximal if enzymes with higher or lower control are present in higher or lower concentration, respectively.

The problem of maximizing the steady-state flux at a given total amount of enzyme is related to the problem of minimizing the total enzyme concentration that allows for a given steady-state flux. For an unbranched reaction pathway (Eq. (4.100)) obeying the flux equation 4.101, minimization of E_{total} results in the same optimal

allocation of relative enzyme concentrations and flux control coefficients as maximization of J .

The principle of minimizing the total enzyme concentration at fixed steady-state fluxes is more general since it may be applied also to branched reaction networks. Application of the principle of maximal steady-state flux to branched networks requires an objective function that balances the different fluxes by specific weights or could lead to conflicting interests between different fluxes in different branches.

Special conditions hold for the flux control coefficients in states of minimal total enzyme concentration at fixed steady-state fluxes. Since the reaction rates v_i are proportional to the enzyme concentrations, that is, $v_i = E_i \cdot f_i$, keeping fixed the steady-state fluxes $J_i^0 = v_i^0$ leads to the following relation between enzyme concentrations and substrate concentrations:

$$E_i = E_i(S_1, S_2, \dots, S_{r-1}) = \frac{v_i^0}{f_i}, \quad (11.21)$$

where the function f_i expresses the kinetic part of the reaction rate that is independent of the enzyme

Example 11.1

Consider the special case that every reaction of the pathway $S_0 \xrightleftharpoons{v_1} S_1 \xrightleftharpoons{v_2} \dots \xrightleftharpoons{v_r} S_r$ obeys mass action kinetics, that is, $v_i = E_i(k_i S_{i-1} - k_{-i} S_i)$ for $i = 1, \dots, r$, with the equilibrium constants $q_i = k_i/k_{-i}$. In this model, the steady-state flux reads $J = \left(S_0 \prod_{j=1}^r q_j - S_r \right) \cdot \left(\sum_{l=1}^r (E_l k_l)^{-1} \cdot \prod_{m=l}^r q_m \right)$ (Eq. (4.101)). Introducing this expression into Eq. (11.15) leads to

$$E_i^{\text{opt}} = E_{\text{total}} \cdot \sqrt{Y_i} \cdot \left(\sum_{l=1}^r \sqrt{Y_l} \right)^{-1} \quad \text{with} \quad (11.19)$$

$$Y_l = \frac{1}{k_l} \prod_{m=l}^r q_m.$$

For the flux control coefficients (compare with Eq. (4.65)) in states of maximal flux,

$$C_i^{\text{opt}} = \sqrt{Y_i} \cdot \left(\sum_{l=1}^r \sqrt{Y_l} \right)^{-1}. \quad (11.20)$$

The effect of optimization for a chain of four consecutive reactions is discussed in Example 4.7 and shown in Figure 4.10. The larger the deviation of the equilibrium constant q from 1, the stronger the effect of the optimization, that is, the larger the difference between maximal flux and reference flux.

concentration. The principle of minimal total enzyme concentration implies

$$\frac{\partial E_{\text{total}}}{\partial S_j} = \sum_{i=1}^r \frac{\partial E_i(S_1, \dots, S_{r-1})}{\partial S_j} - \sum_{i=1}^r \frac{v_i^0 f_i}{f_i^2} = 0, \quad (11.22)$$

which determines the metabolite concentrations in the optimal state. Since $f_i = v_i^0/E_i$, it follows that

$$\sum_{i=1}^r \frac{E_i^{\text{opt}}}{v_i^0} \frac{\partial v_i^0}{\partial S_j} = 0 \quad (11.23)$$

and in matrix representation

$$\left(\frac{dv}{dS} \right)^T (dg J)^{-1} E^{\text{opt}} = 0, \quad (11.24)$$

where E^{opt} is the vector containing the optimal enzyme concentrations. An expression for the flux control coefficients in matrix representation has been given in Eq. (4.98). Its transposed matrix reads

$$\begin{aligned} (C')^T &= I \\ &- (dg J) N^T \left(\left(N \frac{\partial v}{\partial S} \right)^{-1} \right)^T \left(\frac{\partial v}{\partial S} \right)^T (dg J)^{-1}. \end{aligned} \quad (11.25)$$

Postmultiplication with the vector E^{opt} and considering Eq. (11.24) leads to

$$(C')^T E^{\text{opt}} = E^{\text{opt}}. \quad (11.26)$$

This expression represents a functional relation between enzyme concentrations and flux control coefficients for enzymatic networks in states of minimal total enzyme concentration.

11.2.3 Temporal Transcription Programs

In this section, temporal adaptation of enzyme concentration is studied, instead of steady-state solutions as in the previous section. Consider an unbranched metabolic pathway that can be switched on or off by the cell depending on actual requirements. The product S_r of the pathway is desirable, but not essential for the reproduction of the cell. The faster the initial substrate S_0 can be converted into S_r , the more efficiently the

cell may reproduce and outcompete other individuals. If S_0 is available, then the cell produces the enzymes of the pathway to make use of the substrate. If the substrate is not available, then the cell does not synthesize the respective enzymes for economic reasons. Bacterial amino acid synthesis is frequently organized in this way. This scenario has been studied theoretically [73] by starting with a resting pathway; that is, although the genes for the enzymes are present, they are not expressed due to lack of the substrate. Suddenly S_0 appears in the environment (by feeding or change of place). How can the cell make as soon as possible use of S_0 and convert it into S_r ?

The system of ordinary differential equations (ODEs) describing the dynamics of the pathway reads

$$\begin{aligned} \frac{dS_0}{dt} &= -k_1 \cdot E_1 \cdot S_0, \\ \frac{dS_i}{dt} &= k_i \cdot E_i \cdot S_{i-1} - k_{i+1} \cdot E_{i+1} \cdot S_i, \\ \frac{dS_r}{dt} &= k_r \cdot E_r \cdot S_{r-1} \quad (i = 1, \dots, r-1). \end{aligned} \quad (11.27)$$

For the sake of simplicity, we first assume that the cell can make the enzymes instantaneously when necessary (neglecting the time necessary for transcription and translation), but the total amount of enzyme is limited due to limited capacity of the cell to produce and store proteins. The time necessary to produce S_r from S_0 is measured by the transition time

$$\tau = \frac{1}{S_0(0)} \int_{t=0}^{\infty} (S_0(0) - S_r(t)) dt. \quad (11.28)$$

The optimization problem to be solved is to find a temporal profile of enzyme concentrations that minimizes the transition time ($\tau = \min$) at fixed value of $E_{\text{total}} = \sum_{i=1}^r E_i(t) = \text{constant}$.

To solve the above optimization problem for pathways with n enzymes, we divide the time axis into m intervals in each of which the enzyme concentrations are constant. The quantities to be optimized are the switching times T_1, T_2, \dots defining the time intervals and the enzyme concentrations during these intervals. In the reference case with only one interval ($m = 0$ switches), the optimal enzyme concentrations are all equal: $E_i = E_{\text{total}}/r$ ($i = 1, \dots, r$). The transition time for this case reads $\tau = r^2$ in units of $(k \cdot E_{\text{total}})^{-1}$. Permitting one switch ($m = 1$) between intervals of constant enzyme concentrations allows for a considerably lower transition time. An increase in the number of possible switches

Example 11.2

For a pathway consisting of only $r = 2$ reactions with $k_i = k$ ($i = 1, \dots, r$), there is an explicit solution of the above-stated problem [73]. The optimal enzyme profile consists of two phases and an abrupt switch at time T_1 . In the first interval $0 \leq t \leq T_1$, only the first enzyme is present, that is, $E_1 = E_{\text{total}}$ and $E_2 = 0$. The switching time is $T_1 = \ln(2/(3 - \sqrt{5}))$. In the second interval $T_1 < t < \infty$, both enzymes are present with constant concentrations $E_1 = E_{\text{total}} \cdot (3 - \sqrt{5})/2$ and $E_2 = E_{\text{total}} \cdot (\sqrt{5} - 1)/2$. Note, for curiosity, that in this case the ratio E_2/E_1 equals the golden ratio (i.e., $(1 + \sqrt{5})/2 : 1$). The minimal transition time for these optimal concentrations is $\tau^{\min} = 1 + T_1 + (1 - e^{-T_1})^{-1} \approx 3.58$ in units of $(k \cdot E_{\text{total}})^{-1}$. This means that in the first phase all available enzyme is used to catabolize the initial substrate; product is made only in the second phase. The fastest possible conversion of S_0 to S_2 employs a delayed onset in the formation of S_2 that favors an accelerated decay of S_0 in the initial phase and pays off in the second phase. The temporal profiles of enzyme and metabolite concentrations are shown in Figure 11.8.

$(m > 1)$ leads to a decrease in the transition time until the number of switches reaches $m = r - 1$. The corresponding optimal enzyme profiles have the following characteristics: within any time interval, except for the last one, only a single enzyme is fully active whereas all others are shut off. At the beginning of the process, the whole amount of available protein is allocated exclusively to the first enzyme of the chain. Each of the following switches turns off the active enzyme and allocates the total amount of protein to the enzyme that catalyzes the following reaction. The last switch allocates finite fractions of the protein amount to all enzymes with increasing amounts from the first one to the last one (Figure 11.9).

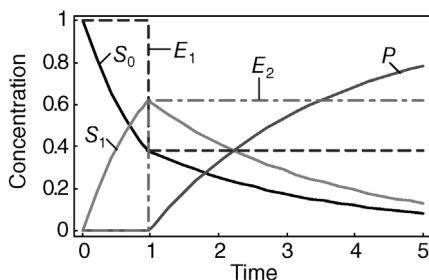


Figure 11.8 Optimal enzyme and metabolite concentration time profiles for a linear metabolic pathway as explained in Example 9.2. Parameters: $S_0(0) = 1$, $S_1(0) = S_2(0) = 0$, $E_{\text{total}} = 1$, and $k = 1$.

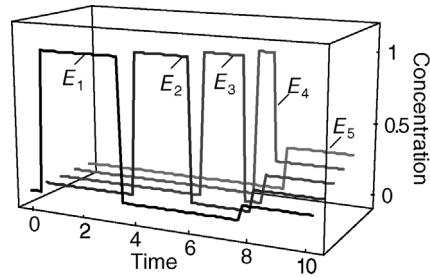


Figure 11.9 Optimal temporal enzyme profiles yielding the minimal transition time for a pathway of five reactions. The switching times are $T_1 = 3.08$, $T_2 = 5.28$, $T_3 = 6.77$, and $T_4 = 7.58$.

If one compares the case of no switch (the reference case) with the case of $m = r - 1$ switches, the drop in transition time (gain in turnover speed) amplifies with increasing length r of the reaction chain.

If we impose a weaker condition for the conversion of substrate S_0 into product S_r by demanding only conversion of 90% instead of 100%, then the optimal solution looks as follows: enzymes E_1 to E_{r-1} are switched on and off successively in the same way as in the case of 100% conversion except for the last interval, where enzyme E_r is fully activated, but none of the other enzymes (Bartl, 2008). The transition time is considerably reduced compared with the strict case. In conclusion, abandonment of perfection (here demanding 90% metabolite conversion instead of 100%) may lead to incomplete, but faster metabolite conversion.

When cells have to survive in a complicated environment, it can be assumed that they should optimize several properties simultaneously, for example, to minimize the time needed for proper response to an external stimulus or to minimize the amount of enzyme necessary for the respective metabolic conversion or to minimize the amount of accumulated intermediates. This multi-objective case has been studied extensively [74]. It leads to multiple optimal solutions, that is, solutions that cannot be improved in one property without impairing another property. This is called a Pareto front. Figure 11.10 shows an example for a linear pathway with three reactions in a row converting substrate S_1 into product S_4 . The objectives are to minimize the accumulation of S_2 and S_3 ($\int_0^\infty (S_2 + S_3) dt \rightarrow \min$) and to minimize the time needed to reach a certain amount of S_4 .

The simple example of a metabolic pathway shows that temporal adjustment of enzyme activities, for instance, by regulation of gene expression, may lead to a considerable improvement of metabolic efficiency. As detailed in Example 11.3, bacterial amino acid production pathways are possibly regulated in the described manner.

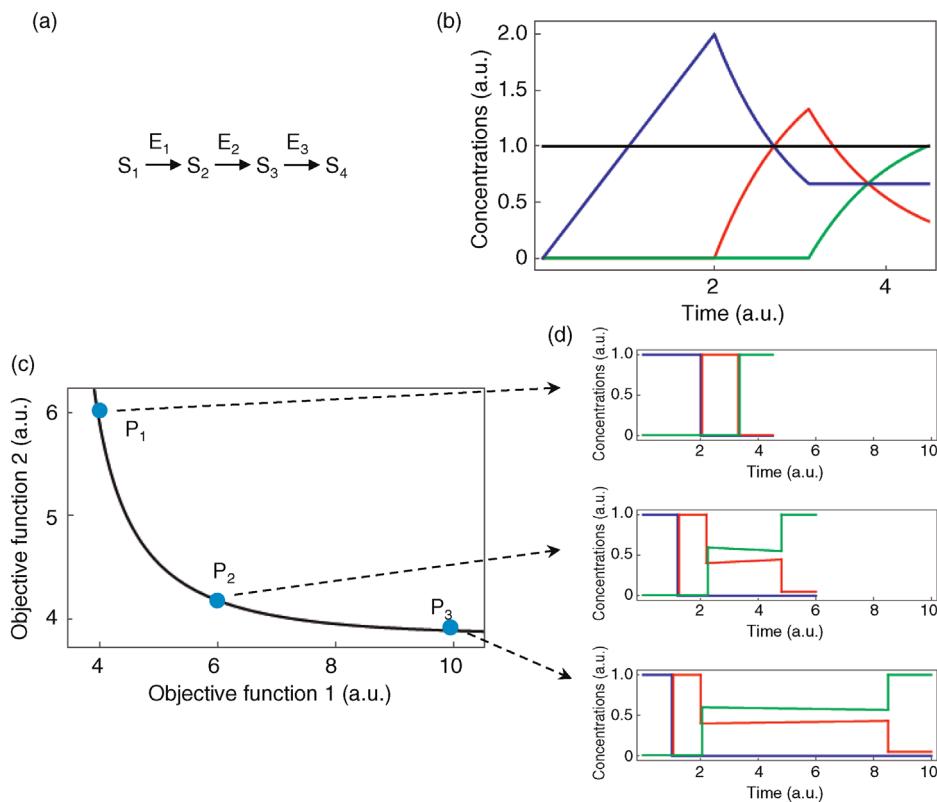


Figure 11.10 Multiple objective functions. (a) Linear pathway of three consecutive reactions. The objective functions are the minimization of intermediates (S_2 and S_3) over time and the minimization of the time needed for substrate S_4 to reach a certain amount. (b) Metabolite dynamics in the optimal case P_1 . (c) Pareto front obtained in the minimization process. At each point, one property can only be improved at the cost of the other property. For different points, the optimal enzyme profiles are represented in (d).

11.3 Evolution and Self-Organization

Summary

How did life arise? How did regulatory circles arise that we study nowadays with sophisticated experimental techniques? Why did some species survive while others got extinct? In the laboratory, we may observe evolutionary processes in quickly reproducing species, for example, the adaptation of *E. coli* to certain nutritional conditions [76] and in the wild we may study speciation, like Darwin did for the birds on the Galapagos Islands. However, these observations don't explain the emergence of evolution and self-organization. Here, we introduce you to early concepts of evolution on the molecular level, which are based on self-replication, mutation of genetic material, and selection of successful species. Starting from selection equations for single species with and without constraints, we continue with the quasispecies concept

allowing for errors in production. The hypercycle concept describes the interaction of several species for successful replication. Spin glass models and neutral evolution are further concepts dealing with the phenomenon of self-organization of replicating species.

11.3.1 Introduction

Since Darwin's famous book "On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life" (1859) [77], it is a widely accepted view that biological species have gradually developed from a few common ancestors in an iterative process of mutations and natural selection during millions of years. Throughout this evolution, new species appeared and existing species adapted themselves to changing environmental conditions; others became extinct.

Example 11.3

Zaslaver *et al.* experimentally investigated amino acid biosynthesis systems of *E. coli*. They identified the temporal expression pattern and showed a hierarchy of expression that matches the enzyme order in the unbranched pathways [75]. They included the time requirements and the costs for enzyme production in their model. For a system of three enzymes, the set of equations governing the temporal response to changes in metabolite availability is

$$\begin{aligned} \frac{dS_i}{dt} &= \frac{V_{\max,i} \cdot E_i \cdot S_{i-1}}{K_{m,i} + S_{i-1}} - \frac{V_{\max,i+1} \cdot E_{i+1} \cdot S_i}{K_{m,i+1} + S_i} - \alpha \cdot S_i, \quad i = 1, \dots, 3, \\ \frac{dE_i}{dt} &= \frac{\beta_i \cdot k_i}{k_i + R(t)} - \alpha \cdot E_i, \quad i = 1, \dots, 3, \\ \frac{dR_T}{dt} &= \frac{\beta_0 \cdot k_0}{k_0 + R(t)} - \alpha \cdot R_T \quad \text{with} \quad R(t) = \frac{R_T \cdot P(t)}{K_R + P(t)}, \end{aligned} \quad (11.29)$$

$$\begin{aligned} C &= a \cdot \sum_i \int_0^T \frac{\beta_i \cdot k_i}{k_i + R(t)} dt + \int_0^T |F - F_{\text{goal}}| dt \quad \text{with} \\ F &= \frac{V_{\max,3} \cdot E_3 \cdot S_2}{K_{m,3} + S_2}, \end{aligned} \quad (11.30)$$

where S_i and E_i are the concentrations of metabolites and enzymes, respectively, $V_{\max,i}$ are the maximal rates, and $K_{m,i}$ are the Michaelis constants. The parameters β_i denote maximal promoter activity of gene i , and k_i are the repression coefficients, that is, the concentration of repressor needed for 50% repression of gene i . R_T is the total repressor concentration, K_R is the dissociation constant, and $R(t)$ is the active repressor level. The cost function (11.30) is composed of two terms, that is, the cost of producing the enzymes and the rate and precision at which F approaches its goal F_{goal} . a is a weight factor and T is the typical time scale of the activation of the system.

Minimization of the cost function by optimizing β_i , k_i , and K_R results in solutions with $\beta_1 > \beta_2 > \beta_3$, representing a hierarchy in the maximal promoter activity, and $k_1 < k_2 < k_3$, representing feedback strength by the repressor that is stronger the earlier the enzyme in the pathway. The time courses of the enzymes are shown in Figure 11.11.

Note that in Example 11.2 enzyme profiles have been optimized without asking how these profiles can be ensured by the cellular machinery, while in Example 11.3 the parameters of a model for the enzyme production machinery were optimized.

Mutations are changes in the genetic material (*genotype*) of organisms. They usually cause changes of properties of the organisms (*phenotype*). They occur by chance. *Natural selection* proves fitness with respect to survival and reproduction in the actual environment with no further goal or plan. The fittest in gaining and using the necessary resources will win and survive, while the others become extinct. The term natural selection has to be distinguished from *artificial selection*. Artificial selection chooses specific features to be retained or eliminated depending on a goal or intention (e.g., the objective of a farmer to improve the growth of corn such that it permits maximal harvest).

The view that biological systems developed during evolution can be applied not only to species, but also to other units of biological consideration, such as cells, metabolic pathways, and gene expression networks. It has been questioned though whether the principle of development by mutation and selection can be used to understand evolution in a theoretical way, to learn how and why biological systems assumed their current state, and to predict structures of biological networks using simple analogies.

A basic assumption of theoretical considerations is that evolution is based on the *trial-and-error* process of variation and natural selection of systems at all levels of complexity. The development of biological systems further involves the feature of *self-organization*, that is, assuming stable structures that (i) employ a global cooperation between the elements, (ii) are inherent in the system, and (iii) are contained independently of external pressure.

Biological evolution is a quite complex process. Like for other subjects of systems biology, this complexity conflicts with the attempt of developing general models for evolution. Biological diversity increases the number of components to be considered in models. Therefore, it seems unrealistic to develop a detailed and fundamental description of phenomena, as it is sometimes possible in theoretical physics. In general, evolutionary models can rarely be experimentally tested, since we will not survive the necessary time span. Nevertheless, the step has been undertaken to clarify with mathematical modeling features of biological phenomena such as competition and cooperativity, self-organization, or emergence of new species. Such models provide a better understanding of biological evolution; they also give generalized descriptions of biological experiments. The following types of evolutionary models have been developed.

Models of the origin of self-replicating systems have been constructed in connection with the origin of life problem. Manfred Eigen and Peter Schuster [78–82] introduced the concept of quasispecies and hypercycles. These models describe mathematically some hypothetical evolutionary stages of prebiological self-reproducing

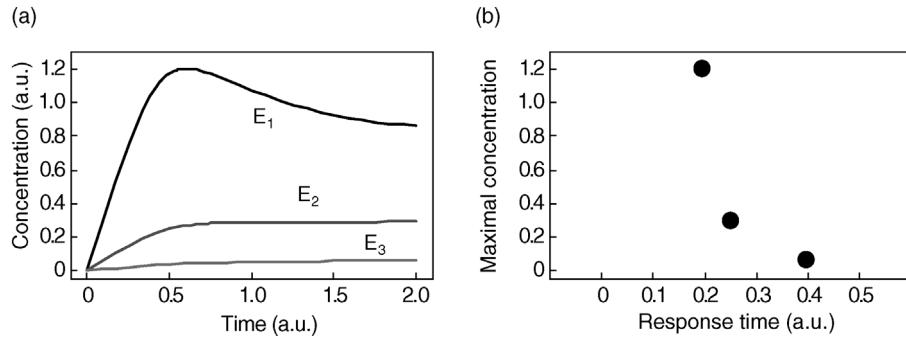


Figure 11.11 Simulation of equation system (11.39) with $V_{\max,i} = K_{m,i} = \alpha = 1$, $\beta_0 = 5$, $\beta_1 = 3.4$, $\beta_2 = 0.67$, $\beta_3 = 0.1$, $k_0 = 0.1$, $k_1 = 0.23$, $k_2 = 0.68$, $k_3 = 2.1$, and $K_R = 0.0001$. (a) Time course of enzyme concentrations. (b) Maximal response of enzymes versus response time, that is, time when enzyme concentration is half-maximal.

macromolecular systems. The *quasispecies* model is a description of the process of the Darwinian evolution of self-replicating entities within the framework of physical chemistry. It is useful mainly in providing a qualitative understanding of the evolutionary processes of self-replicating macromolecules such as RNA or DNA or simple asexual organisms such as bacteria or viruses. Quantitative predictions based on this model are difficult because the parameters that serve as its input are hard to obtain from actual biological systems.

The model relies on four assumptions:

- i) The self-replicating entities can be represented as sequences composed of letters from an alphabet, for example, sequences of DNA consisting of the four bases A, C, G, and T.
- ii) New sequences enter the system solely as either correct or erroneous copies of other sequences that are already present.
- iii) The substrates, or raw materials, necessary for ongoing replication are always present in sufficient quantity in the considered volume. Excess sequences are washed away in an outgoing flux.
- iv) Sequences may decay into their building blocks.

In the quasispecies model, mutations occur by errors made during copying of already existing sequences. Further, selection arises because different types of sequences tend to replicate at different rates, which leads to the suppression of sequences that replicate more slowly in favor of sequences that replicate faster. In the following chapters, we will show how to put these assumptions into equations. Note that these models still disregard some aspects of evolution, which nowadays are also considered important, such as recombination, that is, the exchange of parts of sequences between DNA molecules during meiosis, or the transfer of sequences among species such as from a virus to its host.

General models of evolution describe some informational and cybernetic aspects of evolution such as the neutral evolution theory by Motoo Kimura [83–86] and Stuart Kauffman’s automata [87–90]. Models of artificial life are aimed at understanding the formal laws of life and evolution. These models analyze the evolution of artificial “organisms,” living in computer program worlds.

Computer algorithms have been developed, which use evolutionary methods of optimization to solve practical problems. The genetic algorithm by J.H. Holland [91–93] and the evolutionary programming initiated by Lawrence Fogel *et al.* [94], evolution strategies by Ingo Rechenberg [95], and genetic programming propagated by J. Koza [96] are well-known examples of these researches. Evolutionary optimization has been applied to models of biological systems. The idea is to predict features of a biological system from the requirement that it functions optimally in order to be the fittest that survives.

These models are usually very abstract and much simpler than biological processes. But the abstractness is necessary to find a general representation of the investigated features despite their real complexity and diversity. Furthermore, the concepts must be simple enough to be perceived and to be applicable.

11.3.2 Selection Equations for Biological Macromolecules

The dynamics of selection processes in early evolution may be described with equations of the type used in population dynamics. Instead of species, we consider here as model the macromolecules that are able to self-replicate. For example, DNA molecules are replicated by complementary base pairing (A = T and C ≡ G):



A population is defined as a set $\{S_1, S_2, \dots, S_n\}$ of n DNA sequences. Each sequence is a string of N symbols, s_{ik} , with $k = 1, \dots, N$ and $i = 1, \dots, n$. The symbols are taken from an alphabet containing λ letters. For the DNA as example, we have a four-letter alphabet ($\lambda = 4$, $s_{ik} = A, C, G, T$). Thus, the space of possible sequences covers λ^N different sequences. The sequence length N and the population size n are assumed to be large: $N, n \gg 1$. The concentration of molecules with identical sequences S_i is x_i . We assume that the DNA molecules have some selective value f_i that depends on the sequence of nucleotides. During propagation of the replication process, the molecules with higher selective value will have an advantage compared with those with a lower value. The evolution character depends strongly on the population size n . If n is very large ($n \gg \lambda^N$), the number of all sequences in a population is large and evolution can be considered as a deterministic process. In this case, the population dynamics can be described in terms of the ordinary differential equations. In the opposite case ($n \ll \lambda^N$), the evolution process should be handled as stochastic (not done here).

A basic assumption of this concept is that there is a master sequence, S_m , having the maximal selective value f_m . The selective value f_i of any other sequence S_i depends on the *Hamming distance* h (the number of different symbols at corresponding places in sequences) between S_i and the master sequence S_m : $f_i = f_m(h(S_i, S_m))$ – the smaller the distance h , the greater the selective value f_i .

In the following, we present a number of scenarios for replication of sequences with or without additional constraints and characterize the resulting population dynamics.

Example 11.4

Consider a “soup” containing a million molecules. For a sequence of length $N = 5$ from an alphabet with $\lambda = 4$ letters, there are $4^5 = 1024$ possibilities for different sequences S_i and the mean abundance per sequence is about a thousand. The number of possibilities in the case of a string length $N = 20$ is about 10^{12} , meaning that on average only one of a million possible sequences is present in the soup. In the latter case, a mutation would (most probably) result in a new sequence that was not present before. In the first case, a mutation of one molecule would result in the increase of the abundance of another, already present, sequence.

11.3.2.1 Self-Replication without Interactions

Assume that DNA molecules perform identical replication and are also subject of decay. The time course of their concentration x_i is determined by the following differential equation:

$$\frac{dx_i}{dt} = a_i x_i - d_i x_i = (a_i - d_i)x_i, \quad (11.32)$$

where a_i and d_i are the rate constants for replication and decay, respectively. For constant values of a_i and d_i and an initial concentration x_i^0 for $t = 0$, the solution of Eq. (11.32) is given by

$$x_i(t) = x_i^0 e^{(a_i - d_i)t}. \quad (11.33)$$

Depending on the difference $f_i = a_i - d_i$, Eq. (11.33) describes a monotonous increase ($a_i > d_i$), decrease ($a_i < d_i$), or a constant concentration ($a_i = d_i$). Therefore, the difference f_i can be considered as the selective value (also called *excess productivity*).

11.3.2.2 Selection at Constant Total Concentration of Self-Reproducing Molecules

In Eq. (11.33), the dynamics of species S_i is independent of the behavior of the other species. However, selection will only happen if there is interaction between the species, leading to selective pressure. This is given if, for example, the number of building blocks (mononucleotides) is limited or if the concentration may not exceed a certain maximum. In the latter case, the total concentration of species can be kept constant by introducing a term describing elimination of supernumerary individuals or dilution by flow out of the considered volume. This is called selection at *constant organization* [97]. Assuming that the dilution rate is proportional to the actual concentration of the species, it follows that

$$\frac{dx_i}{dt} = f_i \cdot x_i - \phi \cdot x_i, \quad (11.34)$$

where again $f_i = a_i - d_i$. Under the condition of constant total concentration ($\sum_i x_i = x_{\text{total}} = \text{constant}$ or $\sum_i dx_i / dt = 0$), it follows that

$$\phi = \frac{\sum_i f_i x_i}{\sum_k x_k} = \bar{f}. \quad (11.35)$$

Since f_i denotes the excess productivity of species S_i , $\phi = \bar{f}(x)$ is the mean excess productivity. Introducing Eq. (11.35) into (11.34) yields the *selection equations*

$$\frac{dx_i}{dt} = (f_i - \bar{f})x_i. \quad (11.36)$$

This is a system of coupled nonlinear ODEs. Nevertheless, it is easy to see that the concentrations increase over

Example 11.5

Consider $n = 4$ competing sequences with equal initial concentrations $x_i^0 = 1/n = 1/4$ and different selective values $f_1 < f_2 < f_3 < f_4$. This results in the temporal behavior depicted in Figure 11.12.

For $t = 0$, $f_1 < \bar{f} < f_2 < f_3 < f_4$; hence, the concentration x_1 decreases, while the other concentrations increase. With time progression, f_2 and f_3 fall one after the other under the rising threshold \bar{f} , and the respective concentrations x_2 and x_3 eventually decrease. Concentration x_4 always moves up until it reaches asymptotically the total concentration. Species 4 is the master species, since it has the highest selective value.

time for all species with $f_i > \bar{f}$ and that the concentrations decrease for all species with $f_i < \bar{f}$. The mean excess productivity \bar{f} is, therefore, a *selection threshold*. Equation (11.36) shows that the concentrations of species with high selective value increase with time. Hence, also the mean excess productivity increases according to Eq. (11.35). Successively, the selective values of more and more species become lower than \bar{f} , their concentrations start to decrease, and eventually they die out. Finally, only the species with the highest initial selective value will survive, that is, the so-called *master species*.

The explicit solution of the equation system (11.36) reads

$$x_i(t) = \frac{x_{\text{total}} \cdot x_i^0 e^{f_i t}}{\sum_j x_j^0 e^{f_j t}}. \quad (11.37)$$

The species with the highest excess productivity is the *master sequence* S_m . In the current scenario, it will survive and the other species will become extinct.

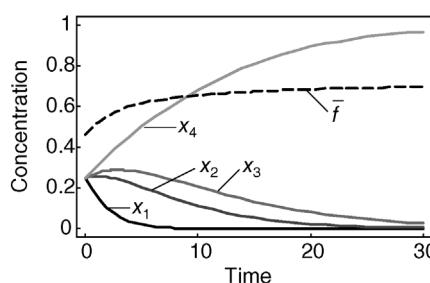


Figure 11.12 Time course of concentration of four competing sequences performing identical replication as described in Eq. (11.37). Initial concentrations are $x_i^0 = n/4$; the selective values are $f_1 = 0.01$, $f_2 = 0.52$, $f_3 = 0.58$, and $f_4 = 0.7$. The dashed line shows the mean excess productivity \bar{f} that increases with time.

11.3.3

The Quasispecies Model: Self-Replication with Mutations

Up to now, we considered only the case of identical self-replication. However, mutations are an important feature of evolution. In the case of erroneous replication of sequence S_i , a molecule of sequence S_j is produced that also belongs to the space of possible sequences. The right-hand side of Eq. (11.34) can be extended in the following way:

$$\frac{dx_i}{dt} = a_i q_i x_i - d_i x_i + \sum_{j \neq i} m_{ij} x_j - \phi x_i. \quad (11.38)$$

The expression $a_i q_i x_i$ denotes the rate of identical self-replication, where $q_i \leq 1$ is the ratio of correctly replicated sequences. The quantity q_i characterizes the quality of replication. The term $d_i x_i$ denotes again the decay rate of sequence S_i , and ϕx_i is the dilution term ensuring constant total concentration. The expression $\sum_{j \neq i} m_{ij} x_j$ characterizes the synthesis rate of sequence S_i from other sequences S_j by mutation.

Since every replication results in the production of a sequence from the possible sequence space, the rate of erroneous replication of all sequences must be equal to the synthesis rate of sequences by mutation. Therefore,

$$\sum_i a_i (1 - q_i) x_i = \sum_i \sum_{j \neq i} m_{ij} x_j. \quad (11.39)$$

Taking again into account the constant total concentration and Eq. (11.39) yields

$$0 = \sum_i \frac{dx_i}{dt} = \sum_i a_i x_i - \sum_i d_i x_i - \phi \sum_i x_i. \quad (11.40)$$

This way, Eq. (11.35) again holds for ϕ . The selection equation for self-replication with mutation reads

$$\frac{dx_i}{dt} = (a_i q_i - d_i - \bar{f}) x_i + \sum_{j \neq i} m_{ij} x_j. \quad (11.41)$$

Equation (11.41) differs from Eq. (11.36) by an additional coupling term that is due to the mutations. A high precision of replication requires small coupling constants m_{ij} . For small m_{ij} , one may expect similar behavior as in the case without mutation: the sequences with the high selective value will accumulate, whereas sequences with low selective value die out. But the existing sequences always produce erroneous sequences that are closely related to them and differ only in a small number of mutations. A species and its close relatives that appeared by mutation are referred to as *quasispecies*. Therefore, there is not

selection of a single master species, but of a set of species. The species with the highest selective value and its close relatives form the *master quasispecies distribution*.

In conclusion, the quasispecies model does not predict the ultimate extinction of all but the fastest replicating sequence. Although the sequences that replicate more slowly cannot sustain their abundance level by themselves, they are constantly replenished as sequences that replicate faster mutate into them. At equilibrium, removal of slowly replicating sequences due to decay or outflow is balanced by replenishing, so that even relatively slowly replicating sequences can remain present in finite abundance.

Due to the ongoing production of mutant sequences, selection does not act on single sequences, but on so-called *mutational clouds* of closely related sequences, the *quasispecies*. In other words, the evolutionary success of a particular sequence depends not only on its own replication rate, but also on the replication rates of the mutant sequences it produces, and on the replication rates of the sequences of which it is a mutant. As a consequence, the sequence that replicates fastest may even disappear completely in selection–mutation equilibrium, in favor of more slowly replicating sequences that are part of a quasispecies with a higher average growth rate [98]. Mutational clouds as predicted by the quasispecies model have been observed in RNA viruses and in *in vitro* RNA replication [92,99].

11.3.3.1 The Genetic Algorithm

The evolution process in the quasispecies concept can also be viewed as a stochastic algorithm. This has been widely used as a strategy for a computer search algorithm. The evolution process produces consequent generations. We start with an initial population $\mathbf{S}(0) = \{S_1(0), \dots, S_n(0)\}$ at time $t = 0$. A new generation $\mathbf{S}(t+1)$ is obtained from the old one $\mathbf{S}(t)$ by random selection and mutation of sequences $S_i(t)$, where t corresponds to the generation number. Assume that all $f_i \leq 1$, which can be ensured by normalization. The model evolution process can be described formally in the following computer program-like manner.

- i) *Initialization.* Form an initial population $\mathbf{S}(0) = \{S_1(0), \dots, S_n(0)\}$ by choosing for every sequence $i = 1, \dots, n$ and for every position $k = 1, \dots, N$ in the sequence randomly a symbol from the given alphabet (e.g., A, T, C, G, or “0” and “1”).
- ii) *Sequence selection for the new generation.* Select sequences by choosing randomly numbers i' with the probability $f_{i'}$ and adding a copy of the old sequence $S_{i'}(t)$ to the new population as $S_{i'}(t+1)$.

- iii) *Control of population size.* Repeat (ii) until the new population has reached size n of the initial population.
- iv) *Mutation.* Change with a certain probability for every sequence $S_i(t)$ with $i = 1, \dots, n$ and for every position $s_{ik}(t)$ with $k = 1, \dots, N$ the current symbol to another symbol of the alphabet.
- v) *Recombination.* As a possible extension to the previous steps, dice a number k' with $1 \leq k' \leq N$ and two numbers i' and i'' with $1 \leq i', i'' \leq n$ and then exchange positions $1, \dots, k'$ between sequences $s_{i'}$ and $s_{i''}$.
- vi) *Iteration.* Repeat steps (ii) to (iv) or (v) for successive time points.

The application of this algorithm allows for a visualization of the emergence of quasispecies and the master quasispecies distribution from an initial set of sequences during time.

11.3.3.2 Assessment of Sequence Length for Stable Passing On of Sequence Information

The number of possible mutants of a certain sequence depends on its length N . Since the alphabet for DNA molecules has four letters, each sequence has $3N$ neighboring sequences (mutants) with Hamming distance $h = 1$. For mutants with arbitrary distance h , there are $N_h = 3^h \binom{N}{h}$ possibilities. Therefore, the number of sequences belonging to a quasispecies can be very high.

The quality measure q entering Eq. (11.38) for a sequence of length N can be expressed by the probability p_q of correct replication of the individual nucleotides. Assuming that this probability is independent of position and type of the nucleotide, the quality measure reads

$$q = p_q^N. \quad (11.42)$$

Mathematical investigation confirms that stable passing on of information is only possible if the value of the quality measure is above a certain threshold $q > 1/s$. The parameter s is the relative growth rate of the master sequence, which is referred to as superiority. The generation of new species, and therefore development during evolution, is only possible with mutations. But too large mutation rates lead to destruction of already accumulated information. The closer q is to 1, the longer sequences may replicate in a stable manner. For the maximal length of a sequence, Eigen and coworkers [100] determined the relation

$$N \left(1 - p_q\right) < \ln s. \quad (11.43)$$

During evolution, self-reproducing systems became more and more complex and the respective sequences

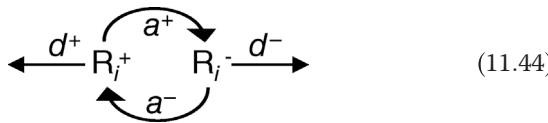
became longer. Hence, the accuracy of replication also had to increase. Since the number of nucleotides in mammalian cells is about $N \approx 10^9$, Eq. (11.43) implies that the error rate per nucleotide is on the order of 10^{-9} or lower. This is in good agreement with the accuracy of the replication in mammalian cells. Such a level of accuracy cannot be reached by simple self-replication based on chemical base pairing; it necessitates help from polymerases that catalyze the replication. For uncatalyzed replication of nucleic acids, the error rate is at least 10^{-2} , which implies a maximal sequence length of $N \approx 10^2$.

11.3.3.3 Coexistence of Self-Replicating Sequences: Complementary Replication of RNA

As we saw, only mutation and selection cannot explain the complexity of currently existing sequences and replication mechanisms. Their stable existence necessitates cooperation between different types of molecules besides their competition.

Consider the replication of RNA molecules, a mechanism used by RNA phages. The RNA replicates by complementary base pairing. There are two complementary strands, R_i^+ and R_i^- . The synthesis of one strand always requires the presence of the complementary strand, that is, R_i^+ derives from R_i^- and vice versa. Thus, both strands have to cooperate for replication.

For a single pair of complementary strands, we have the following cartoon:



with the kinetic constants a^+ and a^- for replication and the kinetic constants d^+ and d^- for degradation. Denoting the concentrations of R_i^+ and R_i^- with x^+ and x^- , respectively, the corresponding ODE system reads

$$\begin{aligned}\frac{dx^+}{dt} &= a^-x^- - d^+x^+, \\ \frac{dx^-}{dt} &= a^+x^+ - d^-x^-,\end{aligned}\quad (11.45)$$

and in matrix notation

$$\frac{d}{dt} \begin{pmatrix} x^+ \\ x^- \end{pmatrix} = \begin{pmatrix} -d^+ & a^- \\ a^+ & -d^- \end{pmatrix} \begin{pmatrix} x^+ \\ x^- \end{pmatrix}. \quad (11.46)$$

The eigenvalues of the Jacobian matrix (see Chapter 15) in Eq. (11.46) are

$$\lambda_{1/2} = -\frac{d^+ + d^-}{2} \pm \sqrt{\left(\frac{d^+ - d^-}{2}\right)^2 + a^+a^-}. \quad (11.47)$$

They are always real, since the kinetic constants have nonnegative values. While λ_2 (the “−” solution) is always

Example 11.6

For the simple case where both strands have the same kinetic properties ($a^+ = a^- = a$ and $d^+ = d^- = d$), the temporal behavior is shown in Figure 11.13.

If $a > d$, then the exponential increase dominates and both strands accumulate. If $a < d$, then both strands become extinct. In both cases, the initial concentration differences eventually become negligible and both strands behave identically.

negative, λ_1 (the “+” solution) may assume positive or negative values depending on the parameters:

$$\begin{aligned}\lambda_1 < 0 &\text{ for } a^+a^- < d^+d^-, \\ \lambda_1 > 0 &\text{ for } a^+a^- > d^+d^-.\end{aligned}\quad (11.48)$$

The solution of Eq. (11.46) reads

$$\mathbf{x}(t) = \mathbf{b}^{(1)} e^{\lambda_1 t} + \mathbf{b}^{(2)} e^{\lambda_2 t} \quad (11.49)$$

with concentration vector $\mathbf{x} = (x^+, x^-)^T$ and eigenvectors $\mathbf{b}^{(1)}$ and $\mathbf{b}^{(2)}$ of the Jacobian.

According to Eq. (11.49), a negative eigenvalue λ_1 indicates the extinction of both strands R_i^+ and R_i^- . For a positive eigenvalue λ_1 , the right-hand side of Eq. (11.49) comprises an exponentially increasing and an exponentially decreasing term, the first of which dominates for progressing time. Thus, the concentration of both strands rises exponentially. Furthermore, the growth of each strand depends not only on its own kinetic constants but also on the kinetic constants of the other strand.

11.3.4

The Hypercycle

The *hypercycle* model describes a self-reproducing macromolecular system, in which RNAs and enzymes cooperate in the following manner. There are n RNA species;

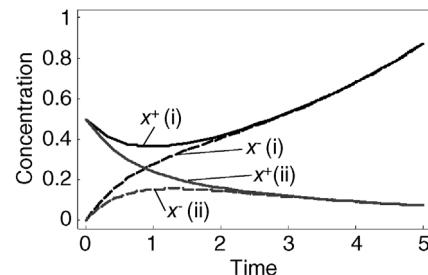
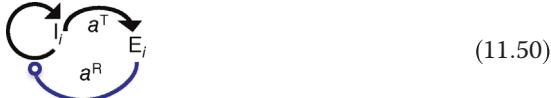


Figure 11.13 Concentration time courses for complementary replication of RNA molecules according to Eq. (11.45). Both strands have the same kinetic properties. Parameters in case (i) are $a = 1$ and $d = 0.75$ (black lines, solid: + strand, dashed: − strand); both strands accumulate. In case (ii), $a = 0.75$ and $d = 1$ (gray lines, solid: + strand, dashed: − strand); both strands become extinct with time.

the i th RNA codes for the i th enzyme ($i = 1, 2, \dots, n$). The enzymes cyclically increase the replication rates of the RNAs, that is, the first enzyme increases replication rate of the second RNA, the second enzyme increases replication rate of the third RNA, \dots , and eventually the n th enzyme increases replication rate of the first RNA. In addition, the described system possesses primitive translation abilities, in that the information stored in RNA sequences is translated into enzymes, analogously to the usual translation processes in contemporary cells. M. Eigen and P. Schuster consider hypercycles as predecessors of protocells (primitive unicellular biological organisms). The action of the enzymes accelerates the replication of the RNA species and enhances the accuracy. Although the hypercycle concept may explain the advantages of the cooperation of RNA and enzymes, it cannot explain how it aroused during evolution. A special problem in this regard is the emergence of a genetic code.

The simplest cooperation between enzymes and nucleotide sequences is given when (i) the enzyme E_i is the translation product of the nucleic acid I_i and (ii) the enzyme E_i catalyzes primarily the identical replication of I_i as depicted below.



a^T and a^R are the kinetic constants of translation and replication, respectively.

In general, a hypercycle involves several enzymes and RNA molecules. In addition, the mentioned macromolecules cooperate to provide primitive translation abilities; thus, the information encoded in RNA sequences is translated into enzymes, analogously to the usual translation processes in biological objects. The cyclic organization of the hypercycle shown in Figure 11.14 ensures its structural stability.

In the following, we will consider the dynamics for the competition of two hypercycles of the type depicted in Eq. (11.50). Under the condition of constant total concentration and negligible decay of compounds, the ODE system for RNAs with concentrations I_1 and I_2 and enzymes E_1 and E_2 reads

$$\begin{aligned} \frac{dE_1}{dt} &= a_1^T I_1 - \phi_E E_1, \\ \frac{dI_1}{dt} &= a_1^R I_1 E_1 - \phi_I I_1, \\ \frac{dE_2}{dt} &= a_2^T I_2 - \phi_E E_2, \\ \frac{dI_2}{dt} &= a_2^R I_2 E_2 - \phi_I I_2. \end{aligned} \quad (11.51)$$

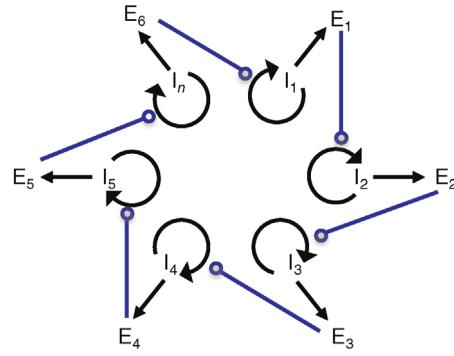


Figure 11.14 Schematic representation of the hypercycle consisting of RNA molecules I_i and enzymes E_i ($i = 1, \dots, n$). The i th RNA codes for the i th enzyme E_i . The enzymes cyclically increase RNA's replication rates, namely, E_1 increases the replication rate of I_2 , E_2 increases the replication rate of I_3 , \dots , and, eventually, E_n increases the replication rate of I_1 .

For the ODE system (11.51), it is assumed that all reactions follow simple mass action kinetics. Since the total concentrations of RNAs and enzymes are constant ($I_1 + I_2 = c_I$ and $E_1 + E_2 = c_E$), it follows for the dilution terms that

$$\phi_E = \frac{a_1^T I_1 + a_2^T I_2}{c_E} \quad \text{and} \quad \phi_I = \frac{a_1^R I_1 E_1 + a_2^R I_2 E_2}{c_I}. \quad (11.52)$$

There are three steady-state solutions of Eqs. (11.51) and (11.52) with nonnegative concentration values. The following two steady states are stable:

$$\begin{aligned} \text{(i)} \quad E_1^{(i)} &= 0, & I_1^{(i)} &= 0, & E_2^{(i)} &= c_E, & I_2^{(i)} &= c_I, \\ \text{(ii)} \quad E_1^{(i)} &= c_E, & I_1^{(i)} &= c_I, & E_2^{(i)} &= 0, & I_2^{(i)} &= 0, \end{aligned} \quad (11.53)$$

and the third steady state is not stable (a saddle, see Chapter 15). In both stable steady states, one of the hypercycle survives and the other one dies out.

Example 11.7

Let's consider the dynamics of two competing hypercycles as described in Eq. (11.51). Here, the chosen parameters allow for a better growth of the first hypercycle ($a_1^T > a_2^T$, $a_1^R > a_2^R$). Nevertheless, it is not always the case that the first hypercycle survives and the second hypercycle dies out. The final steady state depends strongly on the initial conditions. The temporal behavior can be illustrated as a time course and in the phase plane as shown in Figure 11.15.

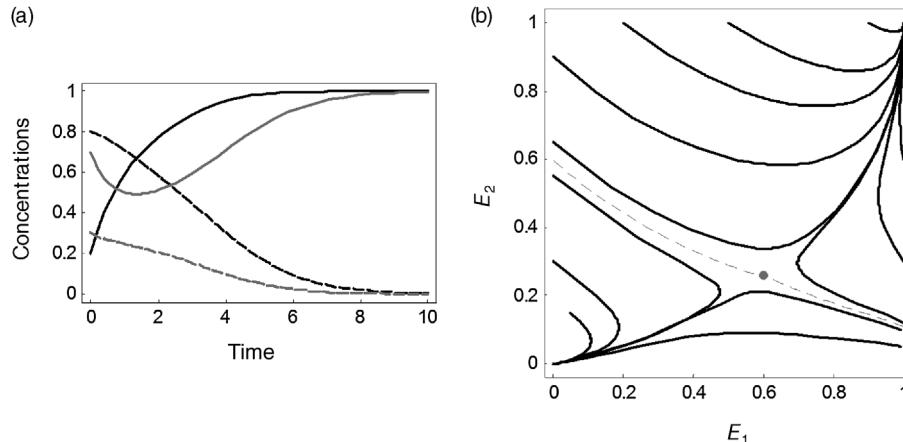


Figure 11.15 Dynamics of two competing two-component hypercycles as given in Eq. (11.51). Parameters: $a_1^T = 1$, $a_1^R = 1$, $a_2^T = 0.75$, $a_2^R = 0.75$, $c_l = 1$, and $c_E = 1$. (a) Time course of RNA and enzymes for the initial conditions $l_1(0) = 0.7$ and $E_1(0) = 0.2$. The solid and dashed lines represent quantities of first and second cycles, while the black and gray lines stand for enzyme and RNA concentrations, respectively. For the given parameters and initial conditions, the first hypercycle succeeds, while the second hypercycle dies out. (b) Phase plane representation for the first hypercycle (values for the second hypercycle are determined by the condition of constant total concentrations for enzymes and RNAs). Depending on the initial conditions, the system evolves toward one of the stable steady states (Eq. (11.53)). The gray dot at point (0.6, 0.257) marks the saddle point and the dotted gray line separates the basin of attraction of the both stable steady states.

For competing hypercycles, the dynamics depends both on the individual kinetic parameters of the hypercycles and on the initial conditions. Both steady states are attractive for a certain region of the concentration space. If the initial state belongs to the basin of attraction of a steady state, the system will always move toward this state. The hypercycle with less favorable parameters and lower growth rate may survive if it is present with high initial concentration. Nevertheless, the steady state that ensures the survival of the hypercycle with higher growth rates has a larger basin of attraction, and this hypercycle may also win with a lower initial concentration.

The dependence on initial conditions in the selection of hypercycles is a new property compared with the identical or complementary self-replication without catalysts. If a new hypercycle would emerge due to mutation, the initial concentrations of its compounds are very low. It can only win against the other hypercycles if its growth rate is much larger. Even then it may happen that it becomes extinct – depending on the actual basins of attraction for the different steady states. This dependence causes that during evolution even nonoptimal systems may succeed if they are present in sufficiently high initial concentration. This is called *once forever* selection favoring the survival of systems that had by chance good conditions, independent of their actual quality. This behavior is quite different from the classical Darwinian selection process.

Even the nonoptimal hypercycles combine advantageously the properties of polynucleotides (self-reproduction) with the properties of enzymes (enhancement of

speed and accuracy of polynucleotide replication). The cyclic organization of the hypercycle ensures its structural stability. This stability is even enhanced if hypercycles are organized in compartments [101]. This way, external perturbations by parasites can be limited and functionally advantageous mutations can be selected for.

In conclusion, one may state that the considered models, of course, can't explain the real-life origin process, because these models are based on various plausible assumptions rather than on a strong experimental evidence. Nevertheless, quasispecies and hypercycles provide a well-defined mathematical background for understanding the evolution of first molecular genetic systems. These models can be considered a step toward development of more realistic models.

11.3.5 Other Mathematical Models of Evolution: Spin Glass Model

The quasispecies concept as model of evolution (Section 11.3.3) implies a strong assumption: the Hamming distance between the particular and unique master sequences determines the selective value. Only one maximum of the selective value exists. Using the physical spin glass concept, we can construct a similar model, where the fitness function can have many local maxima.

D. Sherrington and S. Kirkpatrick [102,103] proposed a simple spin glass model to interpret the physical properties of the systems, consisting of randomly interacting

spins. This well-known model can be described briefly as follows:

- i) There is a system \mathbf{s} of spins s_k with $k = 1, \dots, N$, where N is large ($N \gg 1$). Each spin has a value of either 1 or -1 .
- ii) Spins can exchange their values by random interactions, which lead to spin reversals.
- iii) The energy $E(\mathbf{s})$ of the spin system is

$$E(\mathbf{s}) = - \sum_{k < l} J_{kl} s_k s_l. \quad (11.54)$$

The values J_{kl} are elements of the exchange interaction matrix with normally distributed random values and a probability density P_D of

$$P_D(J_{kl}) = (2\pi)^{-1/2} (N-1)^{1/2} \exp\left[-J_{kl}^2 \frac{(N-1)}{2}\right]. \quad (11.55)$$

According to Eqs. (11.54) and (11.55), the mean spin glass energy is zero, $\langle E \rangle = 0$, and for one-spin reversal the mean square root of energy variation is equal to 2.

The interesting feature of the spin glass concept is the large number of local energy minima M , where a local energy minimum is defined as a state \mathbf{s}_L at which any one-spin reversal would increase the energy E . Furthermore, there is a global energy minimum E_0 with $E_0 \approx 0.8N$.

A spin glass model of evolution represents a model sequence as a vector S_i of spins and a population as a set $\mathbf{S} = \{S_1, \dots, S_n\}$ of n sequences. Each sequence has a selective value that depends on its energy:

$$f(S_i) = \exp[-\beta E(S_i)] \quad (11.56)$$

for a choice of J_{kl} . β is a parameter for the selection intensity.

Spins s_{ik} and s_{il} interact according to the interaction matrix J_{kl} . The selective value of a sequence becomes maximal and its energy minimal when the combinations of spins in the sequences provide maximally cooperative interactions for a given matrix J_{kl} .

To this end, we consider again an evolutionary process with subsequent generations (compare with Section 11.3.3.1). The initial population is generated by chance. New generations are obtained by selection and mutation. Selection occurs with respect to the selective value defined in Eq. (11.56). Mutation is a sign reversal of a spin $s_{ik} \rightarrow -s_{ik}$ with certain probability P .

The evolutionary process can be followed by considering for successive generations T_i the number of sequences possessing certain energy $n(E)$ as illustrated in Figure 11.16.

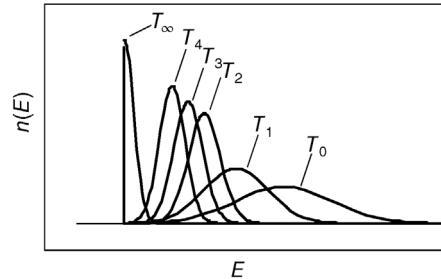


Figure 11.16 Schematic representation of sequence distributions $n(E)$ for subsequent generations $T_0 < T_1 < \dots < T_4$. For T_∞ , the system is trapped into a local energy minimum E_L . The global energy minimum is E_0 .

In the spin glass-type of evolution, the system converges to one of the local energy minima E_L , which can be different for different runs of simulation.

One may compare the evolutionary method with the sequential method of energy minimization, that is, consequent changes of symbols ($s_{ik} \rightarrow -s_{ik}$) of one sequence and fixation of only successful reversals. The sequential search is computationally simpler than the evolutionary search. Nevertheless, the evolutionary search results on average in a deeper local energy minimum E_L , because different valleys in energy landscape are examined in the evolution process simultaneously that are never reached in the sequential method. Thus, in the spin glass case, the evolutionary search has a certain advantage with respect to the sequential search: it results on average in the greater selective value.

11.3.6

The Neutral Theory of Molecular Evolution

The neutral theory of molecular evolution introduced by Motoo Kimura [83] states that mutations are mostly neutral or only slightly disadvantageous. The historical background for this statement was the deciphering of the genetic code and the structure of DNA by Watson and Crick in 1953 [104–106] and the understanding of the principle of protein synthesis. In addition, the evolutionary rate of amino acid substitutions and the protein polymorphism were estimated. The assumption of neutrality of mutations agrees with the mutational molecular substitution rate observed experimentally and with the fact that the rate of the substitutions for the less biologically important part of macromolecules is greater than that for the active centers of macromolecules.

The mathematical models of the neutral theory are essentially stochastic; that is, a relatively small population size plays an important role in the fixation of the neutral mutations.

The features of the neutral selection can easily be explained using the *game of neutral evolution*.

Consider populations (of sequences or organisms or, in the following example, of balls) with a finite population size n . The rules describing the evolutionary process are the following:

- i) The population contains black and white balls with a total population size n .
- ii) The next generation is created in two steps. First, all balls are duplicated preserving their color. A black ball has a black offspring, and a white ball a white one. Second, half of the population is removed irrespective of the “age” of a ball (i.e., whether it is an offspring or a parent ball) and with equal probability for black and white balls.

The state of the population is given by the number l of black balls. Consequently, there are $n - l$ white balls. The evolution is characterized by the probability P_{lm} for the transition from a state with l black balls to a state with m black balls in the next generation. P_{lm} can be calculated by applying combinatorial considerations:

$$P_{lm} = \begin{cases} \frac{\binom{2l}{m} \cdot \binom{2n-2l}{n-m}}{\binom{2n}{n}}, & \text{if } 2l-n \leq m \leq 2l, \\ 0, & \text{if } 2l < m \text{ or } m < 2l-n, \end{cases} \quad (11.57)$$

with $\binom{a}{b} = \frac{a!}{(a-b)!b!}$.

Possible evolutionary processes for a population of size 10 with initially 5 black and 5 white balls are illustrated in Figure 11.17.

The matrix P_{lm} determines a random Markovian process, which can be considered as an example of a simple stochastic genetic process. For the behavior of this process, the following hold:

- i) The process always converges to one of the two states $l = 0$ (only white balls) or $l = n$ (only black balls).
- ii) For large population size n , the characteristic number of generations needed to converge to either of these states is equal to $2n$.

Thus, although this evolution is purely neutral (black and white balls have equal chances to survive), only one species is selected.

It can be questioned how progressive evolution is possible if molecular substitutions are neutral. To answer this question, M. Kimura used the concept of gene duplication developed by S. Ohno [107]. According to M. Kimura, gene duplications create unnecessary, surplus DNA sequences, which in turn drift further because of random mutations, providing the raw material for the creation of new, biologically useful genes.

The evolutionary concepts of the neutral theory came from interpretations of biological experiments; this theory was strongly empirically inspired. The other type of theory, a more abstract one, was proposed by Stuart A. Kauffman: *NK* automata or Boolean networks (see Section 7.1).

11.4 Evolutionary Game Theory

Summary

Phenomena such as symbiosis and coevolution are usually neglected in optimality studies. The reason is that, by interacting with each other or with their environment, individuals effectively change their own fitness functions. The dynamics of cell populations and their fitness landscapes become coupled and can be studied with population dynamics. In evolutionary game theory, it is assumed that the selection advantage of biological traits depends on the prevalence of traits in other individuals. Such a coupling between individual behavior and population dynamics can help explain apparently paradoxical findings, for example, the fact that inefficient metabolic strategies can outcompete efficient strategies in a direct competition even if they perform worse when compared in isolation. Nevertheless, there are evolutionary mechanisms, such as kin selection, that can make efficient behavior evolutionarily stable. Thus, optimality in metabolism depends on the ecological context, and an absolute notion of optimality may not even make sense.



Figure 11.17 Representative runs for the game of neutral evolution. Starting with five black balls and five white balls at generation T_0 (bottom line), the system converges within several generations to a state with either only black balls (left panel) or only white balls (second and fourth panels from left). For the third and fifth panels, the final state is not yet decided at generation T_{10} .

Evolutionary theory suggests that mutation and selection can increase the average fitness of a population in a given environment. However, evolution need not lead to an optimization: first, gene mutations can spread and be fixed in a population without providing a fitness advantage – a phenomenon called neutral evolution. Second, nonoptimal traits may arise as by-products of other traits that are under selective pressure. But the optimality assumption also ignores a third, very common phenomenon: by their very presence, organisms influence their environment and can change their own fitness landscape; trying to maximize fitness within such flexible fitness landscapes may lead, paradoxically, to an overall fitness decrease: if you change your own fitness landscape while running, you may run uphill, but in fact move down.

Let us illustrate this point with an example. Natural selection favors trees that grow higher and thereby receive more light than their neighbors. This makes the selection pressure on height even stronger, and trees need to allocate more and more of their resources to growth, forcing other trees to do the same. Eventually, they reach a limit where growing taller would consume more resources than it provides – but the tree population does not receive more light than in the beginning. This deadlock (in terms of cost efficiency) is hard to avoid because a population of small trees – which would entail the least effort for all individuals – can always be invaded by higher growing mutants or outcompeted by other species of taller trees and is therefore inherently unstable. Similar competition scenarios exist for other characteristics of plants, such as average foliage heights, shading and self-shading, temporal growth strategies, flowering times, or the fraction of biomass allocated to leaves, stems, branches, and roots [108]. Thinking in terms of optimality only, we cannot grasp such phenomena: instead, we need to describe how an optimization of individuals affects their environment and thus the conditions under which their own optimization takes place. Evolutionary game theory provides concepts for this.

11.4.1 Social Interactions

Notions of “social behavior” also play a role in systems biology [109], for instance, when describing the production of beneficial or toxic substances or the consumption of nutrients, which can be described as public goods. Growing on a common substrate, cells could push their metabolism toward either maximal production rates (product per time) or maximal efficiency (product yield per substrate molecule), which often oppose each other. Optimality studies would start from such objectives and study how they can be maximized, or which of them

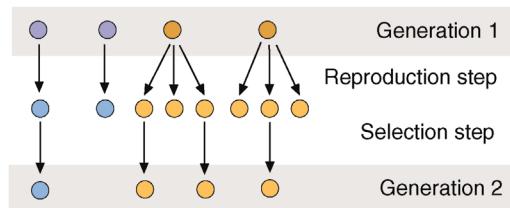


Figure 11.18 Simple model of reproduction and selection. A population consists of two subtypes (blue and brown), which differ in their numbers of offspring. Generation 1 consists of two blue and two brown individuals. They grow up and have offspring, which form the next generation. In a stochastic model, the number of children depends on the type of individual, and new adults are picked randomly from the previous generation of children such that the total population size is preserved. For large populations, this scheme can be approximated by the deterministic replicator equation (11.62).

seem to play a role [110,111]. However, *why* certain objectives matter for selection depends on how individuals interact and how they shape their environment.

Some central aspects of evolution can be understood from population dynamics. Figure 11.18 shows a simple scheme of neutral evolution as described in Section 11.3.6: individuals produce offspring, but only some of the children reach an age at which they also have offspring themselves. In a simple stochastic model, each individual has a certain number of children: n children from the population are randomly chosen to form the next generation, and the entire process is iterated. In a scenario of neutral evolution, each individual has the same number of children and the sizes of subpopulations (blue and brown in Figure 11.18) will drift randomly. A new mutant has a fixation probability (probability to spread in the entire population) of $1/n$, where n is the population size. If subpopulations have different reproduction rates, those with more offspring are more likely to outcompete others.

As a further complication, we may assume that reproduction rates are not fixed, but depend on the subpopulation sizes. For instance, if survival and reproduction of individuals are affected by random encounters between individuals, each genotype’s fitness will depend on the probability to meet individuals of certain genotypes, that is, on the relative subpopulation sizes.

With such a dependence, a mutant that exploits other individuals may lose its advantage when it becomes abundant. Examples are “selfish” virus mutants that cannot produce certain proteins and that instead use proteins produced by wild-type viruses in the same cell [112]. By reducing their genome size, the mutants reach a higher fitness. However, they also become dependent on the presence of wild-type viruses; as the mutant spreads in the population, its fitness decreases. In other cases, mutants may require a “critical mass” of individuals to

become beneficial; such mutants may easily disappear before they can start spreading. This effect of a “critical mass” is why some evolutionary steps (e.g., the usage of L-amino acids as building blocks of proteins) could hardly be reverted, even in case such reversion was beneficial (“once forever selection”). Finally, different species can influence each other’s evolution: pathogens, for instance, force the immune system to evolve, which in turn exerts a selection pressure on the pathogen strains.

11.4.2 Game Theory

Game theory [113] studies rational decisions in strategic games. In a two-player game, each player can choose between several strategies, and the payoff depends on the player’s and the coplayer’s choice. Both players try to maximize their own return. The payoffs are defined by a *payoff matrix F*: in a game with two strategies A and B, the payoffs for the first player read

$$\begin{array}{c|cc} & 2 : A & 2 : B \\ \hline 1 : A & f_{AA} & f_{AB} \\ 1 : B & f_{BA} & f_{BB} \end{array} \quad (11.58)$$

Rows correspond to the first player’s choice, and columns correspond to the choice of his/her coplayer. If a game is symmetric (as we assume for simplicity), the payoffs for the second player are given by the transposed matrix.

Hawk–Dove Game and Prisoner’s Dilemma

Figure 11.19 shows two well-studied games, called *hawk–dove game* [114] (or “snow drift game”) and *prisoner’s dilemma*. In both games, a cooperative strategy is confronted by an aggressive strategy. If both players act aggressively, both will lose. The payoff matrix of the *hawk–dove game* has the form

$$\begin{array}{c|cc} & 2 : \text{ dove} & 2 : \text{ hawk} \\ \hline 1 : \text{ dove} & v/2 & 0 \\ 1 : \text{ hawk} & v & (v - c)/2 \end{array} \quad (11.59)$$

with positive parameters v and c . The matrix F contains the payoffs for player 1 (left player in Figure 11.19) and the game is symmetric, so the payoff matrix for player 2 is the transposed matrix F^T .

The elements of matrix (11.59) can be seen as expected payoffs in the following scenario: players compete for a common resource (for instance, food) and can choose between two strategies, termed “hawk” and “dove.” A hawk initiates aggressive behavior and maintains it until he/she either wins the conflict or loses and gets seriously injured. A dove retreats immediately if the other player is aggressive. Figure 11.19 illustrates the four types of encounters and the resulting payoffs: if two doves meet, the resource is shared equally and each player obtains $v/2$ as a payoff. If a hawk meets a dove, the hawk obtains the full resource v and the dove gets nothing. If a hawk meets another hawk, he/she wins (payoff v) or becomes injured

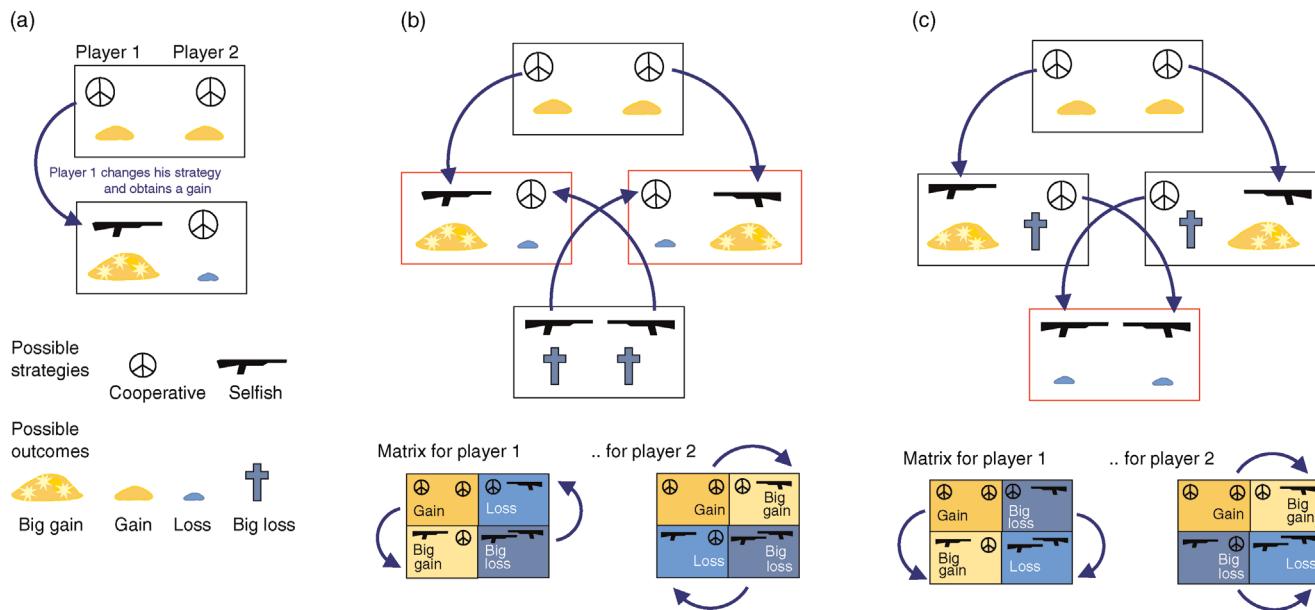


Figure 11.19 Hawk–dove game and prisoner’s dilemma. (a) Scheme of a two-player game with two strategies (cooperative and selfish). Knowing the opponent’s choice, a player may increase his/her payoff by a change of strategy (arrow). (b) Hawk–dove game. The four boxes illustrate all of the choices available to the two players (peaceful or aggressive behavior) and the possible payoffs. If none of the players can further increase his/her payoff, a stable situation – a so-called Nash equilibrium – is reached (red boxes). Tables at the bottom show the qualitative payoffs for both players as described by the payoff matrices. (c) Prisoner’s dilemma, shown in the same way.

(cost c) with equal probabilities, so the expected payoff is $(v - c)/2$. In the hawk–dove game, the cost of injury c is assumed to exceed the value of the resource v . The prisoner’s dilemma has a similar structure: its two strategies “cooperate” and “defect” correspond to “dove” and “hawk,” and the same type of payoff matrix is used. However, we assume that $c < v$, which means that the best response to a defector is to defect as well.

Nash Equilibrium

In both games, the optimal response to a coplayer’s strategy can be seen from the payoff matrix. In the hawk–dove game, one should choose “hawk” if the coplayer plays “dove,” but “dove” if the coplayer plays “hawk.” Hence, players should always respond with the opposite strategy. In the prisoner’s dilemma, this is different: here it is always best to defect, no matter which strategy the coplayer chooses. Paradoxically, the fact that both players choose their optimal strategies does *not* mean that the outcome is good for them. To predict the outcome, we need to search for states in which none of the players can improve his/her payoff by a change of strategy. Such a state is called a *Nash equilibrium*. The Nash equilibria for the above games are (1: hawk/2: dove) or (1: dove/2: hawk) for the hawk–dove game and (1: defect/2: defect) for the prisoner’s dilemma. Once a Nash equilibrium has been reached, none of the players has an incentive to deviate from his/her strategy, even if another state (e.g., 1: cooperate/2: cooperate) would provide higher payoffs for both players. In the prisoner’s dilemma, this leads to the paradoxical situation that both players end up in the unfavorable selfish state because all other states (including the favorable “cooperative” state) would be locally nonoptimal for one of the players. Even if both players had an agreement to choose “cooperate,” the agreement itself would be threatened because of the initial advantage of defecting.

Repeated Games

So far, we assumed that a game is played only once. Additional complexity emerges in sequential games, in which games are played repeatedly and players can respond to the coplayer’s behavior in previous rounds. A successful sequential strategy for the repeated prisoner’s dilemma is “tit for tat,” a form of *reciprocal altruism*: the player plays “cooperate” in the first round, and then responds in kind to each of the coplayer’s previous actions. Confronted with itself, this strategy is able to maintain the beneficial cooperate/cooperate state, but in contrast to the pure “cooperate” strategy, it also contains a threat against strategies that would attempt to exploit it.

11.4.3 Evolutionary Game Theory

Game theory was originally developed to study rational decisions based on an intention to maximize one’s own profit. However, later it became clear that the same formalism also applies to the evolution of biological traits, without any reference to conscious decisions. Evolutionary game theory [115] describes this evolution in population models, assuming that evolutionary fitness depends on the success in hypothetical games. Strategies are not chosen by rational consideration, but genetically determined, and their prevalence in a population changes dynamically, depending on how they perform in the game. The language of game theory can be helpful to describe such processes, but it is not absolutely necessary: the same results can also be obtained from population dynamics models without an explicit usage of game theoretical terminology.

In a common scenario, individuals encounter each other randomly. Depending on their genotypes, they play different strategies and the obtained average payoffs determine how fast each genotype will replicate. Given a two-player game, we can define a frequency-dependent fitness for each strategy. Assume two subpopulations playing strategies A and B with respective frequencies x_A and x_B , satisfying $x_A + x_B = 1$; with the payoff matrix \mathbf{F} , an individual of type A will obtain an expected payoff

$$f_A = f_{AA}x_A + f_{AB}x_B. \quad (11.60)$$

In a direct competition, successful genotypes will replicate faster and spread in the population. However, this will change the frequency of strategies, which then, following Eq. (11.60), affects the fitness functions. For example, a strategy may first spread, but then impair its own success at higher frequencies.

11.4.4 Replicator Equation for Population Dynamics

Population dynamics can be described by different mathematical frameworks, including deterministic and stochastic approaches and models with spatial structure (e.g., partial differential equations or cellular automata; see Section 7.3.1). In the following, we employ the *deterministic replicator equation*, a rate equation for the relative subpopulation sizes in a well-mixed population.

The Replicator Equation

In the replicator equation, the reproduction rate of a subpopulation is given by a baseline rate plus an average payoff from pairwise random encounters between individuals. The probabilities to meet members of different

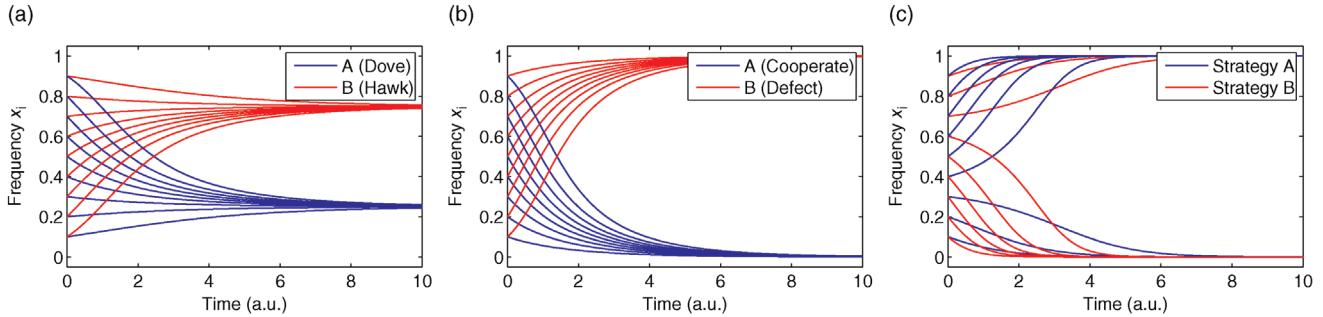


Figure 11.20 Population dynamics simulated by the replicator equation (11.62). Two populations playing different strategies are represented by their time-dependent frequencies (normalized population sizes x_i). The curves show simulation results starting from different initial frequencies. (a) The hawk–dove game leads to coexistence of individual strategies. (b) In the prisoner’s dilemma, the strategy “defect” dominates “cooperate,” which eventually dies out. (c) Two evolutionarily stable strategies forming a bistable pair: who the winner is depends on the initial frequencies.

subpopulations are given by their frequencies x_1, x_2, \dots , which sum up to 1. When individuals of type i meet individuals of type j , they obtain a payoff f_{ij} . Such encounters happen with probability x_j , so the average payoff of strategy i is

$$f_i(\mathbf{x}) = \sum_j f_{ij}x_j. \quad (11.61)$$

In a simple deterministic model, the frequencies x_i evolve in time according to a selection equation

$$\frac{dx_i}{dt} = x_i(f_i(\mathbf{x}) - \langle f \rangle) \quad (11.62)$$

called the replicator equation. The term $\langle f \rangle = \sum_j x_j f_j(\mathbf{x})$ denotes the current average fitness of the population. Subtracting it in Eq. (11.62) ensures that the frequencies remain normalized (i.e., $\sum_i x_i = 1$) and that the baseline fitness value cancels out.

A competition between subpopulations A and B can lead to different types of dynamics, including sustained oscillations (see Section 11.4.6) and convergence to a fixed point. A subpopulation frequency x_i remains constant if

$$0 = x_i(f_i(\mathbf{x}) - \langle f \rangle), \quad (11.63)$$

that is, if the strategy has died out ($x_i = 0$) or if its fitness $f_i = \langle f \rangle$ matches the average fitness of the entire population. Thus, in a fixed point of Eq. (11.62), all surviving subpopulations have the same fitness – given the subpopulation frequencies in this state.

Outcomes of Frequency-Dependent Selection

The number and types of fixed points of Eq. (11.62) depend on the payoff matrix. There are different cases: (i) A *dominant* strategy is a best reply to itself and to the other strategy (e.g., “defect” in the prisoner’s dilemma); it always wins, that is, its frequency will approach $x_i = 1$.

(ii) *Coexistence*: if two strategies are best replies to each other, they can coexist in stable proportions (like, for instance, in the hawk–dove game). (iii) *Bistability*: if each strategy is a best reply to itself, many individuals of the other strategy would be needed to undermine it. In this bistable case, one of the strategies eventually wins ($x_i \rightarrow 1$), and who the winner is depends on their initial frequencies. Strategy A is called *risk-dominant* if it has the larger basin of attraction ($f_{AA} + f_{AB} > f_{BA} + f_{BB}$). Such a strategy will win if both strategies start at equal subpopulation frequencies. (iv) *Neutrality*: both strategies have the same expected payoff independently of the subpopulation frequencies: in terms of fitness, they are effectively identical. According to the deterministic equation (11.62), the subpopulation sizes will be constant in time; in stochastic population models, they would drift randomly.

Figure 11.20 shows simulations for a payoff matrix (11.59) with values $v = 3$, $c = 4$ ((a), hawk–dove game) and $v = 4$, $c = 3$ ((b), prisoner’s dilemma). For each type of game, curves with different initial frequencies between 0.1 and 0.9 are shown. The hawk–dove game leads to a coexistence at a frequency $x_2 = c/v = 3/4$ of hawks; in the prisoner’s dilemma, the cooperative strategy always dies out. In a third game (Figure 11.20c), with an identity payoff matrix $\mathbf{F} = \mathbf{I}$, each strategy is a best response to itself. The dynamics is bistable, so different initial conditions can lead to success of either A or B.

11.4.5 Evolutionarily Stable Strategies

A basic type of question in evolutionary game theory is whether some population can be invaded by mutants playing a different strategy. If the resident strategy dominates all possible mutant strategies, it will outcompete any mutants and is called *unbeatable*. However, strategies

need not be dominant to persist in evolution. The weaker concept of *evolutionarily stable strategies* (ESS) takes into account that a mutant initially appears in small numbers. Strategy A is evolutionary stable if competing strategies B, starting at a small frequency, cannot invade a population of type A. There are two ways by which A can be evolutionary stable: either A dominates B or A and B form a bistable pair and the mutant subpopulation does not reach the critical size for taking over the population. An evolutionarily stable strategy, confronted with itself, is always a Nash equilibrium. However, the opposite does not hold: a strategy may form a Nash equilibrium with itself, but may still be invaded by new mutants, so the resident strategy is not evolutionarily stable.

What is the mathematical criterion for evolutionarily stable strategies? Assume that A is the strategy of the resident population and B is a possible mutant strategy. With relative frequencies x_A and x_B , the average payoffs for individuals of type A and B read

$$\begin{aligned} f_A &= f_{AA}x_A + f_{AB}x_B, \\ f_B &= f_{BA}x_A + f_{BB}x_B. \end{aligned} \quad (11.64)$$

By definition, A is evolutionarily stable if it cannot be invaded by a few individuals of type B: thus, $f_A > f_B$ must hold for small frequencies x_B . If x_B is very small, an individual (of type A or B) will almost never encounter an individual of type B. Therefore, we can disregard the second terms in (11.64) and obtain the simple condition $f_{AA} > f_{BA}$. However, in the case of equality ($f_{AA} = f_{BA}$), the rare encounters with B individuals will play a role, and we must additionally require that $f_{AB} > f_{BB}$. Thus, an evolutionarily stable strategy A must perform better than B when encountering individuals of type A; if it performs only equally well, it must at least perform better in encounters with individuals of type B.

Example 11.8 Evolutionarily Stable Strategy in the Prisoner's Dilemma

Using this criterion, we can determine evolutionarily stable strategies in the prisoner's dilemma. "Cooperate" is not evolutionarily stable because then $f_{AA} = v/2$ would have to be larger than or equal to $f_{BA} = v$, which is not the case. "Defect," in contrast, is evolutionarily stable if $f_{BB} \geq f_{AB}$, or $(v - c)/2 \geq 0$, that is, if the cost of an injury is smaller than the gain of the resource, as we assumed in the prisoner's dilemma. Therefore, a few defectors can successfully invade a population of cooperators, but a few cooperators cannot invade a defector population.

11.4.6 Dynamical Behavior in the Rock–Scissors–Paper Game

Frequency-dependent selection can lead not only to stationary states in population dynamics, but also to sustained dynamic behavior [116] as in the following example. The rock–scissors–paper game consists of the three strategies rock (A), paper (B), scissors (C), which beat each other in a circle (Figure 11.21a): A dominates C, B dominates A, and C dominates B. This game can be used to describe an arms race between three strains of *E. coli* bacteria [117]. A strain K ("killer") produces the toxin colicin against other bacteria, together with an immunity protein to protect itself. Strain I ("immune") produces the immunity protein, but not the toxin, and strain S ("sensitive") produces neither of them. In pairwise competitions, K kills S with its toxin, I outcompetes K because it does not produce toxin and saves resources for growth, and S outcompetes I because it saves the effort of producing the immunity protein.

In a direct competition, each strategy can invade the previous one, so the subpopulation sizes will oscillate. Figure 11.21 shows dynamic behavior resulting from two different payoff matrices

$$F = \begin{pmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{pmatrix}, \quad F' = \begin{pmatrix} 0 & -1 & 2 \\ 2 & 0 & -1 \\ -1 & 2 & 0 \end{pmatrix}, \quad (11.65)$$

which lead, respectively, to sustained and damped oscillations. Ongoing arms races have been suggested [118] as an explanation for the huge diversity found in microbial communities. At first sight, such diversity would contradict the *competitive exclusion principle*, which states that the number of species surviving in a fixed environment must equal the number of limited resources ("paradox of the plankton") [119].

11.4.7 Evolution of Cooperative Behavior

Many types of behavior in people, animals, and even in microbes can be seen as forms of cooperation. Formally, cooperation can be defined as follows: a *cooperator* C pays a cost c for another individual to receive a benefit $b > c$. A *defector* D does not deal out benefits, but will receive benefits from cooperators. The payoff matrix

	2 cooperates	2 defects	
1 cooperates	$f_{CC} = b - c$	$f_{CD} = -c$	(11.66)
1 defects	$f_{DC} = b$	$f_{DD} = 0$	

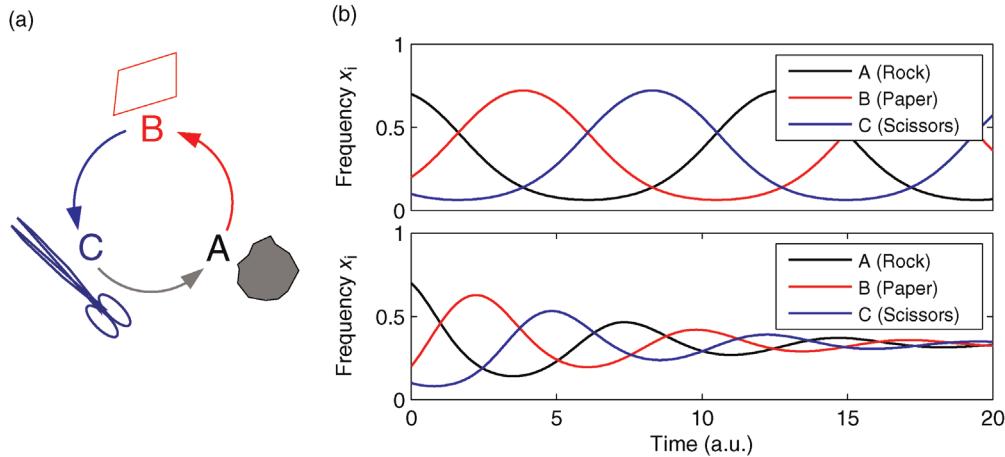


Figure 11.21 Population dynamics in the rock–scissors–paper game. (a) Each strategy (A: rock; B: paper; C: scissors) beats the previous one in a circle (arrowheads point to the winning strategy). (b) Simulated dynamics based on the replicator equation. The population frequencies vary periodically. Depending on the specific payoff matrix chosen, the subpopulations replace each other periodically (top) or converge to a stable mixture (bottom).

shows that defectors will dominate cooperators because $f_{DC} > f_{CC}$ and $f_{DD} > f_{CD}$. In this basic scenario, cooperation will not be evolutionarily stable because $f_{DC} \leq f_{CC}$ can never be satisfied. The prediction would be that evolution selects for noncooperative, “selfish” traits. However, various examples of cooperation have evolved, and evolutionary theory has found ways to explain this. In fact, cooperation can emerge in a number of specific evolution scenarios [120], which we shall now discuss.

Kin Selection

Altruism toward one’s own genetic relatives can evolve by a mechanism called *kin selection*. To see how this works, let us consider a gene in a single individual: for the gene’s long-term reproduction, it does not matter whether it is reproduced via its “host” individual or via other individuals that carry the same gene. This second possibility provides a selection advantage to genes that promote altruistic behavior. If two individuals X and Y are relatives, there is a chance r that an arbitrarily chosen gene of X is shared by Y because of common inheritance. This so-called *coefficient of relatedness* has a value of $r = 1/2$ for siblings and $r = 1/8$ for cousins. Altruistic behavior toward relatives can help the relatives reproduce, and thereby reproduce some shared genes: therefore, the payoff (seen from the perspective of a single gene) is increased by the amount of the coplayer’s payoff (given by the transposed payoff matrix), multiplied by r . According to the modified payoff matrix

	2 cooperates	2 defects	
1 cooperates	$(b - c)(1 + r)$	$br - c$	(11.67)
1 defects	$b - rc$	0	

kin selection will take place (i.e., altruism toward kins will be evolutionarily stable) if $f_{CC} > f_{CD}$, that is, $r > c/b$. J.B. S. Haldane has expressed this criterion, called *Hamilton’s rule*, as “I’d lay down my life for two brothers or eight cousins.” The higher the typical coefficient of relatedness in encounters (large r), the more easily the rule can be satisfied. Hamilton’s rule has been studied in detail with bacteria, where cooperators produce a beneficial substance (an activator for the expression of an antibiotic resistance gene), while defectors only benefit from it [121].

Cooperation with relatives – and therefore kin selection – requires *kin recognition*. The plant *Cakile edentula*, for instance, grows stronger roots when it shares a pot with other plants, thus competing with them for water and nutrients. However, when groups of siblings share one pot, they grow weaker roots [122]. This suggests two things: that competition among siblings is weaker, and that siblings can recognize each other, possibly by sensing chemicals via their roots.

Reciprocity and Group Selection

Cooperation can also evolve if individuals recompense each other for earlier cooperative behavior: different scenarios presuppose that individuals can remember earlier encounters (direct reciprocity), know the reputation of other individuals (indirect reciprocity), or interact locally with neighbors in a network (network reciprocity). In spatially structured models, cooperators can grow in local clusters, which makes them less prone to exploitation by defectors. Moreover, cooperative behavior can evolve by a mechanism called *group selection*, which involves two levels of competition: individuals compete within groups,

and groups compete with each other. In this scenario, selfish behavior within the group may be selected against because it can lead to extinction of the entire group and, thus, of the selfish individual itself.

In a group selection model by Nowak [120], individuals obtain a fitness value f from encounters within their group and replicate with a rate $1 - \omega + \omega f$, where ω is a parameter describing the intensity of selection. Groups can only grow to a maximum size of n – when a larger size is reached, one of the two things will happen: either a randomly selected individual from the group dies (with probability $1 - p$) or the group is split into two and another randomly selected group dies (with probability p). Due to the second possibility, groups must replicate fast to evade extinction.

In this model, defectors have a direct growth advantage within their own group, but this advantage is counterbalanced by the risk of killing their group as a result of their selfish behavior. If n denotes the size at which groups are split and m is the total number of groups, then in cases of weak selection ω and small splitting probability p , we obtain the effective payoff matrix [120]

	2 cooperates	2 defects
1 cooperates	$f_{CC} = (b - c)(m + n)$	$f_{CD} = (b - c)m - cn$
1 defects	$f_{DC} = bn$	$f_{DD} = 0$

(11.68)

Again, cooperation becomes evolutionarily stable if $f_{CC} > f_{CD}$, in this case if $b/c > 1 + n/m$. This criterion is more likely to be satisfied when groups are small (small n) and numerous (large m).

11.4.8 Compromises between Metabolic Yield and Efficiency

In Section 11.1, we saw that choices between high-yield and low-yield metabolic strategies can be predicted from optimality considerations, assuming that protein resources are allocated in a way that maximizes cell growth. Which of the strategies is optimal depends on several factors, in particular nutrient supply and the extra protein cost of the high-yield strategy [123]. However, a fundamental problem remained open: with a low-yield strategy, cell populations grow to smaller sizes and are less sustainable because they deplete the nutrients in their environment. But still, in a direct competition with efficient cells, this strategy brings a growth advantage. If this is true, how can sustainable, high-yield strategies, in which growth is deliberately restricted, evolve at all? The existence of multicellular organisms, which strictly control cell growth, shows that this must be possible.

High-Yield and Low-Yield Fluxes Seen as Strategies

Pfeiffer *et al.* [124] applied a game theoretical perspective to the study of high-flux or high-yield strategies in metabolism. Instead of using a notion of “optimal” strategies, they asked whether each of the strategies can be evolutionarily stable. Since high-yield strategies spare resources for other cells, they can be seen as “cooperative,” while low-yield strategies can be seen as “selfish.” The competition between selfish and cooperative individuals for shared nutrients leads to a dilemma called “tragedy of the commons”. If respirators and fermenters grow separately on a constant sugar supply, the respirators will grow to a higher cell density than the fermenters, so their fitness as a population will be higher. However, in a direct competition for sugar, respirators are outperformed by fermenters, so their population may be invaded by fermenting mutants. Only fermentation is evolutionarily stable in this scenario.

Nevertheless, respiration has evolved. Unicellular organisms, which compete for external resources, tend to use fermentation, while cells in higher animals tend to use respiration, with cancer cells being a prominent exception (Warburg effect) [125]. In a multicellular organism, ingested nutrients are shared among cells, so there is no competition for food: instead, the well-being of the entire organism requires cooperative behavior. Since cells in an organism are genetically identical, this can be seen as an example of Hamilton’s rule.

Pfeiffer *et al.* suggested that the evolution of multicellularity, which has appeared at least 10 times independently in the tree of life, may have been triggered by a competition between fermenting and respiring cells [124]. When oxygen accumulated in the atmosphere as a waste product of photosynthesis, respiration arose as a new, efficient energy source. However, as a downside of their high yield, respiring cells would have grown more slowly than the existing fermenters and would have been outcompeted. This dilemma could have been solved by group selection: the formation of cell aggregates may have enabled respirors to evade competition and, thus, to benefit from their new energy source.

Spatial Structure Can Promote Cooperative Behavior

The evolution of yield-efficient strategies can be favored by spatially structured environments. This has been shown for bacteria undergoing an evolution in the laboratory. In a normal serial dilution experiment, cells are suspended in a growth medium and share the same resources; this leads to a selection for fast-growing cells. To create an incentive for yield-efficient behavior, Bachmann *et al.* grew *Lactococcus lactis* bacteria in an emulsion of small water droplets in oil, each containing only a few cells [126] (see Figure 11.22). In each dilution step,

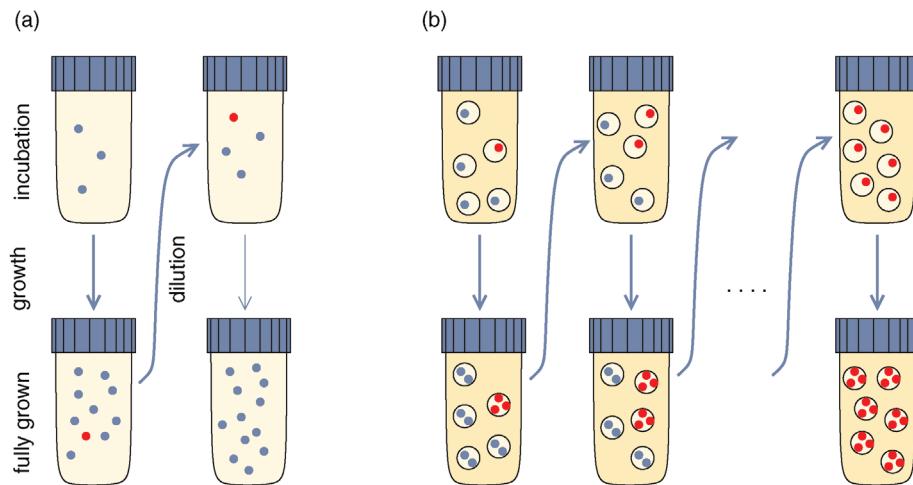


Figure 11.22 A serial dilution experiment in which yield-efficient behavior can evolve. Slowly growing bacteria mutants with a high ATP yield (red) compete with the low-yield, fast-growing wild-type bacteria (blue). (a) In a cell suspension, mutants would be outcompeted due to their slower growth. (b) In a water-in-oil emulsion (bacteria living within small droplets), the mutants can replicate to larger numbers within droplets, allowing them to take over the population. (Redrawn from Ref. [126].)

the emulsion was broken, cells were diluted in fresh medium, and a new emulsion was prepared. The shared resources within droplets lead to an evolution toward yield-efficient behavior, namely, mixed acid fermentation

(producing three molecules of ATP per glucose molecule) instead of lactate excretion (producing only two molecules of ATP, but at a faster flux), as well as a measurable higher biomass yield, but also smaller cell sizes.

Exercises

Section 11.1

- 1) *ATP production in glycolysis.* A glycolytic pathway converts glucose into lactate and effectively turns two ADP molecules into two ATP molecules. (a) Is this process thermodynamically feasible? Assume a decrease of Gibbs free energy by 205 kJ mol^{-1} for the conversion of glucose into two lactate molecules and an increase of 49 kJ mol^{-1} for each ADP molecule converted to ATP. (b) Imagine alternative variants of glycolysis that also convert glucose into lactate, but produce different numbers of ATP molecules. What are the minimal and maximal numbers possible? (c) Assume, for simplicity, that the glycolytic flux is proportional to the total decrease of Gibbs free energy. Which number of ATP molecules produced would lead to (i) a maximal rate of ATP production or (ii)

maximal yield efficiency, that is, maximal ATP production per amount of glucose consumed?

Section 11.2

- 2) Calculate the steady-state flux for an unbranched metabolic pathway with four reactions. Use formula (4.15). Use the parameter and concentration values $P_0 = P_r = 1$, $E_1 = E_2 = E_3 = E_4 = 1$, $q_1 = q_2 = q_3 = q_4 = 5$, and $k_1 = k_2 = k_3 = k_4 = 1$.
- 3) Determine the maximal steady-state flux for the case where the enzyme concentrations may vary in the range from 0 to 2. Compare with the steady-state flux for the case where all enzyme concentrations are equal to 2.

- 4) Repeat Exercise 11.3 by applying the conditions (i) that the sum of enzyme concentrations is 8 and (ii) all individual enzyme concentrations are positive.
- 5) Calculate optimal enzyme concentrations according to formula (11.19) and the resulting steady-state flux.
- 6) Consider the temporal regulation of enzymatic pathways. Calculate the transition time for the case where (i) the enzyme concentrations are constant and (ii) the pathway contains only one reaction, $S_0 \leftrightarrow S_1$, and the temporal change of the reaction is given by the decay of S_0 .

Note: First integrate the appropriate set of equations for the dynamics of S_1 (Eq. (11.27)) and then introduce the result into Eq. (11.28).

Repeat for a pathway consisting of two consecutive enzymes.

Section 11.3

- 7) Consider the case of selection under constant total concentration. Could you predict which species will survive if you would describe the system with a stochastic approach?
- 8) Can you find general conditions under which hypercycles become extinct?

- 9) Compute steady states for a system of three competing hypercycles (choose reasonable parameter values).

Section 11.4

- 10) *Fixed point of the replicator equation.* Consider the deterministic replicator equation with a fitness function defined by the hawk–dove game (1: dove; 2: hawk). Show that the stationary ratio x_1/x_2 of subpopulation frequencies is given by $(c - v)/v$.
- 11) *Maximal efficiency or maximal rate?* Consider the following model of competition for a common resource [127]: several microbial strains (denoted by i) compete for a resource S (with concentration s), which arrives at a constant rate v . The consumption rates $J_i^S(s)$ per cell and the efficiencies $\eta_i = J_i^{\text{ATP}}(s)/J_i^S(s)$ differ between the strains. Cells replicate at a rate proportional to their ATP production and die at a constant rate, with the same constants for all strains. Show that (a) for a single strain (no competition with other strains), the steady-state population size is proportional to the efficiency η_i of ATP production and that (b) If several strains compete for the resource, a strain's success depends on the rate $J_i^{\text{ATP}}(S)$ of ATP production.

References

- 1 Dekel, E. and Alon, U. (2005) Optimality and evolutionary tuning of the expression level of a protein. *Nature*, 436, 588–692.
- 2 Alon, U. (2003) Biological networks: the tinkerer as an engineer. *Science*, 301, 1866–1867.
- 3 Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, University of Michigan Press, Ann Arbor, MI.
- 4 Kirschner, M. and Gerhart, J. (1998) Evolvability. *Proc. Natl. Acad. Sci. USA*, 95 (15), 8420–8427.
- 5 Kashtan, N., Noor, E., and Alon, U. (2007) Varying environments can speed up evolution. *Proc. Natl. Acad. Sci. USA*, 104 (34), 13711–13716.
- 6 Barve, A. and Wagner, A. (2013) A latent capacity for evolutionary innovation through exaptation in metabolic systems. *Nature*, 500, 203–206.
- 7 Wolff, J. (1892) *Das Gesetz der Transformation der Knochen*, Verlag von August, Berlin.
- 8 Frost, H.M. (2001) From Wolff's law to the Utah paradigm: insights about bone physiology and its clinical applications. *Anat. Rec.*, 262, 398–419.
- 9 Robling, A.G., Castillo, A.B., and Turner, C.H. (2006) Biomechanical and molecular regulation of bone remodeling. *Annu. Rev. Biomed. Eng.*, 8, 455–498.
- 10 Huiskens, R. et al. (2000) Effects of mechanical forces on maintenance and adaptation of form in trabecular bone. *Nature*, 405, 704–706.
- 11 Heinrich, R. and Rapoport, T.A. (2005) Generation of non-identical compartments in vesicular transport systems. *J. Cell Biol.*, 168, 271–280.
- 12 Heckmann, D., Schulze, S., Denton, A., Gowik, U., Westhoff, P., Weber, A.P.M., and Lercher, M.J. (2013) Predicting C₄ photosynthesis evolution: modular, individually adaptive steps on a Mount Fuji fitness landscape. *Cell*, 153 (7), 1579–1588.
- 13 Chait, R., Craney, A., and Kishony, R. (2007) Antibiotic interactions that select against resistance. *Nature*, 446, 668–671.
- 14 Gomes, A.L.C., Galagan, J.E., and Segrè, D. (2013) Resource competition may lead to effective treatment of antibiotic resistant infections. *PLoS One*, 8 (12), e80775.
- 15 Trigg, G.L. (2005) *Mathematical Tools for Physicists*, Wiley-VCH Verlag GmbH, Weinheim.
- 16 Feynman, R. (1985) *QED: The Strange Theory of Light and Matter*, Princeton University Press, Princeton, NJ.
- 17 Fleming, R.M.T., Maes, C.M., Saunders, M.A., Ye, Y., and Palsson, B.Ø. (2012) A variational principle for computing nonequilibrium fluxes and potentials in genome-scale biochemical networks. *J. Theor. Biol.*, 292, 71–77.
- 18 Eames, M. and Kortemme, T. (2012) Cost–benefit tradeoffs in engineered lac operons. *Science*, 336, 911–915.
- 19 Reich, J.G. (1983) Zur Ökonomie im Proteinhaushalt der lebenden Zelle. *Biomed. Biochim. Acta*, 42 (7/8), 839–848.

- 20** Liebermeister, W., Klipp, E., Schuster, S., and Heinrich, R. (2004) A theory of optimal differential gene expression. *Biosystems*, 76, 261–278.
- 21** Kalisky, T., Dekel, E., and Alon, U. (2007) Cost–benefit theory and optimal design of gene regulation functions. *Phys. Biol.*, 4, 229–245.
- 22** Dill, K.A., Ghosh, K., and Schmit, J.D. (2011) Physical limits of cells and proteomes. *Proc. Natl. Acad. Sci. USA*, 108 (44), 17876–17882.
- 23** Shoval, O., Sheftel, H., Shinar, G., Hart, Y., Ramote, O., Mayo, A., Dekel, E., Kavanagh, K., and Alon, U. (2012) Evolutionary trade-offs, Pareto optimality, and the geometry of phenotype space. *Science*, 336, 1157–1160.
- 24** Sheftel, H., Shoval, O., Mayo, A., and Alon, U. (2013) The geometry of the Pareto front in biological phenotype space. *Ecol. Evol.*, 3 (6), 1471–1483.
- 25** Bar-Even, A., Noor, E., Savir, Y., Liebermeister, W., Davidi, D., Tawfik, D.S., and Milo, R. (2011) The moderately efficient enzyme: evolutionary and physicochemical trends shaping enzyme parameters. *Biochemistry*, 21, 4402–4410.
- 26** Heinrich, R. and Holzhütter, H.-G. (1985) Efficiency and design of simple metabolic systems. *Biomed. Biochim. Acta*, 44 (6), 959–969.
- 27** Edwards, J.S. and Palsson, B.Ø. (2000) Metabolic flux balance analysis and the *in silico* analysis of *Escherichia coli* K-12 gene deletions. *BMC Bioinform.*, 1, 1.
- 28** Holzhütter, H.-G. (2004) The principle of flux minimization and its application to estimate stationary fluxes in metabolic networks. *Eur. J. Biochem.*, 271 (14), 2905–2922.
- 29** Goelzer, A. and Fromion, V. (2011) Bacterial growth rate reflects a bottleneck in resource allocation. *Biochim. Biophys. Acta*, 1810 (10), 978–988.
- 30** Heinrich, R. and Hoffmann, E. (1991) Kinetic parameters of enzymatic reactions in states of maximal activity: an evolutionary approach. *J. Theor. Biol.*, 151, 249–283.
- 31** Klipp, E. and Heinrich, R. (1999) Competition for enzymes in metabolic pathways: implications for optimal distributions of enzyme concentrations and for the distribution of flux control. *Biosystems*, 54, 1–14.
- 32** Klipp, E., Heinrich, R., and Holzhütter, H.-G. (2002) Prediction of temporal gene expression. Metabolic optimization by re-distribution of enzyme activities. *Eur. J. Biochem.*, 269, 1–8.
- 33** Zaslaver, A., Mayo, A.E., Rosenberg, R., Bashkin, P., Sberro, H., Tsalyuk, M., Surette, M.G., and Alon, U. (2004) Just-in-time transcription program in metabolic pathways. *Nat. Genet.*, 36, 486–491.
- 34** Molenaar, D., van Berlo, R., de Ridder, D., and Teusink, B. (2009) Shifts in growth strategies reflect tradeoffs in cellular economics. *Mol. Syst. Biol.*, 5, 323.
- 35** Bachmann, H., Fischlechner, M., Rabbers, I., Barfa, N., Branco dos Santos, F., Molenaar, D., and Teusink, B. (2013) Availability of public goods shapes the evolution of competing metabolic strategies. *Proc. Natl. Acad. Sci. USA*, 110 (35), 14302–14307.
- 36** Schuetz, R., Kuepfer, L., and Sauer, U. (2007) Systematic evaluation of objective functions for predicting intracellular fluxes in *Escherichia coli*. *Mol. Syst. Biol.*, 3, 119.
- 37** Wagner, A. and Fell, D.A. (2001) The small world inside large metabolic networks. *Proc. Biol. Sci.*, 268 (1478), 1803–1810.
- 38** Hatzimanikatis, V., Li, C., Ionita, J.A., Henry, C.S., Jankowski, M.D., and Broadbelt, L.J. (2005) Exploring the diversity of complex metabolic networks. *Bioinformatics*, 21 (8), 1603–1609.
- 39** Bar-Even, A., Flamholz, A., Noor, E., and Milo, R. (2012) Rethinking glycolysis: a perspective on the biochemical logic of glucose fermentation. *Nat. Chem. Biol.*, 8, 509–517.
- 40** Stephani, A., Nuño, J.C., and Heinrich, R. (1999) Optimal stoichiometric designs of ATP-producing systems as determined by an evolutionary algorithm. *J. Theor. Biol.*, 199, 45–61.
- 41** Meléndez-Hevia, E. and Isodoro, A. (1985) The game of the pentose phosphate cycle. *J. Theor. Biol.*, 117, 251–263.
- 42** Noor, E., Eden, E., Milo, R., and Alon, U. (2010) Central carbon metabolism as a minimal biochemical walk between precursors for biomass and energy. *Mol. Cell*, 39, 809–820.
- 43** Waddell, T.G., Repovic, P., Meléndez-Hevia, E., Heinrich, R., and Montero, F. (1999) Optimization of glycolysis: new discussions. *Biochem. Educ.*, 27 (1), 12–13.
- 44** Meléndez-Hevia, E., Waddell, T.G., Heinrich, R., and Montero, F. (1997) Theoretical approaches to the evolutionary optimization of glycolysis – chemical analysis. *Eur. J. Biochem.*, 244, 527–543.
- 45** Noor, E., Bar-Even, A., Flamholz, A., Reznik, E., Liebermeister, W., and Milo, R. (2014) Pathway thermodynamics uncovers kinetic obstacles in central metabolism. *PLoS Comput. Biol.*, 10, e100348.
- 46** Shachrai, I., Zaslaver, A., Alon, U., and Dekel, E. (2010) Cost of unneeded proteins in *E. coli* is reduced after several generations in exponential growth. *Mol. Cell*, 38, 1–10.
- 47** Stucki, J.W. (1980) The optimal efficiency and the economic degrees of coupling of oxidative phosphorylation. *Eur. J. Biochem.*, 109, 269–283.
- 48** Stoebel, D.M., Dean, A.M., and Dykhuizen, D.E. (2008) The cost of expression of *Escherichia coli* lac operon proteins is in the process, not in the products. *Genetics*, 178 (3), 1653–1660.
- 49** Hoppe, A., Richter, C., and Holzhütter, H.-G. (2011) Enzyme maintenance effort as criterion for the characterization of alternative pathways and length distribution of isofunctional enzymes. *Biosystems*, 105 (2), 122–129.
- 50** Bar-Even, A., Noor, E., Lewis, N.E., and Milo, R. (2010) Design and analysis of synthetic carbon fixation pathways. *Proc. Natl. Acad. Sci. USA*, 107 (19), 8889–8894.
- 51** Byrne, D., Dumitriu, A., and Segrè, D. (2012) Comparative multi-goal tradeoffs in systems engineering of microbial metabolism. *BMC Syst. Biol.*, 6, 127.
- 52** Schuetz, R., Zamboni, N., Zampieri, M., Heinemann, M., and Sauer, U. (2012) Multidimensional optimality of microbial metabolism. *Science*, 336 (6081), 601–604.
- 53** Beg, Q.K., Vazquez, A., Ernst, J., de Menezes, M.A., Bar-Joseph, Z., Barabási, A.-L., and Oltvai, Z.N. (2007) Intracellular crowding defines the mode and sequence of substrate uptake by *Escherichia coli* and constrains its metabolic activity. *Proc. Natl. Acad. Sci. USA*, 104 (31), 12663–12668.
- 54** Tepper, N., Noor, E., Amador-Noguez, D., Haraldsdóttir, H.S., Milo, R., Rabinowitz, J., Liebermeister, W., and Shlomi, T. (2013) Steady-state metabolite concentrations reflect a balance between maximizing enzyme efficiency and minimizing total metabolite load. *PLoS One*, 8 (9), e75370.
- 55** Flamholz, A., Noor, E., Bar-Even, A., Liebermeister, W., and Milo, R. (2013) Glycolytic strategy as a tradeoff between energy yield and protein cost. *Proc. Natl. Acad. Sci. USA*, 110 (24), 10039–10044.
- 56** Wortel, M.T., Peters, H., Hulshof, J., Teusink, B., and Bruggeman, F.J. (2014) Metabolic states with maximal specific rate carry flux through an elementary flux mode. *FEBS J.*, 281 (6), 1547–1555.
- 57** Müller, S., Regensburger, G., and Steuer, R. (2014) Enzyme allocation problems in kinetic metabolic networks: optimal solutions are elementary flux modes. *J. Theor. Biol.*, 347, 182–190.

- 58** Zelcbuch, L., Antonovsky, N., Bar-Even, A., Levin-Karp, A., Barenholz, U., Dayagi, M., Liebermeister, W., Flamholz, A., Noor, E., Amram, S., Brandis, A., Bareia, T., Yofe, I., Jubran, H., and Milo, R. (2013) Spanning high-dimensional expression space using ribosome-binding site combinatorics. *Nucleic Acids Res.*, 41 (9), e98.
- 59** Lee, M.E., Aswani, A., Han, A.S., Tomlin, C.J., and Dueber, J.E. (2013) Expression-level optimization of a multi-enzyme pathway in the absence of a high-throughput assay. *Nucleic Acids Res.*, 41 (22), 10668–10678.
- 60** Moriya, H., Shimizu-Yoshida, Y., and Kitano, H. (2006) *In vivo* robustness analysis of cell division cycle genes in *Saccharomyces cerevisiae*. *PLoS Genet.*, 2 (7), e111.
- 61** Trinh, C.T., Carlson, R., Wlaschin, A., and Srienc, F. (2006) Design, construction and performance of the most efficient biomass producing *E. coli* bacterium. *Metab. Eng.*, 8 (6), 628–638.
- 62** Crabtree, H.G. (1928) The carbohydrate metabolism of certain pathological overgrowths. *Biochem. J.*, 22 (5), 1289–1298.
- 63** Warburg, O., Posener, K., and Negelein, E. (1924) Ueber den Stoffwechsel der Tumoren. *Biochem. Z.*, 152, 319–344.
- 64** Zhuang, K., Vemuri, G.N., and Mahadevan, R. (2011) Economics of membrane occupancy and respiro-fermentation. *Mol. Syst. Biol.*, 7, 500.
- 65** Schuster, S., Pfeiffer, T., and Fell, D. (2008) Is maximization of molar yield in metabolic networks favoured by evolution? *J. Theor. Biol.*, 252, 497–504.
- 66** Li, G.-W., Burkhardt, D., Gross, C., and Weissman, J.S. (2014) Quantifying absolute protein synthesis rates reveals principles underlying allocation of cellular resources. *Cell*, 157, 624–635.
- 67** Pettersson, G. (1989) Effect of evolution on the kinetic properties of enzymes. *Eur. J. Biochem.*, 184 (3), 561–566.
- 68** Pettersson, G. (1992) Evolutionary optimization of the catalytic efficiency of enzymes. *Eur. J. Biochem.*, 206 (1), 289–295.
- 69** Heinrich, R. and Hoffmann, E. (1991) Kinetic parameters of enzymatic reactions in states of maximal activity: an evolutionary approach. *J. Theor. Biol.*, 151 (2), 249–283.
- 70** Wilhelm, T., Hoffmann-Klipp, E., and Heinrich, R. (1994) An evolutionary approach to enzyme kinetics: optimization of ordered mechanisms. *Bull. Math. Biol.*, 56, 65–106.
- 71** Klipp, E. and Heinrich, R. (1994) Evolutionary optimization of enzyme kinetic parameters: effect of constraints. *J. Theor. Biol.*, 171 (3), 309–323.
- 72** Klipp, E. and Heinrich, R. (1999) Competition for enzymes in metabolic pathways: implications for optimal distributions of enzyme concentrations and for the distribution of flux control. *Biosystems*, 54 (1–2), 1–14.
- 73** Klipp, E., Heinrich, R., and Holzhutter, H.G. (2002) Prediction of temporal gene expression. Metabolic optimization by re-distribution of enzyme activities. *Eur. J. Biochem.*, 269 (22), 5406–5413.
- 74** de Hijas-Liste, G.M., Klipp, E., Balsa-Canto, E., and Banga, J.R. (2014) Global dynamic optimization approach to predict activation in metabolic pathways. *BMC Syst. Biol.*, 8, 1.
- 75** Zaslaver, A., Mayo, A.E., Rosenberg, R., Bashkin, P., Sberro, H., Tsalyuk, M. et al. (2004) Just-in-time transcription program in metabolic pathways. *Nat. Genet.*, 36 (5), 486–491.
- 76** Ibarra, R.U., Edwards, J.S., and Palsson, B.O. (2002) *Escherichia coli* K-12 undergoes adaptive evolution to achieve *in silico* predicted optimal growth. *Nature*, 420 (6912), 186–189.
- 77** Darwin, C. (1859) *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life*, John Murray, London.
- 78** Eigen, M. (1971) Molekulare Selbstorganisation und Evolution (Self-organization of matter and the evolution of biological macromolecules). *Naturwissenschaften*, 58 (10), 465–523.
- 79** Eigen, M. and Schuster, P. (1979) *The Hypercycle: A Principle of Natural Self-Organization*, Springer, Berlin.
- 80** Eigen, M. and Schuster, P. (1977) The hypercycle. A principle of natural self-organization. Part A. Emergence of the hypercycle. *Naturwissenschaften*, 64 (11), 541–565.
- 81** Eigen, M. and Schuster, P. (1982) Stages of emerging life: five principles of early organization. *J. Mol. Evol.*, 19 (1), 47–61.
- 82** Eigen, M. and Schuster, P. (1978) The hypercycle. A principle of natural self organization. Part B. The abstract hypercycle. *Naturwissenschaften*, 65 (1), 7–41.
- 83** Kimura, M. (1983) *The Neutral Theory of Molecular Evolution*, Cambridge University Press, Cambridge.
- 84** Kimura, M. (1979) The neutral theory of molecular evolution. *Sci. Am.*, 241 (5), 98–100, 102, 108 passim.
- 85** Kimura, M. and Ota, T. (1971) On the rate of molecular evolution. *J. Mol. Evol.*, 1 (1), 1–17.
- 86** Kimura, M. and Ota, T. (1974) On some principles governing molecular evolution. *Proc. Natl. Acad. Sci. USA*, 71 (7), 2848–2852.
- 87** Kauffman, S.A. and Weinberger, E.D. (1989) The NK model of rugged fitness landscapes and its application to maturation of the immune response. *J. Theor. Biol.*, 141 (2), 211–245.
- 88** Kauffman, S.A. (1991) Antichaos and adaptation. *Sci. Am.*, 265 (2), 78–84.
- 89** Kauffman, S.A. and Macready, W.G. (1995) Search strategies for applied molecular evolution. *J. Theor. Biol.*, 173 (4), 427–440.
- 90** Kauffman, S.A. (1993) *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York.
- 91** Holland, J.H. (1975) *Adaptation in Natural and Artificial Systems*, MIT Press, Boston, MA.
- 92** Domingo, E. and Holland, J.H. (1997) RNA virus mutations and fitness for survival. *Annu. Rev. Microbiol.*, 51, 151–178.
- 93** Holland, J.H. (1992) *Adaptation in Natural and Artificial Systems*, MIT Press, Cambridge, MA.
- 94** Fogel, L.J., Owens, A.J., and Walsh, M.J. (1966) *Artificial Intelligence Through Simulated Evolution*, John Wiley & Sons, Inc., New York.
- 95** Rechenberg, I. (1994) *Evolutionsstrategie '94*, Friedrich Frommann Verlag.
- 96** Koza, J.R., Mydlowec, W., Lanza, G., Yu, J., and Keane, M.A. (2001) Reverse engineering of metabolic pathways from observed data using genetic programming. *Pac. Symp. Biocomput.*, 434–445.
- 97** Schuster, P., Sigmund, K., and Wolff, R. (1978) Dynamical systems under constant organization. I. Topological analysis of a family of non-linear differential equations: a model for catalytic hypercycles. *Bull. Math. Biol.*, 40 (6), 743–769.
- 98** Swetina, J. and Schuster, P. (1982) Self-replication with errors. A model for polynucleotide replication. *Biophys. Chem.*, 16 (4), 329–345.
- 99** Burch, C.L. and Chao, L. (2000) Evolvability of an RNA virus is determined by its mutational neighbourhood. *Nature*, 406 (6796), 625–628.
- 100** Eigen, M. (1971) Molecular self-organization and the early stages of evolution. *Q. Rev. Biophys.*, 4 (2), 149–212.
- 101** Eigen, M., Gardiner, W.C., Jr., and Schuster, P. (1980) Hypercycles and compartments. Compartments assist – but do not replace – hypercyclic organization of early genetic information. *J. Theor. Biol.*, 85 (3), 407–411.

- 102** Sherrington, D. and Kirkpatrick, S. (1975) Solvable model of a spin glass. *Phys. Rev. Lett.*, 35 (26), 1792–1796.
- 103** Sherrington, D. and Kirkpatrick, S. (1978) Infinite-ranged models of spin-glasses. *Phys. Rev. B*, 17, 4384.
- 104** Watson, J.D. and Crick, F.H. (1953) Genetical implications of the structure of deoxyribonucleic acid. *Nature*, 171 (4361), 964–967.
- 105** Watson, J.D. and Crick, F.H. (1953) The structure of DNA. *Cold Spring Harb. Symp. Quant. Biol.*, 18, 123–131.
- 106** Watson, J.D. and Crick, F.H. (1953) Molecular structure of nucleic acids: a structure for deoxyribose nucleic acid. *Nature*, 171 (4356), 737–738.
- 107** Ohno, S. (1970) *Evolution by Gene Duplication*, Springer, Berlin.
- 108** Falster, D.S. and Westoby, M. (2003) Plant height and evolutionary games. *Trends Ecol. Evol.*, 18, 337–343.
- 109** Pfeiffer, T. and Schuster, S. (2005) Game-theoretical approaches to studying the evolution of biochemical systems. *Trends Biochem. Sci.*, 30 (1), 20–25.
- 110** Schuetz, R., Kuepfer, L., and Sauer, U. (2007) Systematic evaluation of objective functions for predicting intracellular fluxes in *Escherichia coli*. *Mol. Syst. Biol.*, 3, 119.
- 111** Schuetz, R., Zamboni, N., Zampieri, M., Heinemann, M., and Sauer, U. (2012) Multidimensional optimality of microbial metabolism. *Science*, 336 (6081), 601–604.
- 112** Turner, P.E. (2005) Cheating viruses and game theory. *Am. Sci.*, 93, 428–435.
- 113** von Neumann, J. and Morgenstern, O. (1944) *Theory of Games and Economic Behavior*, Princeton University Press, Princeton, NJ.
- 114** Smith, J.M. and Price, G.R. (1973) The logic of animal conflict. *Nature*, 246, 15–18.
- 115** Smith, J.M. (1982) *Evolution and the Theory of Games*, Cambridge University Press, Cambridge.
- 116** Nowak, A. and Sigmund, K. (2004) Evolutionary dynamics of biological games. *Science*, 303, 793–799.
- 117** Kerr, B., Riley, M.A., Feldman, M.W., and Bohannan, B.J.M. (2002) Local dispersal promotes biodiversity in a real-life game of rock–paper–scissors. *Nature*, 418, 171–174.
- 118** Czárán, T.L., Hoekstra, R.F., and Pagie, L. (2002) Chemical warfare between microbes promotes biodiversity. *Proc. Natl. Acad. Sci. USA*, 99, 786–790.
- 119** Hutchinson, G.E. (1961) The paradox of the plankton. *Am. Nat.*, 95, 137–145.
- 120** Nowak, M.A. (2006) Five rules for the evolution of cooperation. *Science*, 314 (5805), 1560–1563.
- 121** Chuang, J.S., Rivoire, O., and Leibler, S. (2010) Cooperation and Hamilton's rule in a simple synthetic microbial system. *Mol. Syst. Biol.*, 6, 398.
- 122** Dudley, S.A. and File, A.L. (2007) Kin recognition in an annual plant. *Biol. Lett.*, 4, 435–438.
- 123** Schuster, S., de Figueiredo, L.F., Schroeter, A., and Kaleta, C. (2011) Combining metabolic pathway analysis with evolutionary game theory. Explaining the occurrence of low-yield pathways by an analytic optimization approach. *Biosystems*, 105, 147–153.
- 124** Pfeiffer, T., Schuster, S., and Bonhoeffer, S. (2001) Cooperation and competition in the evolution of ATP-producing pathways. *Science*, 292, 504–507.
- 125** Warburg, O., Posener, K., and Negelein, E. (1924) Ueber den Stoffwechsel der Tumoren. *Biochem. Z.*, 152, 319–344.
- 126** Bachmann, H., Fischlechner, M., Rabbers, I., Barfa, N., Branco dos Santos, F., Molenaar, D., and Teusink, B. (2013) Availability of public goods shapes the evolution of competing metabolic strategies. *Proc. Natl. Acad. Sci. USA*, 110 (35), 14302–14307.
- 127** Pfeiffer, T., Schuster, S., and Bonhoeffer, S. (2001) Cooperation and competition in the evolution of ATP-producing pathways. *Science*, 292, 504–507.

Further Reading

- Optimality in metabolic systems:** Heinrich, R. and Schuster, S. (1998) The modelling of metabolic systems. Structure, control, and optimality. *Biosystems*, 47, 61–77.
- Temporal optimization:** Klipp, E., Heinrich, R., and Holzhütter, H.-G. (2002) Prediction of temporal gene expression. Metabolic optimization by re-distribution of enzyme activities. *Eur. J. Biochem.*, 269 (22), 5406–5413.
- Experimental cost-benefit analysis of Lac operon:** Dekel, E. and Alon, U. (2005) Optimality and evolutionary tuning of the expression level of a protein. *Nature*, 436, 588–692.
- Evolvability:** Kirschner, M. and Gerhart, J. (1998) Evolvability. *Proc. Natl. Acad. Sci. USA*, 95 (15), 8420–8427.
- Flux minimization:** Holzhütter, H.-G. (2004) The principle of flux minimization and its application to estimate stationary fluxes in metabolic networks. *Eur. J. Biochem.*, 271 (14), 2905–2922.
- Multi-objective flux optimization:** Schuetz, R., Zamboni, N., Zampieri, M., Heinemann, M., and Sauer, U. (2012) Multidimensional optimality of microbial metabolism. *Science*, 336 (6081), 601–604.
- Enzyme allocation for maximal growth:** Molenaar, D., van Berlo, R., de Ridder, D., and Teusink, B. (2009) Shifts in growth strategies reflect tradeoffs in cellular economics. *Mol. Syst. Biol.*, 5, 323.
- Metabolic flux and yield in FBA:** Schuster, S., Pfeiffer, T., and Fell, D. (2008) Is maximization of molar yield in metabolic networks favoured by evolution? *J. Theor. Biol.*, 252, 497–504.
- Trade-off between enzyme cost and yield:** Flamholz, A., Noor, E., Bar-Even, A., Lieberman, W., and Milo, R. (2013) Glycolytic strategy as a tradeoff between energy yield and protein cost. *Proc. Natl. Acad. Sci. USA*, 110 (24), 10039–10044.
- Evolutionary game theory:** Smith, J.M. (1982) *Evolution and the Theory of Games*, Cambridge University Press, Cambridge.
- Evolution and self-organization:** Kauffman, S.A. (1993) *Origins of Order: Self-Organization and Selection in Evolution*, Oxford University Press, New York.
- Evolution of cooperation:** Nowak, M.A. (2006) Five rules for the evolution of cooperation. *Science*, 314 (5805), 1560–1563.
- Metabolic flux and yield studied by game theory:** Pfeiffer, T., Schuster, S., and Bonhoeffer, S. (2001) Cooperation and competition in the evolution of ATP-producing pathways. *Science*, 292, 504–507.
- Evolution of yield-efficient behavior:** Bachmann, H., Fischlechner, M., Rabbers, I., Barfa, N., Branco dos Santos, F., Molenaar, D., and Teusink, B. (2013) Availability of public goods shapes the evolution of competing metabolic strategies. *Proc. Natl. Acad. Sci. USA*, 110 (35), 14302–14307.

Models of Biochemical Systems

12

Systems biology attempts to understand structure, function, regulation, or development of biological systems by combining experimental and computational approaches. We must acknowledge that different parts of cellular organization are studied and understood in different ways and to different extent. This is related to diverse experimental techniques that can be used to measure the abundance of metabolites, proteins, mRNA, or other types of compounds. For example, enzyme kinetic measurements are performed for more than a century, while mRNA measurements (e.g., RNAseq or microarray data) or protein measurements (e.g., as mass spectrometry analysis) have been developed more recently. Not all data can be provided with the same accuracy and reproducibility. These and other complications in studying life caused a nonuniform progress in modeling different parts of cellular life. Moreover, the diversity of scientific questions and the availability of computational tools to tackle them provoked the development of very different types of models for different biological processes. Stimulating or modeling of biochemical networks is the recent development to assemble parts lists for various networks, such as genome-scale metabolic reconstructions, compilations of all compounds in signaling networks, lists of all transcription factors, and many more. These can often be found in specialized databases such as Reactome, KEGG, and BioModels (see Chapter 16). In this chapter, we will introduce a number of classical and more recent areas of systems biological research. Specifically, we discuss modeling of metabolic systems, signaling pathways, cell-cycle regulation, and development and differentiation, primarily with ODE systems.

12.1 Metabolic Systems

Summary

Living cells require energy and material for building membranes, storing molecules, replenishing enzymes,

12.1 Metabolic Systems

- Basic Elements of Metabolic Modeling
- Toy Model of Upper Glycolysis
- Threonine Synthesis Pathway Model

12.2 Signaling Pathways

- Function and Structure of Intra- and Intercellular Communication
- Receptor–Ligand Interactions
- Structural Components of Signaling Pathways
- Analysis of Dynamic and Regulatory Features of Signaling Pathways

12.3 The Cell Cycle

- Steps in the Cycle
- Minimal Cascade Model of a Mitotic Oscillator
- Models of Budding Yeast Cell Cycle

12.4 The Aging Process

- Evolution of the Aging Process
- Using Stochastic Simulations to Study Mitochondrial Damage
- Using Delay Differential Equations to Study Mitochondrial Damage

Exercises

References

replication and repair of DNA, movement, and many other processes. Through metabolism cells acquire energy and building blocks and use it to grow and to build new cells. Metabolism is the means by which cells survive and reproduce. Metabolism is the general term for two kinds of reactions: (1) catabolic reactions (breakdown of complex compounds to get energy and building blocks) and (2) anabolic reactions (construction of complex compounds used in cellular functioning). Metabolism is a highly organized process. It involves hundreds or thousands of reactions that are catalyzed by enzymes.

Metabolic networks consist of reactions transforming molecules of one type into molecules of another type. In

modeling terms, the concentrations of the molecules and their rates of change are of special interest. In Chapters 2–4, we explained how to study such networks on three levels of abstraction:

- 1) The network character of metabolism is studied with stoichiometric analysis considering the balance of compound production and degradation.
- 2) Enzyme kinetics investigates the dynamic properties of the individual reactions in isolation.
- 3) Metabolic control analysis quantifies the effect of perturbations in the network employing the individual dynamics of concentration changes and their integration in the network.

Here, we will illustrate the theoretical concepts by applying them to a number of examples. We will specifically discuss cellular energy metabolism with focus on glycolysis and the threonine pathway as an example of amino acid synthesis. You may find the complete models and many other models also in modeling databases such as BioModels or JWS online [1,2].

12.1.1 Basic Elements of Metabolic Modeling

Metabolic networks are on the one side defined by the enzymes converting substrates into products in a reversible or irreversible manner. Without enzymes, those reactions are essentially impossible or too slow. On the other side, the networks are characterized by the metabolites that are converted by the various enzymes. Biochemical studies have revealed a number of important pathways including catabolic pathways and pathways of the energy metabolism, such as glycolysis, the pentose-phosphate pathway, the tricarboxylic acid (TCA) cycle, and oxidative phosphorylation, and anabolic pathways including gluconeogenesis, amino acid synthesis pathways, synthesis of fatty acids, synthesis of nucleic acids, and synthesis and degradation of more complex molecules. Databases such as the Kyoto Encyclopedia of Genes and Genomes Pathway (KEGG, <http://www.genome.jp/kegg/pathway.html>) provide a comprehensive overview of the composition of pathways in various organisms [3].

Here, we will focus on their characteristics that are essential for modeling. Chapter 2 provides a summary of the first steps to build a model. First, we sketch the metabolites and the converting reactions in a cartoon to get an overview and an intuitive understanding (Figure 12.1). Based on that cartoon and on further information, we set the limits of our model. To this end, we consider what kind of question we want to answer with our model, what information in terms of qualitative and quantitative

data is available, and how can we make the model as simple as possible but as comprehensive as necessary. Then, for every compound, which is part of the system, we formulate the balance equations (see also Section 3.1) summing up all reactions that produce the compound (with a positive sign) and all reactions that degrade the compound (with a negative sign). At this stage, the model is suited for a network analysis, such as the stoichiometric analysis (Section 3.1) or, with some additional information, flux balance analysis (Section 3.2). In order to study the dynamics of the system, we must add kinetic descriptions to the individual reactions (Chapter 4). Keep in mind that the reaction kinetics may depend on

- the concentrations of substrates and products (in Figure 12.1: G1P and G6P),
- specific parameters such as K_M -values,
- the amount and activity of the catalyzing enzyme (here hidden in the V_{max} values, see Chapter 4), and
- the activity of modifiers, which are not shown in the example in Figure 12.1.

In the following, we will discuss in more detail models for three example pathways: the upper glycolysis, the full glycolysis, and the threonine synthesis.

12.1.2 Toy Model of Upper Glycolysis

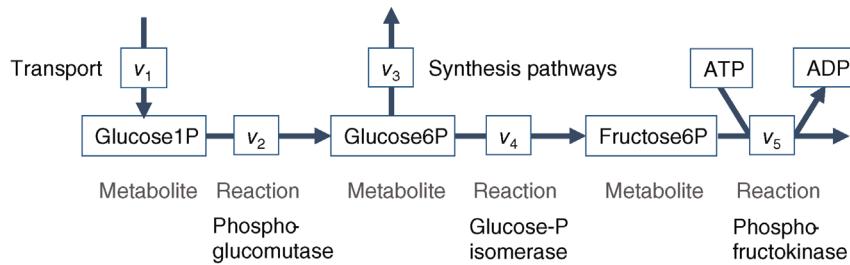
A first model of the upper part of glycolysis is depicted in Figure 12.2. It comprises six reactions and six metabolites. Note that we neglect the formation of phosphate P_i here.

The corresponding ODE system reads

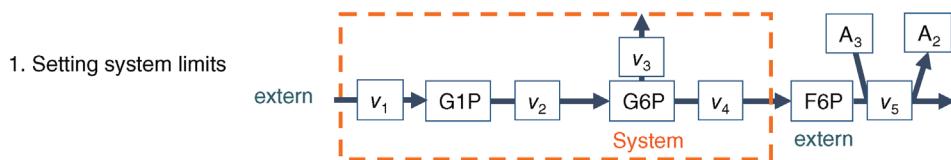
$$\begin{aligned} \frac{d}{dt} \text{Glucose} &= v_1 - v_2, \\ \frac{d}{dt} \text{Gluc6P} &= v_2 - v_3, \\ \frac{d}{dt} \text{Fruc6P} &= v_3 - v_4 + v_5, \\ \frac{d}{dt} \text{Fruc1,6P}_2 &= v_4 - v_5 - v_6, \\ \frac{d}{dt} \text{ATP} &= -\frac{d}{dt} \text{ADP} = -v_2 - v_4 + v_7. \end{aligned} \quad (12.1)$$

With mass action kinetics, the rate equations read $v_1 = \text{constant} = k_1$, $v_2 = k_2 \cdot \text{Glucose} \cdot \text{ATP}$, $v_3 = k_3 \cdot \text{Gluc6P} \cdot k_{-3} \cdot \text{Fruc6P}$, $v_4 = k_4 \cdot \text{Fruc6P} \cdot \text{ATP}$, $v_5 = k_5 \cdot \text{Fruc1,6P}_2$, $v_6 = k_6 \cdot \text{Fruc1,6P}_2$, and $v_7 = k_7 \cdot \text{ADP}$. Given the values of the parameters $k_i, i = 0, \dots, 7$ and the initial concentrations, one may simulate the time behavior of the network as depicted in Figure 12.3.

(a) Basic elements of metabolic networks



(b) Design of structured dynamic models



2. Balancing

$$\frac{d}{dt}G1P = v_1 - v_2 \quad \frac{d}{dt}G6P = v_2 - v_3 - v_4$$

3. Assignment of Kinetics

$$v_1 = \text{const.} \quad v_2 = \frac{\frac{V_{\max 2}}{K_{M2,G1P}} G1P - \frac{V_{\max 2}}{K_{M2,G6P}} G6P}{1 + \frac{G1P}{K_{M2,G1P}} + \frac{G6P}{K_{M2,G6P}}} \quad v_3 = k_3 \cdot G6P \quad v_4 = \frac{V_{\max,4} \cdot G6P}{K_{M,4} + G6P}$$

Figure 12.1 Designing metabolic models. (a) Basic elements of metabolic networks comprise metabolites and the connecting reactions. The reactions can be specific for one enzyme (e.g., the phosphoglucomutase) or they can lump several processes such as transport or branch to synthesis. (b) Basic steps for designing structured dynamic models: first, one has to choose the system of interest and its limits. Second, all metabolite changes have to be balanced according to the contribution reactions (Chapter 3). Next, all reactions are described with kinetic expressions. Note that the shown expressions are a reasonable choice; other expressions can be appropriate depending on the available knowledge and the purpose of the model.

We see starting with zero concentrations of all hexoses (here, glucose, fructose, and their phosphates), that they accumulate until production and degradation are balanced. They approach a steady state. ATP rises and decreases to the same extent as ADP decreases and rises, and their total remains constant. This is due to the conservation of adenine nucleotides, which could be revealed by stoichiometric analysis (Section 3.1).

For this upper glycolysis model, the concentration vector is $\mathbf{S} = (\text{Glucose}, \text{Gluc6P}, \text{Fruc6P}, \text{Fruc1,6P}_2, \text{ATP}, \text{ADP})^T$, the vector of reaction rates is $\mathbf{v} = (v_1, v_2, \dots, v_7)^T$,

and the stoichiometric matrix reads

$$\mathbf{N} = \begin{pmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & -1 & 0 \\ 0 & -1 & 0 & -1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & -1 \end{pmatrix}. \quad (12.2)$$

It comprises $r = 7$ reactions and has $\text{Rank}(\mathbf{N}) = 5$. Thus, the kernel matrix (see Chapter 3) has two linear

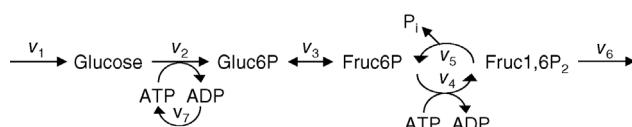


Figure 12.2 Toy model of the upper glycolysis. The model involves the reactions glucose uptake (v_1), the phosphorylation of glucose under conversion of ATP to ADP by the enzyme hexokinase (v_2), intramolecular rearrangements by the enzyme phosphoglucomutase (v_3), a second phosphorylation (and ATP/ADP conversion) by phosphofructokinase (v_4), dephosphorylation without involvement of ATP/ADP by fructose-bisphosphatase (v_5), and splitting of the hexose (6-C-sugar) into two trioses (3-C-sugars) by aldolase (v_6). Abbreviations: Gluc-6P – glucose-6-phosphate, Fruc-6P – fructose-6-phosphate, Fruc-1,6P₂ – fructose-1,6-bisphosphate.

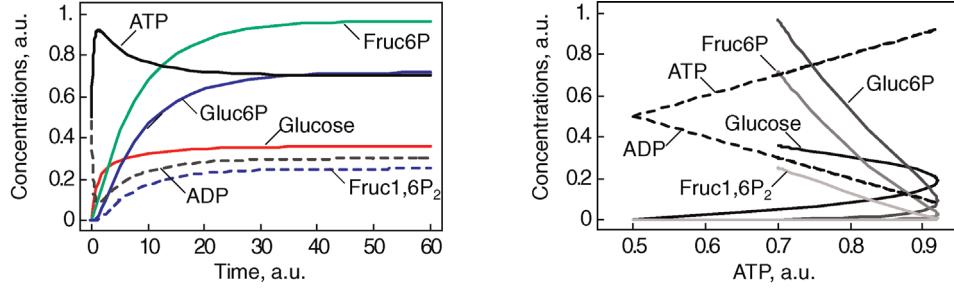


Figure 12.3 Dynamic behavior of the upper glycolysis model (Figure 12.2 and Eq. (12.1)). Initial conditions at $t=0$ $\text{Glucose}(0)=\text{Gluc}-6\text{P}(0)=\text{Fruc}-6\text{P}(0)=\text{Fruc}-1,6\text{P}_2(0)=0$ and $\text{ATP}(0)=\text{ADP}(0)=0.5$ (arbitrary units). Parameters: $k_1=0.25$, $k_2=1$, $k_3=1$, $k_{-3}=1$, $k_4=1$, $k_5=1$, $k_6=1$, and $k_7=2.5$. The steady-state concentrations are $\text{Glucose}^{ss}=0.357$, $\text{Gluc}-6\text{P}^{ss}=0.964$, $\text{Fruc}-6\text{P}^{ss}=0.714$, $\text{Fruc}-1,6\text{P}_2^{ss}=0.25$, and $\text{ATP}^{ss}=0.7$, and $\text{ADP}^{ss}=0.2$. The steady-state fluxes are $J_1=J_2=J_3=J_5=J_6=0.25$, $J_4=0.5$, and $J_7=0.75$. (a) Time course plots (concentration versus time), (b) phase plane plot (concentrations versus concentration of ATP with varying time); all curves start at $\text{ATP}=0.5$ for $t=0$.

independent columns. A possible representation is

$$\mathbf{K} = \begin{pmatrix} \mathbf{k}_1 & \mathbf{k}_2 \end{pmatrix} \text{ with} \quad (12.3)$$

$$\mathbf{k}_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \end{pmatrix} \quad \text{and} \quad \mathbf{k}_2 = \begin{pmatrix} 1 \\ 1 \\ 1 \\ -1 \\ -2 \\ 1 \\ 0 \end{pmatrix},$$

Figure 12.4 shows the flux and concentration control coefficients (see Section 4.2) for the model of upper

glycolysis in gray scale (see scaling bar). Reaction v_1 has a flux control of 1 over all steady-state fluxes, reactions v_2 , v_3 , v_4 , and v_7 have no control over fluxes; they are determined by v_1 . Reactions v_5 and v_6 have positive or negative control over J_4 , J_5 , and J_7 , respectively, since they control the turnover of fructose phosphates.

The concentration control shows a more interesting pattern. As a rule of thumb, it holds that producing reactions have a positive control and degrading reactions have a negative control, such as v_1 and v_2 for glucose. But also distant reactions can exert concentration control, such as v_4 to v_6 over Gluc6P.

More comprehensive models of glycolysis can be used to study details of the dynamics, such as the occurrence

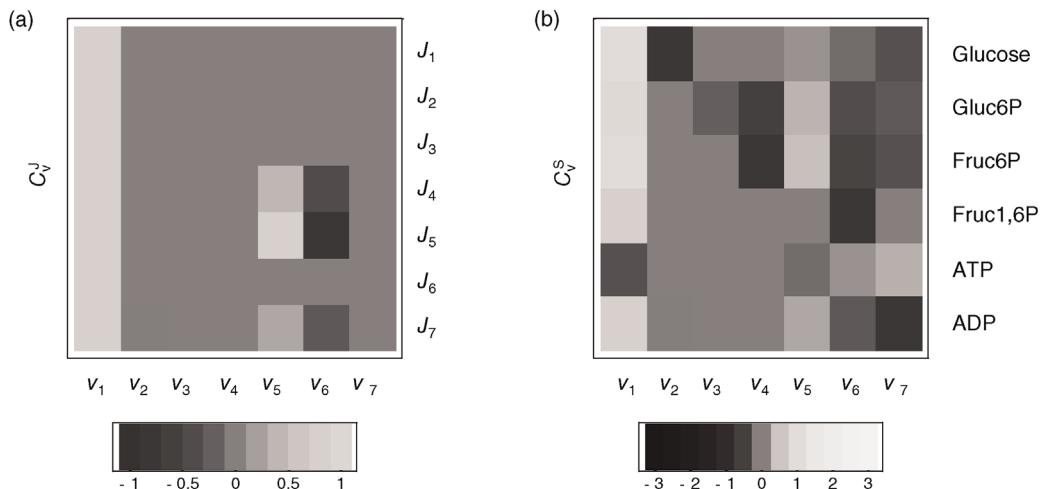


Figure 12.4 Flux and concentration control coefficients for the glycolysis model given by the equation system (12.1) with the parameters given in the legend to Figure 12.3. Values of the coefficients are indicated in gray scale: gray means zero control, white or light gray indicates positive control, and dark gray or black negative control, respectively.

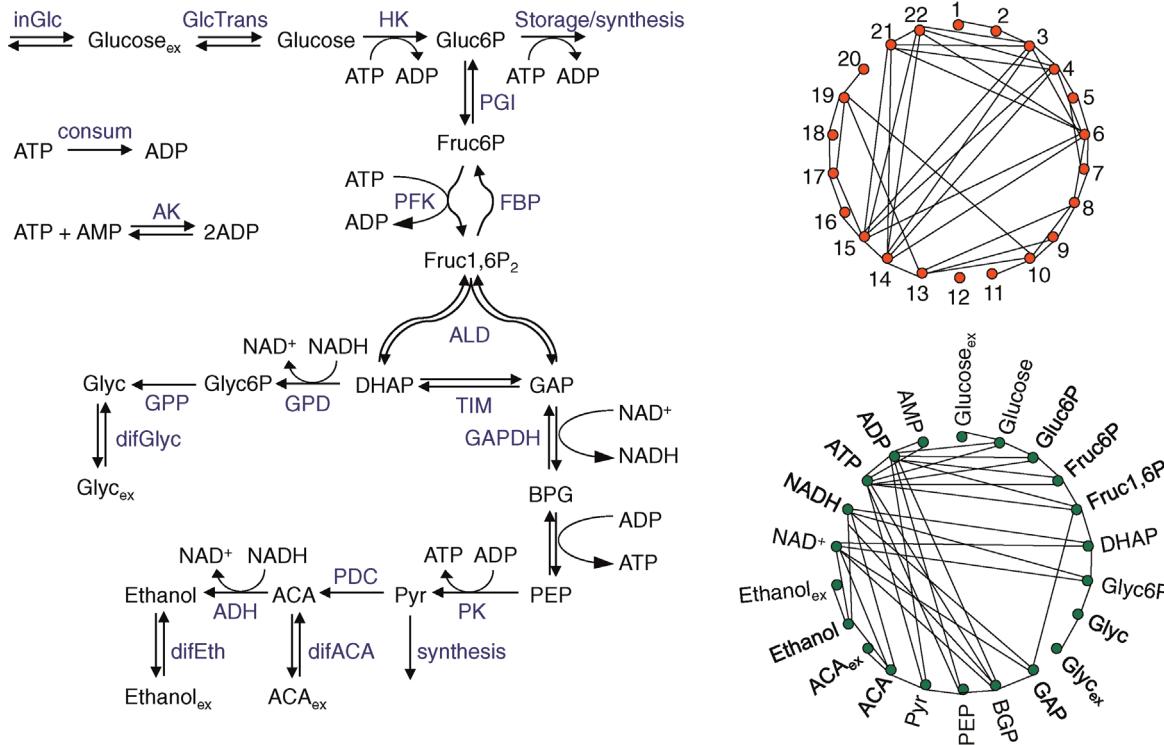


Figure 12.5 Full glycolysis models. (a) Main reactions and metabolites. 1-inGlc – influx of glucose, 2-GlcTrans – transport of glucose, 3-HK – hexokinase, 4-storage/synthesis, 5-PGI – phosphoglucoisomerase, 6-PFK – phosphofructokinase, 7-FBP – fructosbisphosphatase, 8-ALD – aldolase, 9-TIM – triosephosphate isomerase, 10-GPD – NAD-dependent glyceral-3-phosphate dehydrogenase, 11-GPP – glycerol-3-phosphate phosphatase, 12-difGlyc – (facilitated) diffusion of glycerol over membrane, 13-GAPDH – glyceraldehyde phosphate dehydrogenase, 14-phosphoglycerate kinase + phosphoglycerate mutase + enolase, 15-PK pyruvate kinase, 16-synthesis, 17-PDC – pyruvate decarboxylase, 18-diffusion of acetate, 19-ADH – alcohol dehydrogenase, 20-difEth – ethanol diffusion, 21-ATP consumption, 22-AK – adenylate kinase.

(b) Network of reactions connected by common metabolites, (c) network of metabolites connected by common reactions.

of oscillations or the effect of perturbations. Examples are the models of Hynne *et al.* [4] or the Reuss group [5,6] or a model based on genome-scale reconstructions [3]. An overview of the most important reactions in glycolysis is given in Figure 12.5.

12.1.3 Threonine Synthesis Pathway Model

Threonine is an amino acid and it is essential for birds and mammals. The synthesis pathway from aspartate involves five steps. It is known for a long time and has attracted some interest with respect to its economic industrial production for a variety of uses. The kinetics of all the five enzymes from *E. coli* have been studied extensively, the complete genome sequence of this organism is now known and there is an extensive range of genetic tools available. The intensive study and the availability of kinetic information make it a good

example for metabolic modeling of the pathway (Figure 12.6).

The reaction system can be described with the following set of differential equations:

$$\begin{aligned}
 \frac{d}{dt} Asp &= -\nu_{AK\text{ I}} - \nu_{AK\text{ III}}, \\
 \frac{d}{dt} AspP &= \nu_{AK\text{ I}} + \nu_{AK\text{ III}} - \nu_{ASD}, \\
 \frac{d}{dt} ASA &= \nu_{ASD} - \nu_{HDH}, \\
 \frac{d}{dt} HS &= \nu_{HDH} - \nu_{HK}, \\
 \frac{d}{dt} HSP &= \nu_{HK} - \nu_{TS}, \\
 \frac{d}{dt} Thr &= \nu_{TS},
 \end{aligned} \tag{12.4}$$

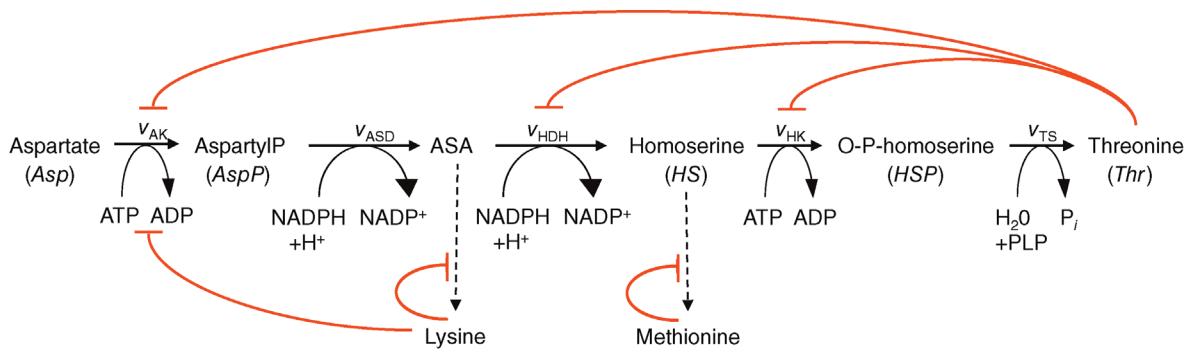


Figure 12.6 Model of the threonine pathway. Aspartate is converted into threonine in five steps. Threonine exerts negative feedback on its producing reactions. The pathway consumes ATP and NADPH.

with

$$v_{AK\ I} = \frac{V_{AK\ I} \cdot \left(Asp \cdot ATP - \frac{AspP \cdot ADP}{K_{eq,AK}} \right)}{\left(K_{Asp,AK\ I} \cdot \frac{1 + \left(\frac{Thr}{K_{iThr,AK\ I}} \right)^{h_1}}{1 + \left(\frac{Thr}{\alpha_{AK\ I} \cdot K_{iThr,AK\ I}} \right)^{h_1}} + AspP \cdot \frac{K_{Asp,AK\ I}}{K_{AspP,AK\ I}} + Asp \right) \cdot \left(K_{ATP,AK\ I} \cdot \left(1 + \frac{ADP}{K_{ADP,AK\ I}} \right) + ATP \right)}$$

$$K_{eq,AK} = 6.4 \times 10^{-4}, K_{Asp,AK\ I} = 0.97 \pm 0.48 \text{ mM}, K_{ATP,AK\ I} = 0.98 \pm 0.5 \text{ mM}$$

$$K_{AspP,AK\ I} = 0.017 \pm 0.004 \text{ mM}, K_{ADP,AK\ I} = 0.25 \text{ mM}, K_{iThr,AK\ I} = 0.167 \pm 0.003 \text{ mM},$$

$$h_1 = 4.09 \pm 0.26, \alpha_{AK\ I} = 2.47 \pm 0.17,$$

$$v_{AK\ III} = \frac{V_{AK\ III} \cdot \left(Asp \cdot ATP - \frac{AspP \cdot ADP}{K_{eq,AK}} \right)}{\left(1 + \left(\frac{Lys}{K_{iLys}} \right)^{h_{Lys}} \right) \left(K_{Asp,AK\ III} \left(1 + \frac{AspP}{K_{AspP,AK\ III}} \right) + Asp \right) \cdot \left(K_{ATP,AK\ III} \left(1 + \frac{ADP}{K_{ADP,AK\ III}} \right) + ATP \right)}$$

$$K_{eq,AK} = 6.4 \times 10^{-4}, K_{Asp,AK\ III} = 0.32 \pm 0.08 \text{ mM}, K_{ATP,AK\ III} = 0.22 \pm 0.02 \text{ mM}$$

$$K_{AspP,AK\ III} = 0.017 \pm 0.004 \text{ mM}, K_{ADP,AK\ III} = 0.25 \text{ mM}, K_{iLys} = 0.391 \pm 0.08 \text{ mM},$$

$$h_{Lys} = 2.8 \pm 1.4,$$

$$v_{ASD} = \frac{V_{ASD} \cdot \left(AspP \cdot NADPH - \frac{ASA \cdot NADP^+ \cdot P_i}{K_{eq,ASD}} \right)}{\left(K_{AspP,ASD} \left(1 + \frac{ASA}{K_{ASA,ASD}} \right) \cdot \left(1 + \frac{P_i}{K_{P_i}} \right) + AspP \right) \cdot \left(K_{NADPH} \left(1 + \frac{NADP^+}{K_{NADP^+}} \right) + NADPH \right)}$$

$$K_{eq,ASD} = 2.84 \times 10^5, K_{AspP,ASD} = 0.022 \pm 0.001 \text{ mM}, K_{NADPH,ASD} = 0.029 \pm 0.002 \text{ mM}$$

$$K_{ASA,ASD} = 0.11 \pm 0.008 \text{ mM}, K_{NADP^+,ASD} = 0.144 \pm 0.02 \text{ mM}, K_{P_i} = 10.2 \pm 1.4 \text{ mM},$$

$$v_{HDH} = \frac{V_{HDH} \cdot \left(ASA \cdot NADPH - \frac{HS \cdot NADP^+}{K_{eq,HDH}} \right)}{\left(\frac{1 + \left(\frac{Thr}{K_{iThr,2}} \right)^{h_2}}{1 + \left(\frac{Thr}{\alpha_2 \cdot K_{iThr,2}} \right)^{h_2}} \right) \left(K_{ASA,HDH} \left(1 + \frac{HS}{K_{HS,HDH}} \right) + ASA \right) \cdot \left(K_{NADPH,HDH} \left(1 + \frac{NADP^+}{K_{NADP^+,AK\ III}} \right) + NADPH \right)}$$

$$K_{eq,HDH} = 1 \times 10^{11} \text{ M}^{-1}, K_{ASA,HDH} = 0.24 \pm 0.03 \text{ mM}, K_{NADPH,HDH} = 0.037 \pm 0.006 \text{ mM}$$

$$K_{HS,HDH} = 3.39 \pm 0.33 \text{ mM}, K_{NADP^+,HDH} = 0.067 \pm 0.006 \text{ mM}, K_{iThr,2} = 0.097 \text{ mM}, h_2 = 1.41,$$

$$\alpha_2 = 3.93,$$

$$v_{HK} = \frac{V_{HK} \cdot hs \cdot ATP}{\left(K_{HS,HK} \left(1 + \frac{ATP}{K_{iATP,HK}} \right) \cdot \left(1 + \frac{Thr}{K_{iThr,HK}} \right) + hs \right) \cdot \left(K_{ATP,HK} \left(1 + \frac{hs}{K_{iHS,HK}} \right) + ATP \right) \cdot \left(1 + \frac{Lys}{K_{iLys,HK}} \right)}$$

$$K_{HS,HK} = 0.11 \text{ mM}, K_{ATP,HK} = 0.072 \text{ mM}, K_{iThr,HK} = 1.09 \text{ mM}, K_{iLys,HK} = 9.45 \text{ mM},$$

$$K_{iHS,HK} = 4.7 \text{ mM}, K_{iATP,HK} = 4.35 \text{ mM}$$

$$v_{TS} = \frac{V_{TS} \cdot HSP}{K_{HSP,TS} + HSP}.$$

$$K_{HSP,TS} = 0.31 \pm 0.03 \text{ mM}$$

This system has no nontrivial steady state, that is, no steady state with nonzero flux, since aspartate is always degraded, while threonine is only produced. The same imbalance holds for the couples ATP/ADP and NADPH+H⁺/NADP⁺. The dynamics is shown in Figure 12.7.

Threonine exerts feedback inhibition on the pathway producing it. This is illustrated in Figure 12.8. The higher the threonine concentration, the lower the rates of the inhibited reactions. The effect is that production of threonine is downregulated as long as its level is sufficient, thereby saving aspartate and energy.

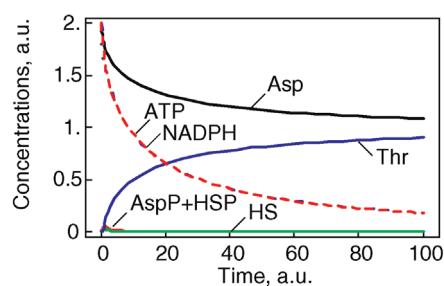


Figure 12.7 Dynamics of the threonine pathway model. The parameters are given in the text.

12.2 Signaling Pathways

Summary

Throughout intercellular communication or cellular stress response, the cell senses extracellular signals. They are converted into intracellular signals and sequences of reactions. Different external cues or events may stimulate signaling. Typical signals are growth hormones, pheromones, heat, cold, light, osmotic pressure, oxidative stress, and appearance or concentration change of substances like glucose, K⁺, Ca²⁺, or cAMP.

In this chapter, we introduce the basic logic of signaling systems in conveying internal and external signals to cellular responses. Modeling of the dynamic behavior of signaling pathways is often not straightforward. Despite increasing information about signal molecules and their interaction, which is frequently available in databases, the knowledge about pathway components and their interaction is still limited and incomplete. The interpretation of experimental data is context- and knowledge-dependent. Furthermore, the effect of a signal often changes the state of the whole cell – this implies difficulties for determination of system limits. But in many cases, we may apply the same tools as introduced in Chapters 2–4.

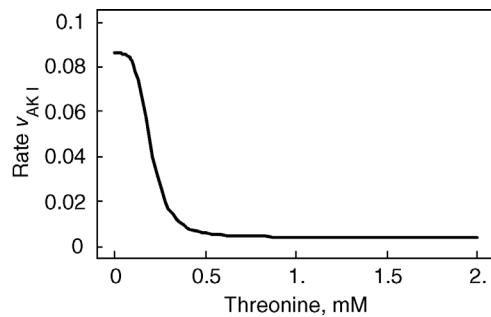
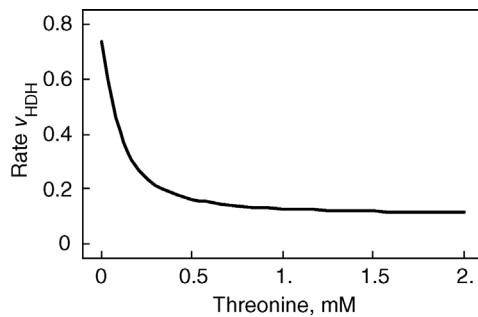


Figure 12.8 Effect of feedback inhibition in the model depicted in Figure 12.6. Threonine is a product of the pathway but exhibits negative feedback on its producing reactions. Consequently, their rates decrease with increasing threonine concentration leading to a homeostasis.

Here, we portrait a number of typical building blocks of such signaling pathways, such as G proteins or MAPK cascades, and discuss their properties and modeling approaches. Finally, we present a number of analysis tools for models of signaling networks.

12.2.1 Function and Structure of Intra- and Inter-cellular Communication

On the molecular level, signaling involves the same type of processes as metabolism: production or degradation of substances, molecular modifications (mainly phosphorylation, also methylation, acetylation), localization of compounds to specific cell compartments, and activation or inhibition of reactions. From a modeling point of view, there are some important differences between signaling and metabolism. (i) Signaling pathways serve for information processing and transfer of information, while metabolism provides mainly mass transfer. (ii) The metabolic network is determined by the present set of enzymes catalyzing the reactions. Signaling pathways involve compounds of different type; they may form highly organized complexes, and may dynamically assemble upon occurrence of the signal. (iii) The quantity of converted material is high in metabolism (amounts are usually given in concentrations in the order of μM or mM) compared to the number of molecules involved in signaling processes (in yeast or *E. coli* cells, typical abundance of proteins in signal cascades is in the order of $10\text{--}10^4$ molecules per cell). (iv) The different amounts of components have an effect on the concentration ratio of catalysts and substrates. In metabolism, this ratio is usually low, that is, the enzyme concentration is much lower than the substrate concentration, which gives rise to the quasi-steady state assumption used in Michaelis–Menten kinetics (Section 4.1). In signaling processes, amounts of catalysts and their substrates are frequently in the same order of magnitude.

Cells have a broad spectrum of receiving and processing signals; not all of them can be considered here. A typical sequence of events in signaling pathways is shown in Figure 12.9 and proceeds as follows.

The signal – a substance acting as ligand, hormones, growth factors or a physical stimulus, oxidative stress, and osmotic stress – approaches the cell surface and is perceived by a transmembrane receptor. The receptor changes its state from susceptible to active and triggers subsequent processes within the cell. The active receptor stimulates an internal signaling cascade.

In order to distinguish between fast and slow processes and to characterize the main location of events, we discriminate between responses on levels I–IV. Level I response comprises all processes that take place at the

membrane due to receptor activation. These processes may include the recruitment of a number of proteins to the vicinity of the receptor that were previously located either at the membrane or in the cytosol. These proteins may form large heteromeric complexes and modify each other and the receptor and they form a local environment of information transmission. They may also serve as anchor for cell structures such as actin cables.

Level II response includes all downstream interaction of signaling molecules in the cytoplasm. There occur different types of protein–protein interactions such as complex formation, phosphorylation and dephosphorylation, or phosphotransfer, as well as targeting for degradation. The effects include both forward signaling as well as positive and negative feedback. Also crosstalk between signaling pathways that are primarily considered responsible for transmission of specific signals is observed. Note that the concept of crosstalk is strongly related to the perception of signaling by the investigator. If we consider separate signaling pathways that transmit the information about a specific cue, we can also consider their crosstalk, that is, interactions between components of diverse signaling pathways that lead to a divergence of the information to other targets than those of the specific pathway (measures for crosstalk are discussed below). If we consider all signaling processes as one large signaling network, then we simply observe various interactions in this network.

Level III response comprises all direct effects of active signaling molecules on other cellular networks such as phosphorylation of cell-cycle components or metabolic enzymes. These effects proceed on fast time scales and may lead to immediate cellular responses.

Level IV response is comparatively slow. The sequence of state changes within the signaling pathway crosses the nuclear membrane. Eventually, transcription factors are activated or deactivated. Such a transcription factor changes its binding properties to regulatory regions on the DNA upstream of a set of genes; the transcription rate of these genes is altered (typically increased). The either newly produced proteins or the changes in protein concentration cause an adaptation response of the cell to the signal. Also on level IV, signaling pathways are regulated by a number of control mechanisms including feedback and feed-forward modulation.

This is somehow the typical picture; many pathways may work in completely different manner. As example, an overview on signaling pathways that are stimulated in yeast stress response is given in Figure 12.10. More extended versions of the yeast signaling network with focus on the precise types of interactions can be found in Ref. [7].

With respect to the activating or inhibiting effect of an external cue on cellular response features, signaling pathways may exhibit different types of logical

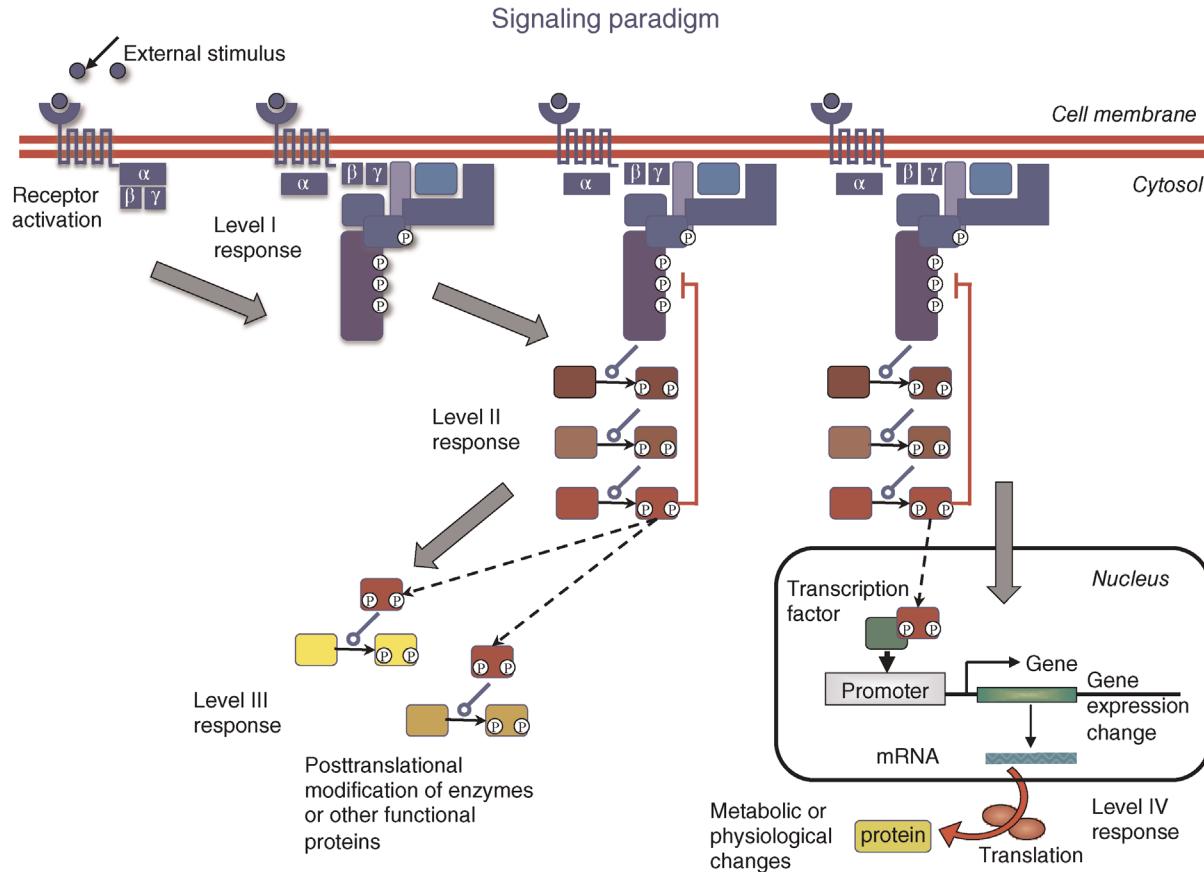


Figure 12.9 Visualization of the signaling paradigm. The receptor is stimulated by a ligand or another kind of signal and becomes active. The active receptor initiates the internal signaling cascade: Level I response comprises processes at the cellular membrane such as receptor phosphorylation, G protein cycle, recruitment of specific proteins, establishment of transient interactions, and modifications. Level II response includes all interactions and mutual modifications of signaling proteins in the cytosol. They lead primarily to the activation of downstream components, but can also exert positive or negative feedback within the current signaling cascade or crosstalk to other areas of the signaling network. Level III response denotes immediate actions of signaling molecules on functional proteins such as metabolic enzymes or cell-cycle proteins leading to immediate regulatory effects. These interactions can be fast, i.e. within a few minutes in yeast. For level IV response, an activated protein enters the nucleus, transcription factors are activated or deactivated. The transcription factors regulate the transcription rate of a set of genes. The absolute amount or the relative changes in protein concentrations alter the state of the cell and trigger the long-term response to the signal.

behavior. Positive and negative logic of signaling systems can be found:

- 1) Before activation, the signaling pathway is OFF (or the activity of all components is on a basic level). Ligand binding or physical cue leads to activation or switch-on of the pathway's components. Examples: MAPK pathway, G proteins.
- 2) Before activation, some components of the signaling pathway are ON or have high activity or are produced and degraded with high rates. Ligand binding or a physical cue reduces the activity or interrupts the production process. Examples: Phosphorelay system (constant consumption and relay of ATP-derived phosphate group is interrupted by osmotic stress), Wnt signaling pathway (constant production and

destruction of β-catenin is modified as a consequence of Wnt binding through interruption of destruction and, hence, accumulation of β-catenin).

These examples are explained below in more detail along with the general structure of signal pathway building blocks (Section 12.2.3).

12.2.2 Receptor–Ligand Interactions

Many receptors are transmembrane proteins. They receive the signal and transmit it. Upon signal sensing they change their conformation. In the active form, they are able to initiate a downstream process within the cell (Figure 12.11).

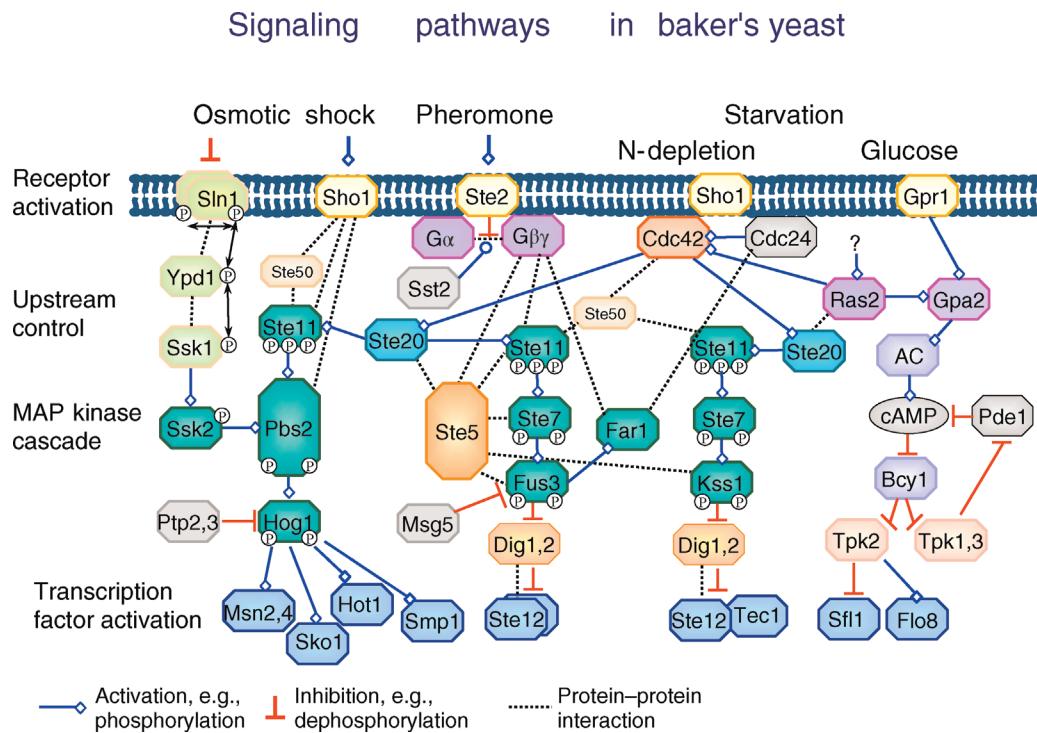


Figure 12.10 Overview of signaling pathways in yeast: the HOG pathway is activated by osmotic shock, the pheromone pathway is activated by a pheromone from cells of opposite mating type, and the filamentous growth pathway is stimulated by starvation conditions. In each case, the signal interacts with the receptor. The receptor activates a cascade of intracellular processes including complex formations, phosphorylations, and transport steps. A MAP kinase cascade is a particular part of many signaling pathways. Eventually, transcription factors are activated that regulate the expression of a set of genes. Beside the indicated connections further interactions of components are possible. For example, crosstalk may occur, that is, the activation of the downstream part of one pathway by a component of another pathway. This is supported by the frequent incidence of some proteins like Ste11 in the scheme.

The simplest concept of the interaction between receptor R and ligand L is reversible binding to form the active complex LR:



The dissociation constant is calculated as

$$K_D = \frac{L \cdot R}{LR}. \quad (12.6)$$

Typical values for K_D are $10^{-12} \text{ M} \dots 10^{-6} \text{ M}$.

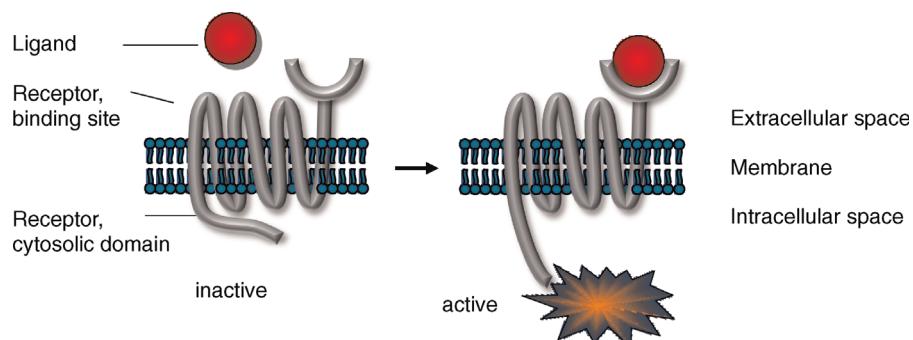


Figure 12.11 Schematic representation of receptor activation: the binding of the ligand at the external side leads to conformational changes at the internal side allow for new interactions or further modifications.

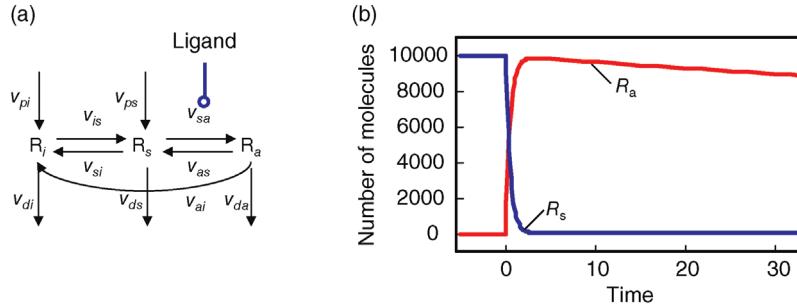


Figure 12.12 Receptor activation by ligand. (a) Schematic representation: L – ligand, R_i – inactive receptor, R_s – susceptible receptor, R_a – active receptor. v_{p*} – production steps, v_{d*} – degradation steps, other steps – transition between inactive, susceptible, and active state of receptor. (b) Time course of active (red line) and susceptible (blue line) receptor after stimulation with $1 \mu\text{M} \alpha$ -factor at $t = 0$. The total number of receptors is 10.000. The concentration of the active receptor increases immediately and then declines slowly, while the susceptible receptor is effectively reduced to zero.

Cells have the ability to regulate the number and the activity of specific receptors, for example in order to weaken the signal transmission during long-term stimulation. Balancing production and degradation regulates the number of receptors. Phosphorylation of serine, threonine or tyrosine residues of the cytosolic domain by protein kinases mainly regulates the activity.

Hence, a more realistic scenario for ligand–receptor interaction is depicted in Figure 12.12.

We assume that the receptor may be present in an inactive state R_i or in a susceptible state R_s . The susceptible form can interact with the ligand to form the active state R_a . The inactive or the susceptible forms are produced from precursors (v_{pi}, v_{ps}), all three forms may be degraded (v_{di}, v_{ds}, v_{da}). The rates of production and degradation processes as well as the equilibria between the different states are influenced by the cell state, for example, by the cell-cycle stage. In general, the dynamics of this scenario can be described by the following set of differential equations:

$$\begin{aligned}\frac{d}{dt}R_i &= v_{pi} - v_{di} - v_{is} + v_{si} + v_{ai}, \\ \frac{d}{dt}R_s &= v_{ps} - v_{ds} + v_{is} - v_{si} - v_{sa} + v_{as}, \\ \frac{d}{dt}R_a &= -v_{da} + v_{sa} - v_{as} - v_{ai}.\end{aligned}\quad (12.7)$$

For the production terms, we may either assume constant values or (as mentioned above) rates that depend on the actual cell state. The degradation terms might be assumed to be linearly dependent on the concentration of their substrates ($v_{d*} = k_{d*} \cdot R_*$). This may also be a first guess for the state changes of the receptor (e.g., $v_{is} = k_{is} \cdot R_i$). The receptor activation is dependent on the ligand concentration (or any other value related to the signal). A linear approximation of the respective rate is $v_{sa} = k_{sa} \cdot R_s \cdot L$. If the receptor is a dimer or oligomer, it

might be sensible to include this information into the rate expression as $v_{sa} = k_{sa} \cdot R_s \cdot \frac{K_b^n \cdot L^n}{1+K_b^n \cdot L^n}$, where K_b denotes the binding constant to the monomer and n denotes the Hill coefficient (Eq. (4.44)).

Example 12.1

An experimentally confirmed example for the activation of receptor and G protein of the pheromone pathway has been presented by Yi *et al.* [8] for the binding of the pheromone α -factor to the receptor Ste2 in yeast. Concerning the receptor activation dynamics, they report a susceptible and an active form of the receptor, but no inactive form ($R_i = 0$, $v_{si} = v_{is} = 0$). The remaining rates are determined as follows:

$$\begin{aligned}v_{ps} &= k_{ps}, \\ v_{ds} &= k_{ds} \cdot R_s, \\ v_{da} &= k_{da} \cdot R_a, \\ v_{sa} &= k_{sa} \cdot R_s \cdot L, \\ v_{as} &= k_{as} \cdot R_a,\end{aligned}\quad (12.8)$$

with the following values for the rate constants: $k_{ps} = 4(\text{molecules per cell}) \text{s}^{-1}$, $k_{ds} = 4 \times 10^{-4} \text{s}^{-1}$, $k_{da} = 4 \times 10^{-3} \text{s}^{-1}$, $k_{sa} = 2 \times 10^6 \text{ M}^{-1} \text{s}^{-1}$, and $k_{as} = 1 \times 10^{-2} \text{s}^{-1}$.

The time course of receptor activation is depicted in Figure 12.12b.

12.2.3 Structural Components of Signaling Pathways

Signaling pathways constitute often highly complex networks, but it has been observed that they are frequently composed of typical building blocks. These components include Ras proteins, G protein cycles, phosphorelay

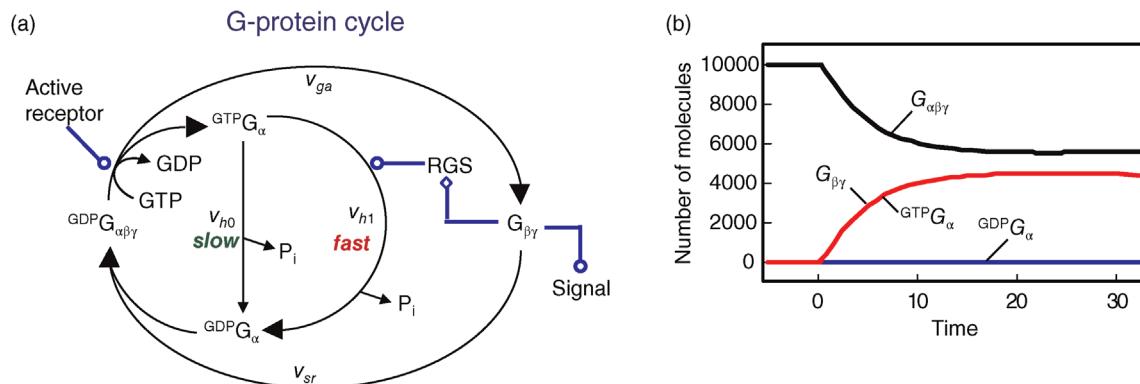


Figure 12.13 Activation cycle of G protein. (a) Without activation, the heterotrimeric G protein is bound to GDP. Upon activation by the activated receptor, an exchange of GDP with GTP occurs and the G protein is divided into GTP-bound G α and the heterodimer G $\beta\gamma$. G α -bound GTP is hydrolyzed, either slowly in reaction v_{h0} or fast in reaction v_{h1} supported by the RGS protein. GDP-bound G α can reassociate with G $\beta\gamma$ (reaction v_{sr}). (b) Time course of G protein activation. The total number of molecules is 10,000. The concentration of GDP-bound G α is low for the whole period due to fast complex formation with the heterodimer G $\beta\gamma$.

systems, and MAP kinase cascades. In this chapter, we will discuss their general composition and function as well as modeling approaches.

12.2.3.1 G Proteins

G proteins are essential parts of many signaling pathways. The reason for their name is that they bind the guanine nucleotides GDP and GTP. They are heterotrimers, that is, consist of three different subunits. G proteins are associated to cell surface receptors with a heptahelical transmembrane structure, the G protein-coupled receptors (GPCR). Signal transduction cascades involving (i) such a transmembrane surface receptor, (ii) an associated G protein, and (iii) an intracellular effector that produces a second messenger play an important role in cellular communication and are well studied [9,10]. In humans, G protein-coupled receptors mediate responses to light, flavors, odors, numerous hormones, neurotransmitters, and other signals [11–13]. In unicellular eukaryotes, receptors of this type mediate signals that affect such basic processes as cell division, cell–cell fusion (mating), morphogenesis, and chemotaxis [11,14–16].

The cycle of G protein activation and inactivation is shown in Figure 12.13. When GDP is bound, the G protein α subunit (G α) is associated with the G protein $\beta\gamma$ heterodimer (G $\beta\gamma$) and is inactive. Agonist binding to a receptor promotes guanine nucleotide exchange; G α releases GDP, binds GTP, and dissociates from G $\beta\gamma$. The dissociated subunits G α or G $\beta\gamma$, or both, are then free to activate target proteins (downstream effectors), which initiates signaling. When GTP is hydrolyzed, the subunits are able to reassociate. G $\beta\gamma$ antagonizes receptor action by inhibiting guanine nucleotide exchange. RGS (regulator of G protein signaling) proteins bind to G α , stimulate

GTP hydrolysis, and thereby reverse G protein activation. This general scheme can also be applied to the regulation

Example 12.2

The model of heterotrimeric G protein cycle of the yeast pheromone pathway was already mentioned in Example 12.1 and is linked to the receptor activation model via the concentration of the active receptor. The G protein cycle model comprises two ODEs and two algebraic equations for the mass conservation of the subunits G α and G $\beta\gamma$:

$$\begin{aligned} \frac{d}{dt} G_{\alpha\beta\gamma} &= -v_{ga} + v_{sr}, \\ \frac{d}{dt} G_{\alpha}GTP &= v_{ga} - v_{h0} - v_{h1}, \\ G_t &= G_{\alpha\beta\gamma} + G_{\alpha}GTP + G_{\alpha}GDP, \\ G_{\text{total}} &= G_{\alpha\beta\gamma} + G_{\beta\gamma}. \end{aligned} \quad (12.9)$$

The rate equations for the G protein activation, v_{ga} , the hydrolysis of G α GTP, v_{h0} and v_{h1} , and the subunit reassociation, v_{sr} , follow simple mass action kinetics:

$$\begin{aligned} v_{ga} &= k_{ga} \cdot R_a \cdot G_{\alpha\beta\gamma}, \\ v_{hi} &= k_{hi} \cdot G_{\alpha}GTP, \quad i = 0, 1, \\ v_{sr} &= k_{sr} \cdot G_{\beta\gamma} \cdot G_{\alpha}GDP. \end{aligned} \quad (12.10)$$

The parameters are $k_{ga} = 1 \times 10^{-5}$ (molecules per cell) $^{-1}$ s $^{-1}$, $k_{h0} = 0.004$ s $^{-1}$, $k_{h1} = 0.11$ s $^{-1}$, and $k_{sr} = 1$ (molecule per cell) $^{-1}$ s $^{-1}$. Note that in the original work, two different yeast strains have been considered. For the strains with a constantly active RGS (*SST2⁺*) or with a deletion of RGS (*sst2Δ*), the rate constants k_{h1} and k_{h0} have been set to zero, respectively. The time courses are shown in Figure 12.13b.

of small monomeric Ras-like GTPases, such as Rho. In this case, the receptor, $G\beta\gamma$, and RGS are replaced by GEF and GAP (see below).

Direct targets include different types of effectors, such as adenylyl cyclase, phospholipase C, exchange factors for small GTPases, some calcium and potassium channels, plasma membrane Na^+/H^+ exchangers, and certain protein kinases [10,17–19]. Typically, these effectors produce second messengers or other biochemical changes that lead to stimulation of a protein kinase or a protein kinase cascade (or, as mentioned, are themselves a protein kinase). Signaling persists until GTP is hydrolyzed to GDP and the $G\alpha$ and $G\beta\gamma$ subunits reassociate, completing the cycle of activation. The strength of the G protein-initiated signal depends on (i) the rate of nucleotide exchange, (ii) the rate of spontaneous GTP hydrolysis, (iii) the rate of RGS-supported GTP hydrolysis, and (iv) the rate of subunit reassociation. RGS proteins act as GTPase-activating proteins (GAPs) for a variety of different $G\alpha$ classes. Thereby, they shorten the lifetime of the activated state of a G protein, and contribute to signal cessation. Furthermore, they may contain additional modular domains with signaling functions and contribute to diversity and complexity of the cellular signaling networks [20–23].

12.2.3.2 Small G Proteins

Small G proteins are monomeric G proteins with molecular weight of 20–40 kDa. Like heterotrimeric G proteins, their activity depends on the binding of GTP. More than 100 small G proteins have been identified. They belong to five families: Ras, Rho, Rab, Ran, and Arf. They regulate a wide variety of cell functions as biological timers that initiate and terminate specific cell functions and determine the periods of time [24].

The transition from GDP-bound to GTP-bound states is catalyzed by a guanine nucleotide exchange factor (GEF), which induces exchange between the bound GDP and the cellular GTP. The reverse process is facilitated by a GTPase-activating protein (GAP), which induces hydrolysis of the bound GTP. Its dynamics can be described with the following equation with appropriate choice of the rates v_{GEF} and v_{GAP} :

$$\frac{d}{dt} {}^{GTP}\text{Ras} = - \frac{d}{dt} {}^{GDP}\text{Ras} = v_{\text{GEF}} - v_{\text{GAP}}. \quad (12.11)$$

Figure 12.14 illustrates the wiring of a *Ras* protein and the dependence of its activity on concentration ratio of the activating GEF and the deactivating GAP.

Mutations of the *Ras* protooncogenes (*H-Ras*, *N-Ras*, and *K-Ras*) are found in many human tumors. Most of these mutations result in the abolishment of normal GTPase activity of Ras. The Ras mutants can still bind to GAP, but cannot catalyze GTP hydrolysis. Therefore, they stay active for a long time.

12.2.3.3 Phosphorelay Systems

Most phosphorylation events in signaling pathways take place under consumption of ATP. Phosphorelay (or phosphotransfer) systems employ another mechanism: after an initial phosphorylation using ATP (or another phosphate donor), the phosphate group is transferred directly from one protein to the next without further consumption of ATP (or external donation of phosphate). Examples are the bacterial phosphoenolpyruvate:carbohydrate phosphotransferase [25–28], the two-component system of *E. coli*, or the Sln1 pathway involved in osmoresponse of yeast [29].

Figure 12.15 shows a scheme of a phosphorelay system from the high osmolarity glycerol (HOG) signaling

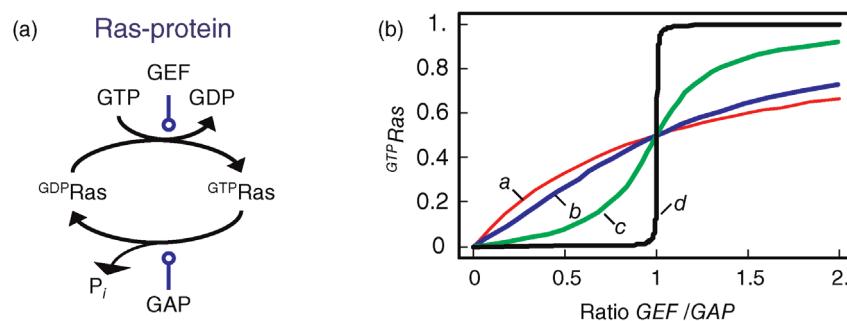


Figure 12.14 The Ras activation cycle. (a) Wiring diagram: GEF supports the transition from GDP-bound to GTP-bound states to activate Ras, while GAP induces hydrolysis of the bound GTP resulting in Ras deactivation. (b) Steady states of active Ras depending on the concentration ratio of its activator GEF and the inhibitor GAP. We compare the behavior for a model with mass action kinetics (curve *a*) with the behavior obtained with Michaelis–Menten kinetics for decreasing values of the K_M -value (curves *b*–*d*). The smaller the K_M -value, the more sigmoidal the response curve leading to an almost step-like shape in case of very low K_M -values. Parameters: $\text{Ras}_{\text{total}} = {}^{GTP}\text{Ras} + {}^{GDP}\text{Ras} = 1$, $k_1 = k_2 = 1$ (all panels), (b) $K_{M1} = K_{M2} = 1$, (c) $K_{M1} = K_{M2} = 0.1$, (d) $K_{M1} = K_{M2} = 0.001$.

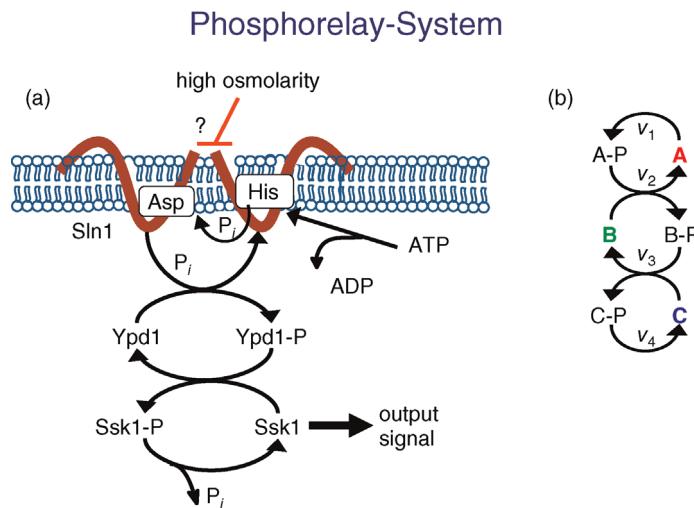


Figure 12.15 Schematic representation of a phosphorelay system. (a) Phosphorelay system belonging to the Sln1-branch of the HOG pathway in yeast. (b) General scheme of phosphorylation and dephosphorylation in a phosphorelay.

pathway in yeast as well as a schematic representation of a phosphorelay system.

This pathway is organized as follows [30]. It involves the transmembrane protein Sln1, which is present as a dimer. Under normal conditions, the pathway is active, since Sln1 continuously autophosphorylates at a histidine residue, Sln1H-P, under consumption of ATP. Subsequently, this phosphate group is transferred to an aspartate group of Sln1 (resulting in Sln1A-P), then to a histidine residue of Ypd1 and finally to an aspartate residue of Ssk1. Ssk1 is continuously dephosphorylated by a phosphatase. Without stress, the proteins are mainly present in their phosphorylated form. The pathway is blocked by an increase in the external osmolarity and a concomitant loss of turgor pressure in the cell. The phosphorylation of Sln1 stops, the pathway runs out of transferable phosphate groups, and the concentration of Ssk1 rises. This constitutes the downstream signal. The temporal behavior of a generalized phosphorelay system (Figure 12.15) can be described with the following set of ODEs:

$$\begin{aligned} \frac{d}{dt}A &= -k_1 \cdot A + k_2 \cdot AP \cdot B, \\ \frac{d}{dt}B &= -k_2 \cdot AP \cdot B + k_3 \cdot BP \cdot C, \\ \frac{d}{dt}C &= -k_3 \cdot BP \cdot C + k_4 \cdot CP. \end{aligned} \quad (12.12)$$

For the ODE system in Eq. (12.12), the following conservation relations hold

$$\begin{aligned} A_{\text{total}} &= A + AP, \\ B_{\text{total}} &= B + BP, \\ C_{\text{total}} &= C + CP. \end{aligned} \quad (12.13)$$

The existence of conservation relations is in agreement with the assumption that production and degradation of the proteins occurs on a larger time scale as the phosphorylation events.

The temporal behavior of a phosphorelay system upon external stimulation is shown in Figure 12.16. Before the stimulus, the concentrations of A, B, and C assume low, but nonzero, levels due to continuous flow of phosphate groups through the network. During stimulation, they increase one after the other up to a maximal level that is determined by the total concentration of each protein. After removal of stimulus, all three concentrations return quickly to their initial values.

Figure 12.16b illustrates the dependence of the sensitivity of the phosphorelay system on the value of the terminal dephosphorylation. For a low rate constant k_4 , for example, $k_4 < 0.001$, the concentration C is low (almost) independent of the value of k_1 , while for high k_4 , for example, $k_4 > 10$, the concentration C is (almost always) maximal. Changing of k_1 leads to a change of C-levels only in the range $0.001 < k_4 < 10$.

This system is an example for a case where we can draw initial conclusions about feasible parameter values just from the network structure and the task of the module.

12.2.3.4 MAP Kinase Cascades

Mitogen-activated protein kinases (MAPKs) are a family of serine/threonine kinases that transduce signals from the cell membrane to the nucleus in response to a wide range of stimuli. Independent or coupled kinase cascades participate in many different intracellular signaling pathways that control a spectrum of cellular processes, including cell growth, differentiation, transformation, and

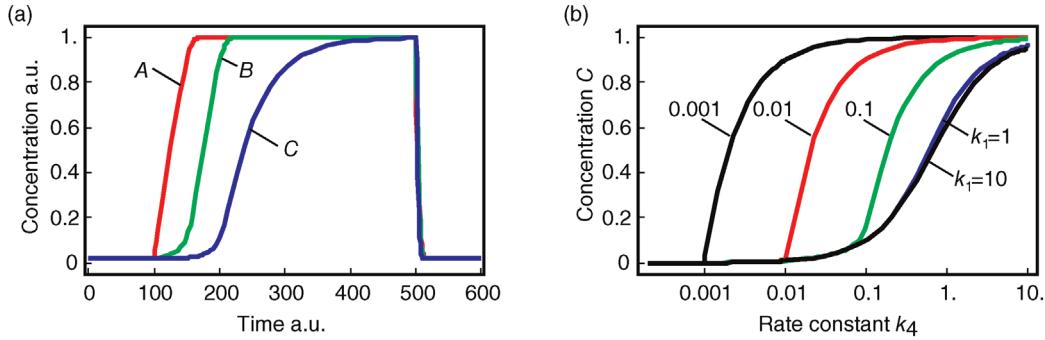


Figure 12.16 Dynamics of the phosphorelay system. (a) Time courses after stimulation from time 100 to time 500 (a.u.) by decreasing k_1 to zero. (b) Dependence of steady-state level of the phosphorelay output, C , on the cascade activation strength, k_1 , and the terminal dephosphorylation, k_4 . Parameter values: $k_1 = k_2 = k_3 = 1$, $k_4 = 0.02$, $A_{\text{total}} = B_{\text{total}} = C_{\text{total}} = 1$.

apoptosis. MAPK cascades are widely involved in eukaryotic signal transduction, and these pathways are conserved from yeast to mammals.

A general scheme of a MAPK cascade is depicted in Figure 12.17. This pathway consists of several levels (usually three to four), where the activated kinase at each level phosphorylates the kinase at the next level down the cascade. The MAP kinase (MAPK) is at the terminal level of the cascade. It is activated by the MAPK kinase (MAPKK) by phosphorylation of two sites, conserved threonine and tyrosine residues. The MAPKK is itself phosphorylated at serine and threonine residues by the MAPKK kinase (MAPKKK). Several mechanisms are known to activate MAPKKKs by phosphorylation of a tyrosine residue. In

some cases, the upstream kinase may be considered as a MAPKKK kinase (MAPKKKK). Dephosphorylation of either residue is thought to inactivate the kinases, and mutants lacking either residue are almost inactive. At each cascade level, protein phosphatases can inactivate the corresponding kinase, although it is in some cases a matter of debate whether this reaction is performed by an independent protein or by the kinase itself as autodephosphorylation. Also ubiquitin-dependent degradation of phosphorylated proteins is reported.

Although they are conserved through species, elements of the MAPK cascade got different names in various studied systems. Some examples are listed in Table 12.1 (see also [31]).

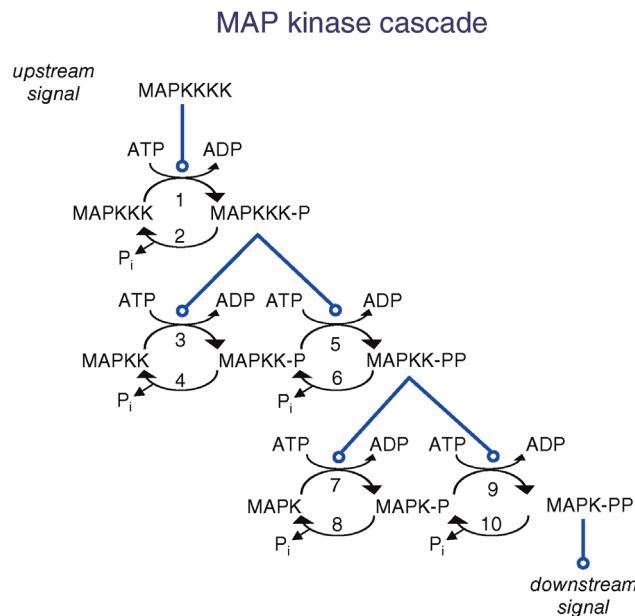


Figure 12.17 Schematic representation of the MAP kinase cascade. An upstream signal (often by a further kinase called MAP kinase kinase kinase kinase) causes phosphorylation of the MAPKKK. The phosphorylated MAPKKK in turn phosphorylates the protein at the next level. Dephosphorylation is assumed to occur continuously by phosphatases or autodephosphorylation.

Table 12.1 Names of the components of MAP kinase pathways in different organisms and different pathways.

Organism	Budding yeast		Xenopus oocytes	Human, cell-cycle regulation		
	HOG pathway	Pheromone pathway		p38 pathway	JNK pathway	
MAPKKK	Ssk2/Ssk22	Ste11	Mos	Rafs (c-, A-, and B-)	Tak1	MEKKs
MAPKK	Pbs2	Ste7	MEK1	MEK1/2	MKK3/6	MKK4/7
MAPK	Hog1	Fus3	p42 MAPK	ERK1/2	p38	JNK1/2

In the following, we will present typical modeling approaches and then discuss functional properties of signaling cascades. The dynamics of a MAPK cascade may be represented by the following ODE system:

$$\frac{d}{dt} MAPKKK = -v_1 + v_2, \quad (12.14)$$

$$\frac{d}{dt} MAPKKK - P = v_1 - v_2,$$

$$\frac{d}{dt} MAPKK = -v_3 + v_4, \quad (12.15)$$

$$\frac{d}{dt} MAPKK - P = v_3 - v_4 - v_5 + v_6,$$

$$\frac{d}{dt} MAPKK - P_2 = v_5 - v_6, \quad (12.16)$$

$$\frac{d}{dt} MAPK = -v_7 + v_8, \quad (12.17)$$

$$\frac{d}{dt} MAPK - P = v_7 - v_8 - v_9 + v_{10}.$$

$$\frac{d}{dt} MAPK - P_2 = v_9 - v_{10},$$

The variables in the ODE system fulfill a set of moiety conservation relations, irrespective of the concrete choice of expression for the rates v_1, \dots, v_{10} . It holds

$$MAPKK_{\text{total}} = MAPKK + MAPKK - P, \quad (12.17)$$

$$MAPKK_{\text{total}} = MAPKK + MAPKK - P + MAPKK - P_2,$$

$$MAPK_{\text{total}} = MAPK + MAPK - P + MAPK - P_2.$$

The conservation relations reflect that we do not consider production or degradation of the involved proteins in this model. This is justified by the supposition that protein production and degradation take place on a different time scale as signal transduction.

The choice of the expressions for the rates is a matter of elaborateness of experimental knowledge and of modeling taste. We will discuss here different possibilities. Assuming only mass action results in linear and bilinear expressions such as

$$v_1 = k_1 \cdot MAPKK \cdot MAPKKK, \quad (12.18)$$

$$v_2 = k_2 \cdot MAPKK - P.$$

The kinetic constants k_i are first- (i even) and second-order (i odd) rate constants. In these expressions, the concentrations of the donor and acceptor of the transferred phosphate group, ATP and ADP, are not explicitly considered but included in the rate constants k_1 and k_2 . Considering ATP and ADP explicitly results in

$$v_1 = k_1 \cdot MAPKKK \cdot MAPKKK \cdot ATP, \quad (12.19)$$

$$v_2 = k_2 \cdot MAPKK - P.$$

In addition, we have to care about the ATP–ADP balance and add three more differential equations

$$\frac{d}{dt} ATP = -\frac{d}{dt} ADP = -\sum_{i \text{ odd}} v_i, \quad (12.20)$$

$$\frac{d}{dt} P_i = \sum_{i \text{ even}} v_i.$$

Here, we find two more conservation relations, the conservation of adenine nucleotides, $ATP + ADP = \text{const.}$, and the conservation of phosphate groups

$$MAPKK - P + MAPKK - P + 2 \cdot MAPKK - P_2 + MAPK - P + 2 \cdot MAPK - P_2 + 3 \cdot ATP + 2 \cdot ADP + P = \text{const.} \quad (12.21)$$

One may take into account that enzymes catalyze all steps [32] and therefore consider Michaelis–Menten kinetics for the individual steps [33]. Taking again the first and second reactions as examples for kinase and phosphatase steps, we get

$$v_1 = k_1 \cdot MAPKKK \frac{MAPKK}{K_{m1} + MAPKK}, \quad (12.22)$$

$$v_2 = \frac{V_{\max 2} \cdot MAPKK - P}{K_{m2} + MAPKK - P},$$

where k_1 is a first-order rate constant, K_{m1} and K_{m2} are Michaelis constants, and $V_{\max 2}$ denotes a maximal enzyme rate. Reported values for Michaelis constants are 15 nM [33], 46 nM and 159 nM [34], and 300 nM [32]. For maximal rates, values of about $0.75 \text{ nM} \cdot \text{s}^{-1}$ [33] are used in models.

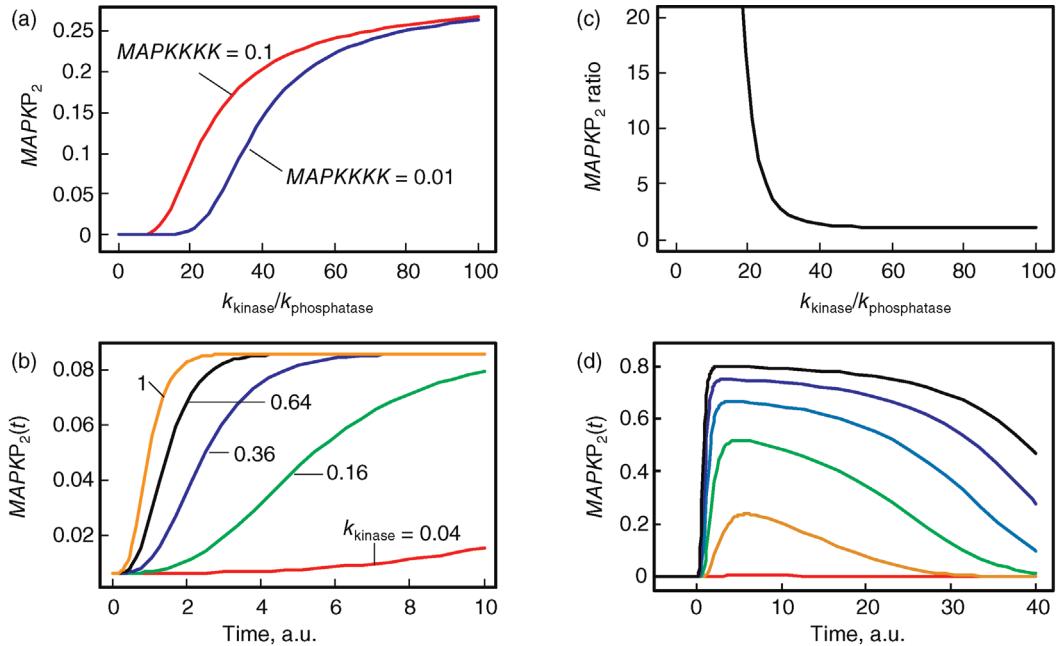


Figure 12.18 Parameter dependence of MAPK cascade performance. Shown are steady-state simulations for changing values of rate constants for kinases, k_+ , and phosphatases, k_- (in arbitrary units). Upper row: Absolute values of the output signal MAPK-PP depending on the input signal (high: MAPKKK = 0.1 or low: MAPKKK = 0.01) for varying ratio of k_+/k_- . Second row: ratio of the output signal for high versus low input signal (MAPKKK = 0.1 or MAPKKK = 0.01) for varying ratio of k_+/k_- ; right panel: time course of MAPK activation for different values of k_+ and a ratio $k_+/k_- = 20$.

The performance of MAPK cascades, that is, their ability to amplify the signal, to enhance the concentration of the double phosphorylated MAPK notably, and the speed of activation, depends crucially on the kinetic constants of the kinases, k_+ , and phosphatases, k_- (Eq. (12.19)), and, moreover, on their ratio (see Figure 12.18). If the ratio k_+/k_- is low (phosphatases stronger than kinases), then the amplification is high, but at very low absolute concentrations of phosphorylated MAPK. High values of k_+/k_- ensure high absolute concentrations of MAPK-P2, but with negligible amplification. High values of both k_+ and k_- ensure fast activation of downstream targets.

Frequently, the proteins of MAPK cascades interact with scaffold proteins. In this case, a reversible assembly of oligomeric protein complexes that includes both enzymatic proteins and proteins without known enzymatic activity precedes the signal transduction. These nonenzymatic components can serve as scaffolds or anchors to the plasma membrane and regulate the efficiency, specificity, and localization of the signaling pathway.

12.2.3.5 The Wnt/β-Catenin Signaling Pathway

The Wnt/β-catenin signaling pathway is an important regulatory pathway for a multitude of processes in higher eukaryotic systems. It has been found due to its role in tumor growth, especially through the use of mouse models of cancer and oncogenic retroviruses. However, it is

also important in normal embryonic development. It is relevant for cell fate specification, cell proliferation, and cell migration or formation of body axis. Recently, its role in stem cell differentiation and reprogramming has been discussed.

In fact, the Wnt signaling comprises not just one pathway, but at least three pathways, that is, the canonical Wnt pathway, the noncanonical planar cell polarity pathway, and the noncanonical Wnt/calcium pathway. All three pathways employ binding of the Wnt ligand to the Frizzled receptors, but have different downstream architecture and effects (Figure 12.19). Wnt denotes a family of 19 genes (in vertebrates, other figures hold for other branches of the animal kingdom).

The major function of the canonical pathway is to stimulate the accumulation of β-catenin, which is normally kept at a low concentration through balanced continuous production and degradation by the destruction complex comprising adenomatous polyposis coli (APC), axin (axin1 or axin2), glycogen synthase kinase 3 (GSK3), and casein kinase 1a (CK1). This complex targets β-catenin for ubiquitination and proteasomal digestions. The G-protein-coupled receptor Frizzled often functions together with coreceptors, such as the lipoprotein receptor-related protein LRP or receptor tyrosine kinases. When the ligand Wnt binds to the receptor and its co-receptors, they recruit cytoplasmic proteins to the

Wnt signaling pathway

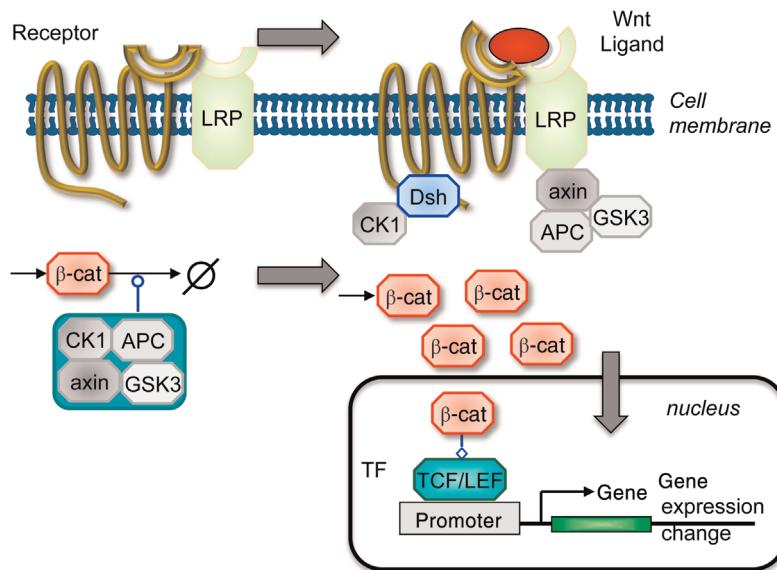


Figure 12.19 The Wnt signaling pathway. The receptor Frizzled is a G-protein-coupled receptor and frequently occurs together with a coreceptor such as LRP. Without Wnt signaling, β -catenin is continuously produced and degraded by the destruction complex comprising APC, axin, GSK3, and CK1, resulting in low β -catenin concentrations. Upon binding of the Wnt ligand, the phosphoprotein Dsh is recruited to the receptor. The destruction complex is reorganized, partially binding to the membrane proteins. β -catenin can accumulate, enter the nucleus and activate transcription factors leading to gene expression changes.

membrane (level 1 response, see Section 12.2.1): the phosphoprotein Dsh is recruited to the receptor, axin binds to LPR. The destruction complex is reorganized and the GSK3 activity is inhibited. β -catenin can accumulate (level 2 response). Among other consequences, it enters the nucleus and activates transcription factors leading to gene expression changes (level 4 response).

An early experiment-based model of the dynamics of the destruction complex and its regulation upon Wnt signaling investigated the quantitative roles of APC and axin (present in low amounts) [35]. They investigated which reaction steps would have high control (see concentration control coefficients introduced in Chapter 4) over the β -catenin concentration and found that strong negative control is exerted by the assembly of the destruction complex and the binding of β -catenin to the destruction complex. Positive control is exerted by reactions leading to the disassembly of the destruction complex, the dissociation of β -catenin from the destruction complex, the axin degradation, and the β -catenin synthesis.

Another model has analyzed the robustness of β -catenin levels in response to Wnt signaling [36]. They found that while the absolute levels of β -catenin are very sensitive to many types of perturbations, the fold changes in

response to Wnt are not. Here fold change means the level of β -catenin after Wnt-stimulation divided by the level before Wnt-stimulation. An incoherent feed-forward loop can cause such a behavior [37].

The following minimal model is suited to demonstrate major features of the canonical Wnt signaling pathway in wild-type and some mutant conditions [38]. The wiring of the model is shown in Figure 12.20. The ODE system reads

$$\begin{aligned} \frac{d\beta\text{-catenin}}{dt} &= v_1 - v_2 - v_3 - v_4 - v_5, \\ \frac{dAPC}{dt} &= -v_4, \\ \frac{dTcf}{dt} &= -v_5 + v_6 - v_7, \\ \frac{d\beta\text{-catenin}/Tcf}{dt} &= v_5, \\ \frac{dDsh_a}{dt} &= v_8 - v_9, \\ \frac{dmRNA}{dt} &= v_{10} - v_{11}, \\ \frac{drepresor}{dt} &= v_{12} - v_{13}. \end{aligned}$$

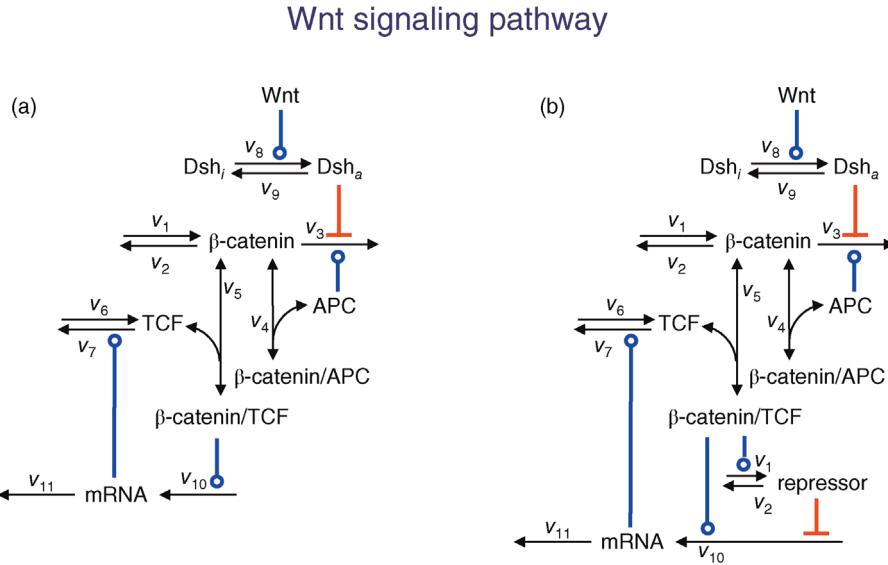


Figure 12.20 Wiring of the minimal model for the Wnt signaling pathway. (a) Basic model, (b) model including the incoherent feedforward loop via consideration of the repressor.

Conservation relations:

$$\begin{aligned} Dsh^{tot} &= Dsh_i + Dsh_a, \\ APC^{tot} &= APC + \beta\text{-catenin}/APC. \end{aligned}$$

Rate equations:

$$\begin{aligned} v_1 &= \text{constant}, \\ v_2 &= k_2 \cdot \beta\text{-catenin}, \\ v_3 &= \frac{k_3 \cdot APC \cdot \beta\text{-catenin}}{K + Dsh_a}, \\ v_4 &= k_4 \cdot APC \cdot \beta\text{-catenin} - k_{-4} \cdot \beta\text{-catenin}/APC, \\ v_5 &= k_5 \cdot TCF \cdot \beta\text{-catenin} - k_{-5} \cdot \beta\text{-catenin}/TCF, \\ v_6 &= \text{constant}, \\ v_7 &= k_7 \cdot TCF, \\ v_8 &= k_8 \cdot Dsh_i \cdot Wnt, \\ v_9 &= k_9 \cdot Dsh_a, \\ v_{10} &= v_{10}^{\max} \frac{K_2 \cdot K_3 \cdot \beta\text{-catenin}/TCF}{K_1 \cdot K_2 \cdot K_3 + K_2 \cdot K_3 \cdot \beta\text{-catenin}/TCF + K_1 \cdot K_3 \cdot \text{repressor} + K_1 \cdot K_2 \cdot \beta\text{-catenin}/TCF \cdot \text{repressor}}, \\ v_{11} &= k_{11} \cdot mRNA, \\ v_{12} &= k_{12} \cdot \beta\text{-catenin}/TCF, \\ v_{13} &= k_{13} \cdot \text{repressor}. \end{aligned} \tag{12.23}$$

The parameter values are $v_1 = 0.423 \text{ nM min}^{-1}$, $k_2 = 2.57 \cdot 10^{-4} \text{ min}^{-1}$, $k_3 = 3.08 \cdot 10^{-3} \text{ min}^{-1}$, $K = 18 \text{ nM}$, $k_4 = 10^5 \text{ nM}^{-1} \text{ min}^{-1}$, $k_{-4} = 1.2 \cdot 10^{-8} \text{ min}^{-1}$, $k_5 = 0.0333 \text{ min}^{-1}$, $k_{-5} = 1 \text{ min}^{-1}$, $v_6 = 0.686 \text{ nM min}^{-1}$, $k_7 = 0.084 \text{ min}^{-1}$, $k_8 = 3 \cdot 10^{-3} \text{ nM}^{-1} \text{ min}^{-1}$, $k_9 = 6.7 \cdot 10^{-4} \text{ min}^{-1}$, $v_{10}^{\max} = 1368.18 \text{ nM min}^{-1}$, $k_{11} = 10^{-2} \text{ min}^{-1}$, $k_{12} = 4.44 \cdot 10^{-3} \text{ min}^{-1}$, $k_{13} = 10^{-3} \text{ min}^{-1}$, $K_1 = 1300 \text{ nM}$, $K_2 = 0.03 \text{ nM}$, and $K_3 = 4 \cdot 10^4 \text{ nM}$.

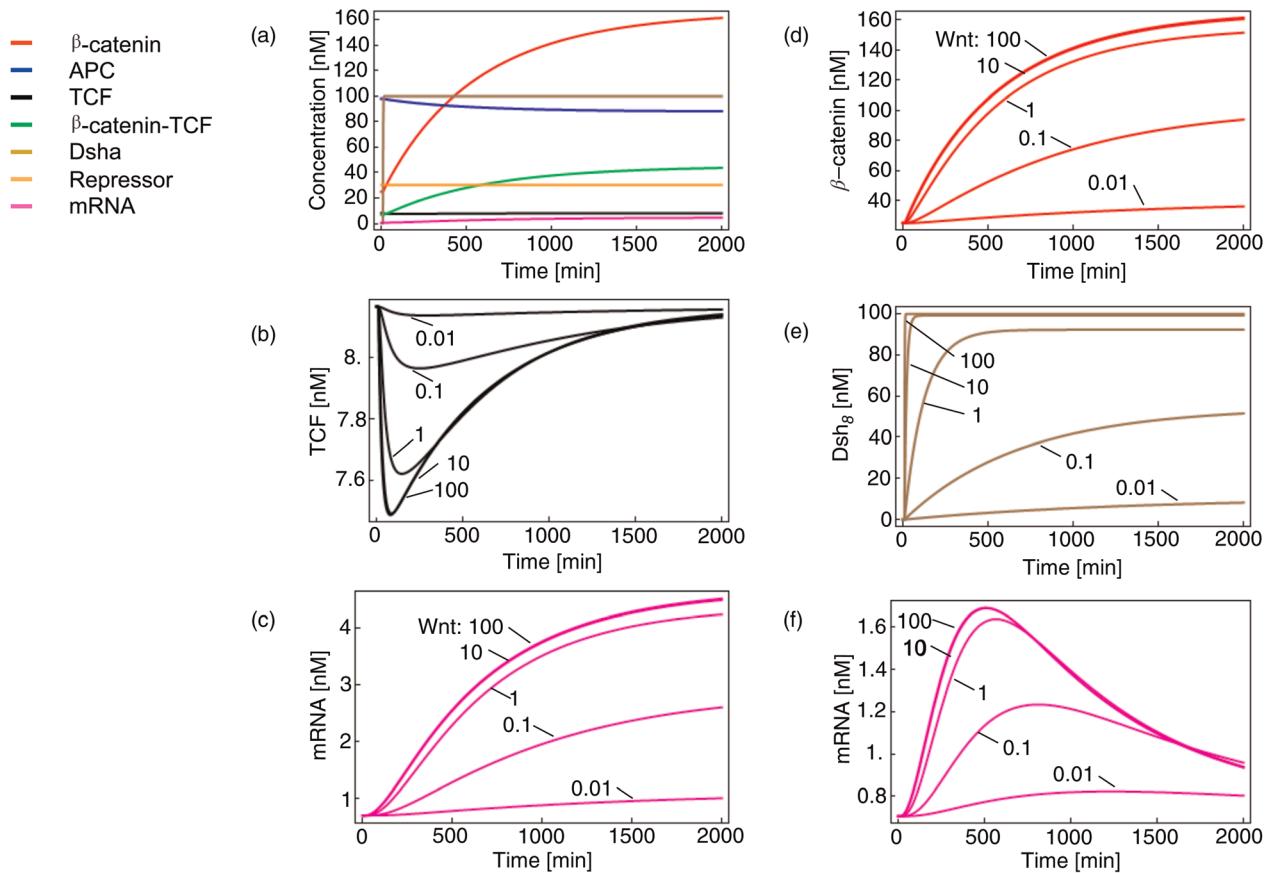


Figure 12.21 Time course simulations for the minimal model of the Wnt pathway as shown in Figure 12.20 and given by equation set (12.23). Panel (a) shows the response of all components to a Wnt stimulus of 100 nM at time point 10 min. All other panels: Wnt stimulus varies from 0.01 to 100 nM. We see saturation for Wnt above 10 nM. Panels (a)–(c) show simulations for the case without repressor, while panels (d)–(f) include a repressor. The effect of repressor is most obvious for mRNA dynamics shown in (c) and (f) where the presence of repressor leads to overshoot behavior.

The dynamics and steady-state behavior of the model are shown in Figure 12.21.

12.2.4 Analysis of Dynamic and Regulatory Features of Signaling Pathways

12.2.4.1 Detecting Feedback Loops in Dynamical Systems

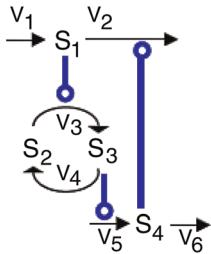
The existence of positive or negative feedback cycles in complex signaling networks that are described with ODE systems can be derived from the analysis of the Jacobian matrix \mathbf{M} (see Chapter 4, Chapter 15, Eq. (15.39)) [39,40]. With $f_i = dx_i/dt$ ($i = 1, \dots, n$) being differential equations describing the dynamics of component x_i and $a_{ij} = \partial f_i / \partial x_j$ being the terms of the Jacobian $\mathbf{M} = \{a_{ij}\}$, we can classify single nonzero terms of the Jacobian or sequences thereof as follows:

- $a_{ii} > 0$ for some i denotes direct autocatalysis,
- $a_{ii} < 0$ for some i denotes direct autoinhibition,
- $a_{ij} \cdot a_{ji} > 0$ for some i and j denotes a positive feedback including two components. It has been termed symbiosis for both $a_{ij}, a_{ji} > 0$ or competition for both $a_{ij}, a_{ji} < 0$,
- $a_{ij} \cdot a_{ji} < 0$ for some i and j denotes a negative feedback including two components,
- $a_{ij} \cdot a_{jk} \cdots a_{pi} > 0$ is a positive feedback of length $p - i$, and
- $a_{ij} \cdot a_{jk} \cdots a_{pi} < 0$ is a negative feedback of length $p - i$.

Obviously, a feedback loop should not contain more than n elements a_{ij} in a sequence (otherwise, it would be repetitive). Zero elements of the Jacobian in a sequence cut a feedback. An odd number of negative elements causes negative feedback, while a positive (or zero) number of negative terms leads to positive feedback.

Example 12.3

Consider the network given below:



With the following dynamics entailing only mass action kinetics

$$\begin{aligned}\frac{dS_1}{dt} &= v_1 - v_2 = v_1 - k_2 S_1 S_4, \\ \frac{dS_2}{dt} &= v_4 - v_3 = k_4 S_3 - k_3 S_2 S_1, \\ \frac{dS_3}{dt} &= -v_4 + v_3 = -k_4 S_3 + k_3 S_2 S_1, \\ \frac{dS_4}{dt} &= v_5 - v_6 = k_5 S_3 - k_6 S_4,\end{aligned}\quad (12.24)$$

the respective Jacobian matrix reads

$$M = \begin{pmatrix} -k_2 S_4 & 0 & 0 & -k_2 S_1 \\ -k_3 S_2 & -k_3 S_1 & k_4 & 0 \\ k_3 S_2 & k_3 S_1 & -k_4 & 0 \\ 0 & 0 & k_5 & -k_6 \end{pmatrix}. \quad (12.25)$$

Since kinetic rate constants and compound concentrations are *per se* positive (or formally at least nonnegative), the systems contains the following feedback cycles:

- Direct autoinhibition of all components through either degradation ($a_{11} = -k_2 S_4 < 0, a_{44} < 0$) or conversion ($a_{22} < 0, a_{33} < 0$) being proportional to their actual concentration.
- One positive two-component feedback from S_2 to S_3 (and *vice versa*) because of $a_{23} \cdot a_{32} = k_4 \cdot k_3 S_1 > 0$. There are no further two-component feedback loops since all other potential combinations are zero ($a_{12} \cdot a_{21} = a_{13} \cdot a_{31} = a_{14} \cdot a_{41} = a_{24} \cdot a_{42} = a_{34} \cdot a_{43} = 0$).
- One negative feedback loop including S_1, S_4 , and S_3 ($a_{14} \cdot a_{43} \cdot a_{31} = -k_2 S_1 \cdot k_5 \cdot k_3 S_2 < 0$).
- One positive feedback loop including all four components ($a_{14} \cdot a_{43} \cdot a_{32} \cdot a_{21} = -k_2 S_1 \cdot k_5 \cdot k_3 S_1 \cdot (-k_3 S_2) > 0$).

Signaling pathways can exhibit interesting dynamic and regulatory features, such as other regulatory pathway. Dynamic motifs are discussed in detail in Section 8.2.

Among the various regulatory features of signaling pathways, negative feedback has attracted outstanding interest. It also plays an important role in metabolic pathways, for example, in amino acid synthesis pathways (Section 12.1.3), where a negative feedback signal from the amino acid at the end to the precursors at the beginning of the pathway prevents an overproduction of this amino acid. The implementation of the feedback and the respective dynamic behavior show a wide variation. Feedback can bring about limit cycle-type oscillations in cell-cycle models. In signaling pathways, negative feedback may cause an adjustment of the response or damped oscillations [41].

12.2.4.2 Quantitative Measures for Properties of Signaling Pathways

The dynamic behavior of signaling pathways can be quantitatively characterized by a number of measures [42]. Let $X_i(t)$ be the time-dependent concentration of the kinase i (or another interesting compound). The signaling time τ_i describes the average time to activate the kinase i . The signal duration ϑ_i gives the average time during which the kinase i remains activated. The signal amplitude S_i is a measure for the average concentration of activated kinase i . The following definitions have been introduced. The quantity

$$I_i = \int_0^\infty X_i(t) dt, \quad (12.26)$$

is the total amount of active kinase i generated during the signaling period, that is, the integrated response of X_i (the area covered by a plot $X_i(t)$ versus time). Further measures are

$$T_i = \int_0^\infty t \cdot X_i(t) dt, \quad (12.27)$$

$$Q_i = \int_0^\infty t^2 \cdot X_i(t) dt, \quad (12.28)$$

The signaling time can now be defined as

$$\tau_i = T_i / I_i, \quad (12.29)$$

that is, as the average of time, analogous to mean value of a statistical distribution. Note that other definitions for characteristic times have been introduced in Section 6.3. The signal duration

$$\vartheta_i = \sqrt{Q_i / I_i - \tau_i^2}, \quad (12.30)$$

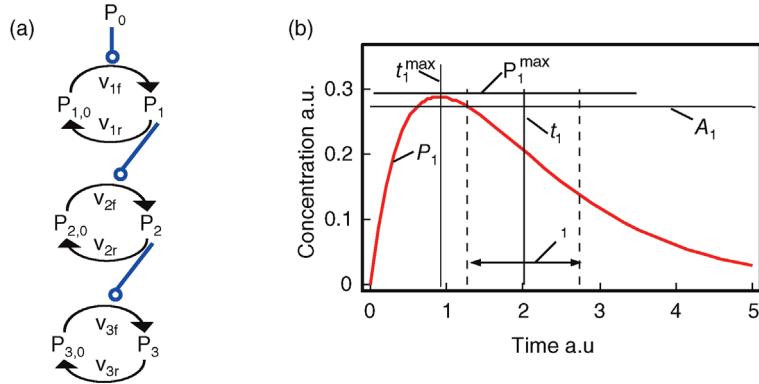


Figure 12.22 Characteristic measures for dynamic variables. (a) Wiring of an example signaling cascade with $v_{if} = k_{if} \cdot P_{i-1}(t) \cdot (1 - P_i(t))$, $v_{ir} = k_{ir} \cdot P_i(t)$, $k_{if} = k_{ir} = 1$, $dP_0(t)/dt = -P_0(t)$, and $P_0(0) = 1$, $P_i(0) = 0$ for $i = 1, \dots, 3$. (b) Time courses of X_i . The horizontal lines indicate the concentration measures for X_1 , that is, the calculated signal amplitude A_1 and P_1^{\max} , and vertical lines indicate time measures for P_1 , that is, the time t_1^{\max} of P_1^{\max} , the characteristic time τ_1 and the dotted vertical lines cover the signaling time θ_1 .

gives a measure of how extended the signaling response is around the mean time (compatible to standard deviation). The signal amplitude is defined as

$$A_i = I_i / (2\theta_i). \quad (12.31)$$

In a geometric representation, this is the height of a rectangle whose length is $2\theta_i$ and whose area equals the area under the curve $X_i(t)$. Note that this measure might be different from the maximal value X_i^{\max} that $X_i(t)$ assumes during the time course.

Figure 12.22 shows a signaling pathway with successive activation of compounds and the respective time courses. The characteristic quantities are given in Table 12.2 and for $X_1(t)$ shown in the figure.

12.2.4.3 Crosstalk in Signaling Pathways

Signal transmission in cellular context is often not unique in the sense that an activated protein has a high specificity for another protein. Instead, there might be crosstalk, that is, proteins of one signaling pathway interact with proteins assigned to another pathway. Strictly speaking, the assignment of proteins to one pathway is often arbitrary and may result, for example, from the history of their function discovery. Frequently, protein interactions form a network with various binding, activation, and inhibition events, such as illustrated in Figure 12.10.

In order to arrive at quantitative measures for crosstalk, let us consider the simplified scheme in Figure 12.23: External signal α binds to receptor A, which activates target A via a series of interactions. In the same way, external signal β binds to receptor B, which activates target B. In addition, there are events that mediate an effect of receptor B on target A.

Let us concentrate on pathway A and define all measures from its perspective. Signaling from α via R_A to T_A shall be called intrinsic, while signals from β to T_A are extrinsic. Further, in order to quantify crosstalk, we need a quantitative measure for the effect of an external stimulus on the target. If we are interested in the level of activation, such a measure might be the integral over the time course of T_A (Eq. (12.27)), its maximal value or its amplitude (Eq. (12.31)). If we are interested in the response timing, we can consider the time of the maximal value or the characteristic time (for an overview on measures see Table 12.2). Whatever measure we chose, shall be denoted by X in the following.

The crosstalk C is the activation of pathway A by the extrinsic stimulus β relative to the intrinsic stimulus α

$$C = \frac{X(\text{e})}{X(\text{i})} = \frac{X_A(\beta)}{X_A(\alpha)}. \quad (12.32)$$

Table 12.2 Dynamic characteristics of the signaling cascade shown in Figure 12.22.

Compound	Integral, I_i	Maximum, X_i^{\max}	Time (X_i^{\max}), t_i^{\max}	Characteristic time, τ_i	Signal duration, θ_i	Signal amplitude, A_i
X_1	0.797	0.288	0.904	2.008	1.458	0.273
X_2	0.695	0.180	1.871	3.015	1.811	0.192
X_3	0.629	0.133	2.855	4.020	2.109	0.149

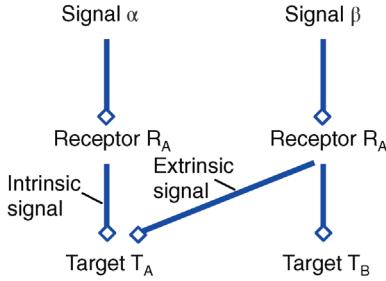


Figure 12.23 Crosstalk of signaling pathways.

The fidelity F [43] is viewed as output due to the intrinsic signal divided by the output in response to the extrinsic signal and reads in our notation:

$$F = \frac{X_{T_A}(\alpha)/X_{R_A}(\alpha)}{X_{T_A}(\alpha)/X_{R_B}(\beta)}. \quad (12.33)$$

In addition, the intrinsic sensitivity S_i expresses how an extrinsic signal modifies the intrinsic signal when acting in parallel, while the extrinsic sensitivity S_e quantifies the effect of the intrinsic signal on the extrinsic signal, respectively [44]:

$$S_i(A) = \frac{X_A(\alpha)}{X_A(\alpha, \beta)} \quad \text{and} \quad S_e(A) = \frac{X_A(\beta)}{X_A(\alpha, \beta)}. \quad (12.34)$$

Table 12.3 shows how different specificity values can be interpreted.

Table 12.3 Effect of crosstalk on signaling.

$S_e > 1$	$S_e < 1$
$S_i > 1$ Mutual signal inhibition	Dominance of intrinsic signal
$S_i < 1$ Dominance of extrinsic signal	Mutual signal amplification

12.3 The Cell Cycle

Summary

The cell cycle is a fundamental cellular process that dominates many aspects of cellular biochemistry. It comprises a series of processes in the cell leading to the duplication of the genetic material and the cell mass and, eventually, to the formation of two daughter cells. In this section, we introduce the different phases of the eukaryotic cell cycle. The regulatory mechanisms that control the periodic process are discussed and mathematical models of different complexity are presented that describe the oscillatory process ensuring successful coordination of growth and replication (Figure 15.25).

Growth and reproduction are major characteristics of life. Crucial for these is the cell division by which one cell divides into two and all constituents of the mother cell are distributed between the two daughter cells. This requires that the genome has to be duplicated in advance, which is accomplished by the DNA polymerase, an enzyme that utilizes deoxynucleotide triphosphates (dNTPs) for the synthesis of two identical DNA double

Example 12.4

Consider the coupling of a fast and a slower signaling pathway as depicted in Figure 12.24 with rate equations similar to those used in Figure 12.22, with the exception $v_{3Af} = k_{3Af} \cdot (P_{2A}(t) + P_{2B}(t)) \cdot (1 - P_{3A}(t))$. Let A be the slower pathway (all $k_{iAf} = k_{iAr} = 1$) and B the faster pathway (all $k_{iBf} = k_{iBr} = 10$). The pathways are activated by setting $P_{0A}(0), P_{0B}(0)$, or both from zero to one. The time courses show that crosstalk from B to A affects the pathway output $P_{3A}(t)$. A and B lead to faster activation of $P_{3A}(t)$ than A alone. Activation by B alone leads to drastically reduced $P_{3A}(t)$. Table 12.4 reports the quantitative crosstalk measures. We note mutual signal amplification in terms of integrated response (I_3) and maximal response ($\text{Max}(P_{3A})$), but dominance of the intrinsic signal on the level of signal timing (here $t_{P_{3A}}^{\max}$).

Table 12.4 Crosstalk measures for the pathway in Example 4.

$X_A(\alpha)$	$X_A(\beta)$	$X_A(\alpha, \beta)$	$S_i(A) = \frac{X_A(\alpha)}{X_A(\alpha, \beta)}$	$S_e(A) = \frac{X_A(\beta)}{X_A(\alpha, \beta)}$	$C = \frac{X_A(\beta)}{X_A(\alpha)}$
$I_3 = \int_0^\infty P_{3A}(t)dt$	0.628748	0.067494	0.688995	0.912557	0.09796
$t_{P_{3A}}^{\max}$	2.85456	0.538455	2.73227	1.04476	0.197072
$\text{Max}(P_{3A})$	0.132878	0.0459428	0.136802	0.971314	0.335833

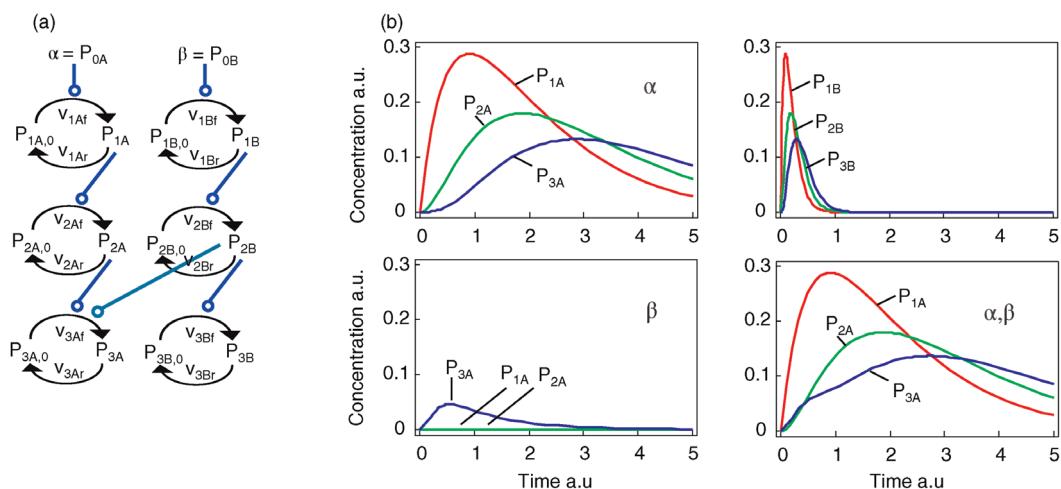


Figure 12.24 Crosstalk of MAP kinase pathways. (a) Pathway A leads to activation of P_{3A} upon stimulation by α , pathway B transmits signal from b to P_{3B} . Crosstalk occurs through signaling from P_{2B} . (b) Dynamics of pathways A and B upon stimulation by α , β , or both (as indicated).

strands from one parent double strand. In this case, each single strand acts as template for one of the new double strands. Several types of DNA polymerases have been found in prokaryotic and eukaryotic cells, but all of them synthesize DNA only in $5' \rightarrow 3'$ direction. In addition to DNA polymerase, several further proteins are involved in

DNA replication: proteins responsible for the unwinding and opening of the mother strand (template double strand), proteins that bind the opened single-stranded DNA and prevent it from rewinding during synthesis, an enzyme called primase that is responsible for the synthesis of short RNA primers that are required by the DNA

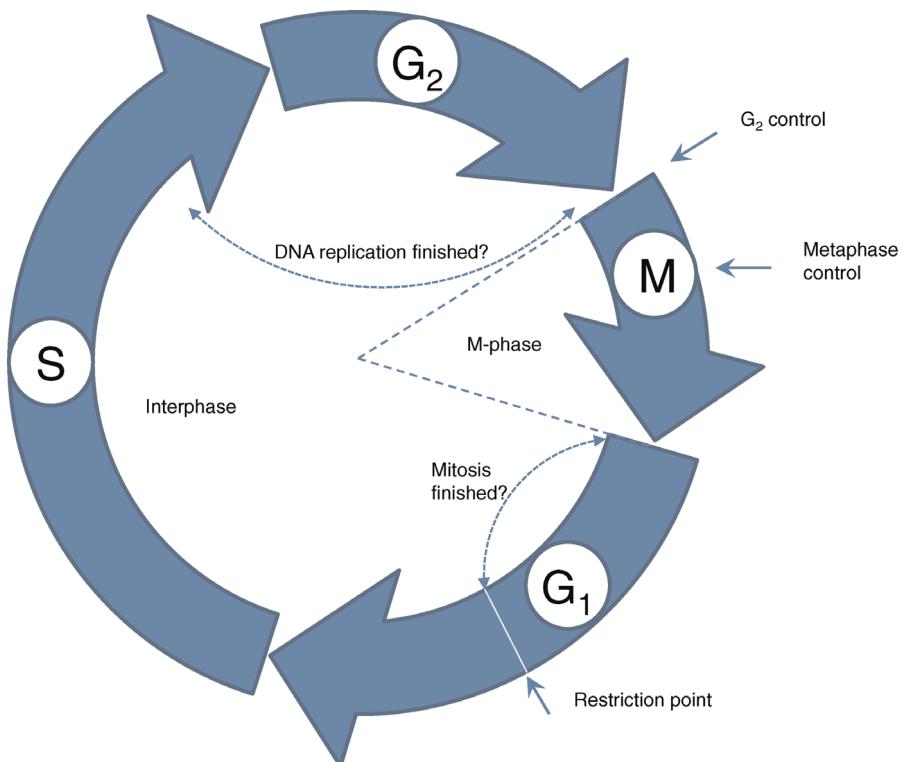


Figure 12.25 The cell cycle is divided into the interphase, which is the period between two subsequent cell divisions, and the M phase, during which one cell separates into two. Major control points of the cell cycle are indicated by arrows. More details are given in the text.

polymerase for the initialization of DNA polymerization, and a DNA ligase that is responsible for linkage of DNA fragments that are synthesized discontinuously on one of the two template strands, because of the limitation to $5' \rightarrow 3'$ synthesis. Like the DNA, also other cellular organelles have to be doubled, such as the centrosome involved in the organization of the mitotic spindle.

The cell cycle is divided into two major phases: the interphase and the M phase (Figure 12.25). The interphase is often a relatively long period between two subsequent cell divisions. Cell division itself takes place during M-phase and consists of two steps: first, the nuclear division in which the duplicated genome is separated into two parts, and second, the cytoplasmic division or cytokinesis, where the cell divides into two cells. The latter not only allocates the two separated genomes to each of the newly developing cells, but also distributes the cytoplasmic organelles and substances between them. Finally, the centrosome is replicated and divided between both cells as well.

DNA replication takes place during interphase in the so-called S phase (S=synthesis) of the cell cycle (Figure 12.25). This phase is usually preceded by a gap phase, G_1 , and followed by another gap phase, G_2 . From G_1 phase, cells can also leave the cell cycle and enter a rest phase, G_0 . The interphase normally represents 90% of the cell cycle length. During interphase, the chromosomes are dispersed as chromatin in the nucleus. Cell division occurs during M phase, which follows the G_2 phase, and consists of mitosis and cytokinesis. Mitosis is divided into different stages. During the first stage – the prophase – chromosomes condense into their compact form and the two centrosomes of a cell begin recruiting microtubules for the formation of the mitotic spindle. In later stages of mitosis, this spindle is used for the equal segregation of the chromatids of each chromosome to opposite cellular poles. During the following prometaphase, the nuclear envelope dissolves and the microtubules of the mitotic spindle attach to protein structures, called kinetochores, at the centromeres of each chromosome. In the following metaphase, all chromosomes line up in the middle of the spindle and form the metaphase plate. During anaphase, the proteins holding together both sister chromatids are degraded and each chromatid of a chromosome segregates into opposite directions. Finally, during telophase, new nuclear envelopes are recreated around the separated genetic materials and form two new nuclei. The chromosomes unfold again into chromatin. The mitotic reaction is often followed by a cytokinesis where the cellular membrane pinches off between the two newly separated nuclei and two new cells are formed.

The cell cycle is strictly controlled by specific proteins. When a certain checkpoint, the restriction point, in the G_1 phase is passed, this leads to a series of specific steps

that end up in cell division. At this point the cell checks, whether it has achieved a sufficient size and whether the external conditions are suitable for reproduction. The control system ensures that a new phase of the cycle is only entered if the preceding phase has been finished successfully. For instance, to enter a new M phase it has to be assured that DNA replication during S phase has been correctly completed.

Passage through the eukaryotic cell cycle is strictly regulated by periodic synthesis and destruction of cyclins that bind and activate cyclin-dependent kinases (CDKs). The term “kinase” expresses that their function is phosphorylation of proteins with controlling properties. A contrary function is carried out by a “phosphatase.” It dephosphorylates a previously phosphorylated protein and thereby toggles its activity. Cyclin-dependent kinase inhibitors (CKI) also play important roles in cell-cycle control by coordinating internal and external signals and impeding proliferation at several key checkpoints.

The general scheme of the cell cycle is conserved from yeast to mammals. The levels of cyclins rise and fall during the stages of the cell cycle. The levels of CDKs appear to remain constant during cell cycle, but the individual molecules are either unbound or bound to cyclins. In budding yeast, one CDK (Cdc28) and nine different cyclins (Cln1–Cln3, Clb1–Clb6) are found that seem to be at least partially redundant. In contrast, mammals employ a variety of different cyclins and CDKs. Cyclins include a G_1 cyclin (cyclin D), S phase cyclins (A and E), and mitotic cyclins (A and B). Mammals have nine different CDKs (referred to as CDK1–9) that are important in different phases of the cell cycle. The anaphase-promoting complex (APC) triggers the events leading to destruction of the cohesions, thus allowing the sister chromatids to separate and degrades the mitotic cyclins. A comprehensive map of cell cycle-related protein–protein interactions can be found in large-scale reconstructions, that is, for budding yeast [45].

12.3.1 Steps in the Cycle

Let us take a course through the mammalian cell cycle starting in G_1 phase. As the level of G_1 cyclins rises, they bind to their CDKs and signal the cell to prepare the chromosomes for replication. When the level of S phase promoting factor (SPF) rises, which includes cyclin A bound to CDK2, it enters the nucleus and prepares the cell to duplicate its DNA (and its centrosomes). As DNA replication continues, cyclin E is destroyed, and the level of mitotic cyclins begins to increase (in G_2). The M phase-promoting factor (the complex of mitotic cyclins with the M-phase CDK)

initiates (1) assembly of the mitotic spindle, (2) breakdown of the nuclear envelope, and (3) condensation of the chromosomes. These events take the cell to metaphase of mitosis. At this point, the M-phase promoting factor activates the APC, which allows the sister chromatids at the metaphase plate to separate and move to the poles (anaphase), thereby completing mitosis. APC destroys the mitotic cyclins by coupling them to ubiquitin, which targets them for destruction by proteasomes. APC turns on the synthesis of G₁ cyclin for the next turn of the cycle and it degrades geminin, a protein that has kept the freshly synthesized DNA in S phase from being rereplicated before mitosis.

A number of checkpoints ensure that all processes connected with cell cycle progression, DNA doubling and separation, and cell division occur correctly. At these checkpoints, the cell cycle can be aborted or arrested. They involve checks on completion of S phase, on DNA damage, and on failure of spindle behavior. If the damage is irreparable, apoptosis is triggered. An important checkpoint in G₁ has been identified in both yeast and mammalian cells. Referred to as "Start" in yeast and as "restriction point" in mammalian cells, this is the point at which the cell becomes committed to DNA replication and completing a cell cycle [46–49]. All the checkpoints require the services of complexes of proteins. Mutations in the genes encoding some of these proteins have been associated with cancer. These genes are regarded as oncogenes. Failures in checkpoints permit the cell to continue dividing despite damage to its integrity. Understanding how the proteins interact to regulate the cell cycle has become increasingly important to researchers and clinicians when it was discovered that many of the genes that encode cell cycle regulatory activities are targets for alterations that underlie the development of cancer. Several therapeutic agents, such as DNA-damaging drugs, microtubule inhibitors, antimetabolites, and topoisomerase inhibitors, take advantage of this disruption in normal cell cycle regulation to target checkpoint controls and ultimately induce growth arrest or apoptosis of neoplastic cells.

For the presentation of modeling approaches, we will focus on the yeast cell cycle since intensive experimental and computational studies have been carried out using different types of yeast as model organisms. Mathematical models of the cell cycle can be used to tackle, for example, the following relevant problems:

- The cell seems to monitor the volume ratio of nucleus and cytoplasm and to trigger cell division at a characteristic ratio. During oogenesis, this ratio is abnormally small (the cells accumulate maternal cytoplasm), while after fertilization cells divide without cell growth. How is the dependence on the ratio regulated?

- Cancer cells have a failure in cell-cycle regulation. Which proteins or protein complexes are essential for checkpoint examination?
- What causes the oscillatory behavior of the compounds involved in the cell cycle?

12.3.2

Minimal Cascade Model of a Mitotic Oscillator

One of the first genes to be identified as being an important regulator of the cell cycle in yeast was *cdc2/cdc28* [50], where *cdc2* refers to fission yeast and *cdc28* to budding yeast. Activation of the *cdc2/cdc28* kinase requires association with a regulatory subunit referred to as a cyclin.

A minimal model for the mitotic oscillator involving a cyclin and the Cdc2 kinase has been presented by Goldbeter [51]. It covers the cascade of posttranslational modifications that modulate the activity of Cdc2 kinase during cell cycle. In the first cycle of the bicyclic cascade model, the cyclin promotes the activation of the Cdc2 kinase by reversible dephosphorylation. In the second cycle, the Cdc2 kinase activates a cyclin protease by reversible phosphorylation. The model was used to test the hypothesis that cell-cycle oscillations may arise from a negative feedback loop with delay, that is, that cyclin activates the Cdc2 kinase, while the Cdc2 kinase eventually triggers the degradation of the cyclin.

The minimal cascade model is represented in Figure 12.26. It involves only two main actors, cyclin and cyclin-dependent kinase. Cyclin is synthesized at constant rate, v_i , and triggers the transformation of inactive (M+) into active (M) Cdc2 kinase by enhancing the rate of a phosphatase, v_1 . A kinase with rate v_2 reverts this modification. In the lower cycle, the Cdc2 kinase phosphorylates a protease (v_3) shifting it from the inactive (X+) to the active (X) form. The activation of the cyclin protease is reverted by a further phosphatase with rate v_4 . The dynamics is governed by the following ODE system:

$$\begin{aligned} \frac{dC}{dt} &= v_i - v_d \frac{X \cdot C}{K_{md} + C} - k_d C, \\ \frac{dM}{dt} &= \frac{V_{m1} \cdot (1 - M)}{K_{m1} + (1 - M)} - \frac{V_{m2} \cdot M}{K_{m2} + M}, \\ \frac{dX}{dt} &= \frac{V_{m3} \cdot (1 - X)}{K_{m3} + (1 - X)} - \frac{V_{m4} \cdot X}{K_{m4} + X}, \end{aligned} \quad (12.35)$$

where C denotes the cyclin concentration; M and X represent the fractional concentrations of active Cdc2 kinase and active cyclin protease, respectively, while $(1 - M)$ and $(1 - X)$ are the fractions of inactive kinase and phosphatase, respectively. K_m values are Michaelis constants (Chapter 4). $V_{m1} = V_1 C / (K_{mc} + C)$ and $V_{m3} = V_3 \cdot M$

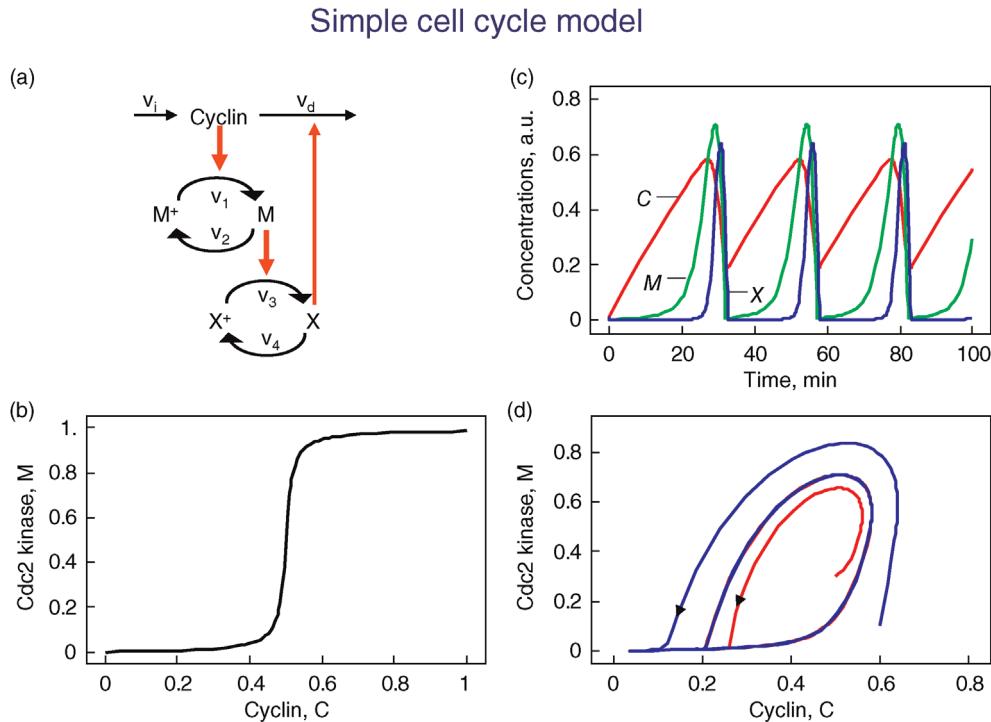


Figure 12.26 Goldbeter’s minimal model of the mitotic oscillator. (a) Illustration of the model comprising cyclin production and degradation, phosphorylation and dephosphorylation of Cdk1, and phosphorylation and dephosphorylation of the cyclin protease (see text). (b) Threshold-type dependence of the fractional concentration of active Cdk1 on the cyclin concentration. (c) Time courses of cyclin (C), active Cdk1 (M), and active cyclin protease (X) exhibition oscillations according to Eq. (12.35). (d) Limit cycle behavior, represented for the variables C and M . Parameter values: $K_{mi} = 0.05$, ($i = 1, \dots, 4$), $K_{mc} = 0.5$, $k_d = 0.01$, $v_i = 0.025$, $v_d = 0.25$, $V_{m1} = 3$, $V_{m3} = 1$, and $V_{m4} = 0.5$. Initial conditions in (b) and (c) are $C(0) = M(0) = X(0) = 0.01$. Units: μM and min^{-1} .

are effective maximal rates (Chapter 4). Note that the use of Michaelis–Menten kinetics in the differential equations for the changes of M and X leads to the so-called Goldbeter–Koshland switch or ultrasensitivity [52], that is, a sharp switch in the activity of the downstream component when the concentration of the activating component reaches the K_m value.

This model involves only Michaelis–Menten-type kinetics, but no form of positive cooperativity. It can be used to test whether oscillations can arise solely as a result of the negative feedback provided by the Cdc2-induced cyclin degradation and of the threshold and time delay involved in the cascade. The time delay is implemented by considering posttranslational modifications (phosphorylation/dephosphorylation cycles v_1/v_2 and v_3/v_4). For certain parameters, they lead to a threshold in the dependence of steady-state values for M on C and for X on M (Figure 12.26b). Provided that this threshold exists, the evolution of the bicyclic cascade proceeds in a periodic manner (Figure 12.26c). Starting from low initial cyclin concentration, this value accumulates at constant rate, while M and X stay low. As soon as C crosses the activation threshold, M rises. If M crosses the threshold,

X starts to increase sharply. X in turn accelerates cyclin degradation and consequently, C , M , and X drop rapidly. The resulting oscillations are of the limit cycle type. The respective limit cycle is shown in phase-plane representation in Figure 12.26d.

12.3.3 Models of Budding Yeast Cell Cycle

Tyson, Novak, Chen and colleagues have developed a series of models describing the cell cycle of budding yeast in great detail [53–56]. These comprehensive models employ a set of assumptions that are summarized in the following.

The cell cycle is an alternating sequence of the transition from G_1 phase to S/M phase, called “Start” (in mammalian cells it is called “restriction point”), and the transition from S/M to G_1 , called “Finish.” An overview is given in Figure 12.27.

The CDK (Cdc28) forms complexes with the cyclins Cln1 to Cln3 and Clb1 to Clb6, and these complexes control the major cell-cycle events in budding yeast cells. The complexes Cln1–2/Cdc28 control budding, the

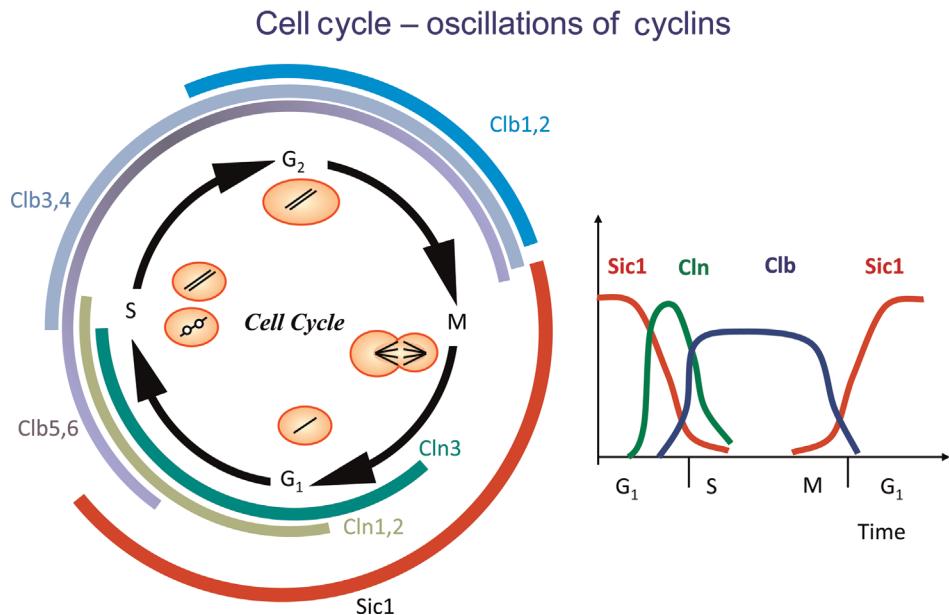


Figure 12.27 Schematic representation of the yeast cell cycle (inspired by Fall *et al.* [60]). The outer ring represents the cellular events. Beginning with cell division, it follows the G₁ phase. The cells possess a single set of chromosomes (shown as one black line). At “Start,” the cell goes into the S phase and replicates the DNA (two black lines). The sister chromatids are initially kept together by proteins. During M phase they are aligned, attached to the spindle body, and segregated to different parts of the cell. The cycle closes with formation of two new daughter cells. The inner part represents main molecular events driving the cell cycle comprising (1) protein production and degradation, (2) phosphorylation and dephosphorylation, and (3) complex formation and disintegration. For sake of clarity, CDK Cdc28 is not shown. The “Start” is initiated by activation of CDK by cyclins Cln2 and Clb5. The CDK activity is responsible for progression through S and M phase. At Finish, the proteolytic activity coordinated by APC destroys the cyclins and renders thereby the CDK inactive.

complex Cln3/Cdc28 governs the execution of the checkpoint “Start,” Clb5–6/Cdc28 ensures timely DNA replication, Clb3–4/Cdc28 assists DNA replication and spindle formation, and Clb1–2/Cdc28 is necessary for completion of mitosis.

The cyclin–CDK complexes are in turn regulated by synthesis and degradation of cyclins and by the Clb-dependent kinase inhibitor (CKI) Sic1. The expression of the gene for Cln2 is controlled by the transcription factor SBF, the expression of the gene for Clb5 is controlled by the transcription factor MBF. Both transcription factors are regulated by CDKs. All cyclins are degraded by proteasomes following ubiquitination. APC is one of the complexes triggering ubiquitination of cyclins.

For the implementation of these processes in a mathematical model, the following points are important. Activation of cyclins and cyclin-dependent kinases occurs in principle by the negative feedback loop presented in Goldbeter’s minimal model (see Section 12.3.2). Furthermore, these models are based on the assumption that the cells exhibit exponential growth, that is, the dynamics of the cell mass M is governed by $dM/dt = \mu M$. At the instance of cell division, M is replaced by $M/2$. In some cases, uneven division is considered. Cell growth implies

adaptation of the negative feedback model to growing cells.

The transitions “Start” and “Finish” characterize the wild-type cell cycle. At “Start,” the transcription factor SBF is turned on and the levels of the cyclins Cln2 and Clb5 increase. They form complexes with Cdc28. The boost in Cln2/Cdc28 has three main consequences: it initiates bud formation, it phosphorylates the CKI Sic1 promoting its disappearance, and it inactivates Hct1, which in conjunction with APC was responsible for Clb2 degradation in G₁ phase. Hence, DNA synthesis takes place and the bud emerges. Subsequently, the level of Clb2 increases and the spindle starts to form. Clb2/Cdc28 inactivates SBF and Cln2 decreases. Inactivation of MBF causes Clb5 to decrease. Clb2/Cdc28 induces progression through mitosis. Cdc20 and Hct1, which target proteins to APC for ubiquitination, regulate the metaphase–anaphase transition. Cdc20 has several tasks in the anaphase. It activates Hct1, promoting degradation of Clb2, and it activates the transcription factor of Sic1. Thus, at “Finish,” Clb2 is destroyed and Sic1 reappears.

The dynamics of some key players in cell cycle according to the model given in Chen *et al.* [54] is shown in Figure 12.28 for two successive cycles. At “Start,” Cln2

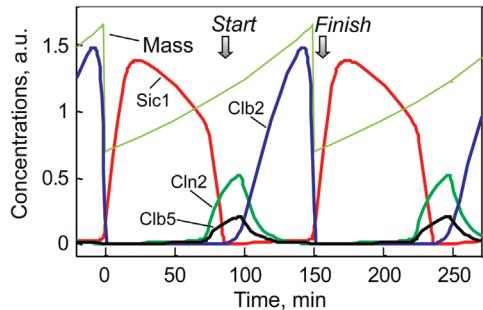


Figure 12.28 Temporal behavior of some key players during two successive rounds of yeast cell cycle. The dotted line indicates the cell mass that halves after every cell division. The levels of Cln2 , $\text{Clb2}_{\text{total}}$, $\text{Clb5}_{\text{total}}$, and $\text{Sic1}_{\text{total}}$ are simulated according to the model presented by Chen *et al.* [54].

and Clb5 levels rise and Sic1 is degraded, while at “Finish,” Clb2 vanishes and Sic1 is newly produced.

Recent models consider that cell growth is determined by nutrient uptake, nutrient conversion, and the resulting metabolic capacity to synthesize proteins, and

are hence dependent on the cellular surface-to-volume ratio [57]. This results in individual growth dynamics between linear and exponential volume increase over time. Within a population context, this can elucidate why small young daughter cells need longer growth periods before division, while big mother cells divide again quickly, yet the population retains stable average cell sizes. This flexibility in cell-cycle length makes some specific assumptions dispensable about how to accomplish all cell-cycle steps in the given period, allowing to formulate self-oscillating models for the dynamics of the important components without artificial events at the end of one period (see Figure 12.29, [58]).

Cell cycle is strongly regulated by signaling pathways conveying information about external stresses (such as osmotic stress transmitted by the HOG pathway discussed in Section 12.2), about the nutritional status, or about the presence of mating partners (sensed via the pheromone pathway). The osmotic stress leads to pausing of cell cycle, providing time for adaptation such as production of osmotically active compounds, while the pheromone

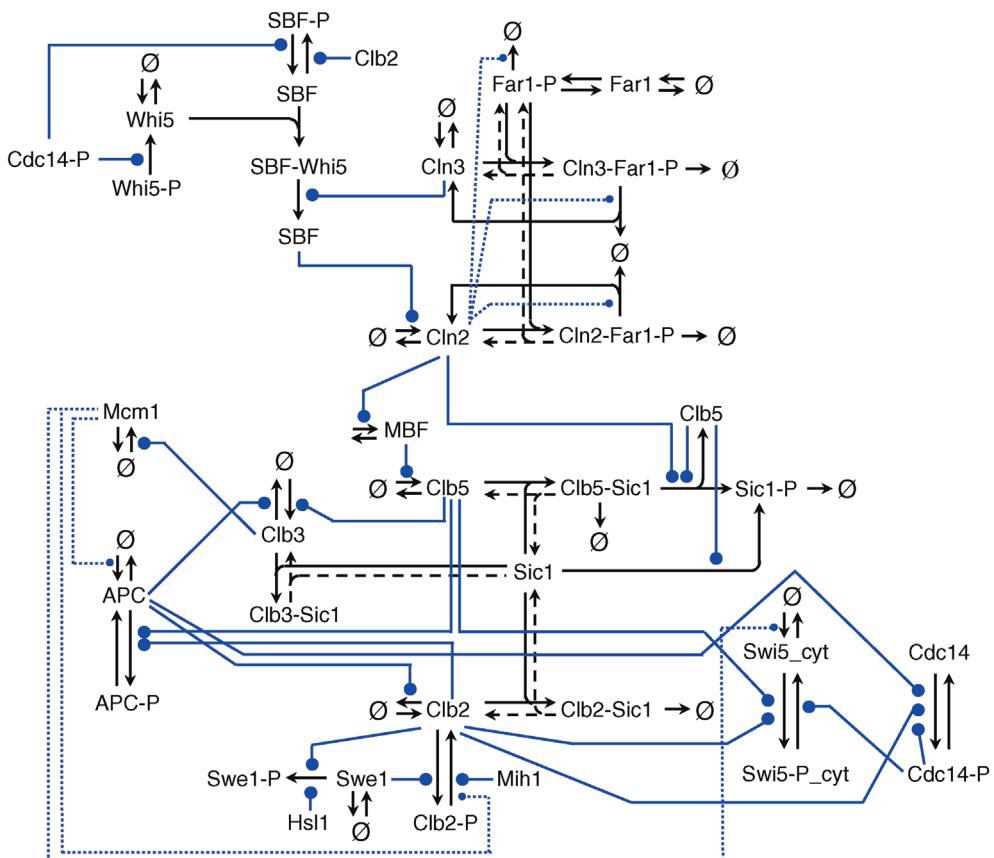


Figure 12.29 Self-oscillating network of yeast cell cycle. (a) The network comprises five cyclins (neglecting the potentially redundant cyclins Clb6 , Clb4 , and Clb1) as well as transcription factors SBF and MBF, the cyclin-dependent kinase inhibitors Far1 and Sic1, and other regulators. (b) Dynamics of selected components over three cell cycle periods.

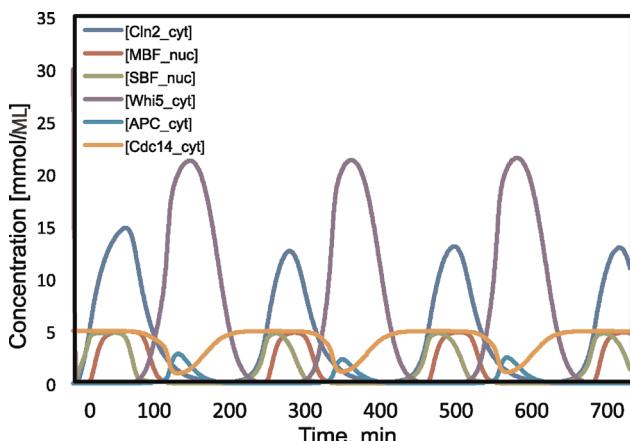


Figure 12.29 (Continued)

pathway triggers a complete cell cycle stop in early G1 phase allowing the cell to prepare for mating. Detailed models and experimentation helped to clarify the differential roles of transcriptional and posttranslational regulation by Hog1 in cell-cycle regulation upon osmotic stress [59].

12.4 The Aging Process

Summary

Aging is a complex biological phenomenon found in practically all multicellular organisms. It is manifested by a progressive and often exponential increase of an organism's mortality rate with time, which ultimately causes the death of the organism also in the absence of any predators or infectious diseases.

Although widespread, it is not trivial to understand why an organism should grow old and die. The challenge is to identify a selection advantage for such a trait and also to explain species-specific differences in life span. Many of the early explanations like avoidance of overcrowding or removal of worn out individuals were based on group selection. However, group selection only works under very special conditions and modern theories regard aging as an unavoidable or unimportant side effect of another trait on which selection works. The disposable soma theory, for instance, sees aging as the consequence of an optimal resource allocation process between maintenance and reproduction. Using a mathematical model, it can be shown that the optimal level of investment in maintenance is always smaller than what would be required for indefinite survival.

Apart from understanding why aging has evolved, it is also important to understand the actual biochemical mechanisms that underlie the aging process. Although

many different theories exist, free radicals and defective mitochondria are the most probable culprits. In old animals, mitochondria often contain damaged DNA and we will study the accumulation of damaged mitochondria using different hypotheses and different mathematical tools.

The aging process affects organisms at the molecular, cellular, and organ level and is therefore a prime candidate for systems biological modeling!

Aging is a complex biological phenomenon that practically affects all multicellular eukaryotes. It is manifested by an ever-increasing mortality risk, which finally leads to the death of the organism. Modern hygiene and medicine has led to an amazing increase in the average life expectancy over the last 150 years, but the underlying biochemical mechanisms of the aging process are still poorly understood. However, a better comprehension of these mechanisms is increasingly important, since the growing fraction of elderly people in the human population confronts our society with completely new and challenging problems. The aim of this chapter is to provide an overview of the aging process, discuss how it relates to system biological concepts, and to discuss some specific examples that make use of interesting modeling techniques such as delay differential equations and stochastic simulations.

What is Aging?

Looking at the enormous rise of average human life span over the last 150 years, one could get the impression that modern research actually has identified the relevant biochemical pathways involved in aging and has successfully reduced the pace of aging. Oeppen and Vaupel [61] collected data on world-wide life expectancy from studies going back to 1840. Figure 12.30 shows the life expectancy for males (squares) and females (circles) for the country that had the highest life expectancy at a given year. Two points are remarkable. First, there is an amazingly linear trend in life expectancy that corresponds to an increase of 3 months per year (!), and second, this trend is unbroken right to the end of the analyzed data (year 2000).

These impressive data strongly suggest that life span will also continue to rise in the next years, but it is not suitable to decide if the actual aging rate has fallen during the last century. Aging can best be described as a gradual functional decline, leading to a continuously rising risk to die within the next time interval (mortality). Already in the nineteenth century, Gompertz and Makeham studied the rise of human mortality with age and found that it can be described extremely well by an exponential function [62,63]. The Gompertz-Makeham equation, $\mu(t) = \mu_0 \cdot e^{\beta t} + \gamma$, models the increase of mortality with age depending on three parameters called intrinsic

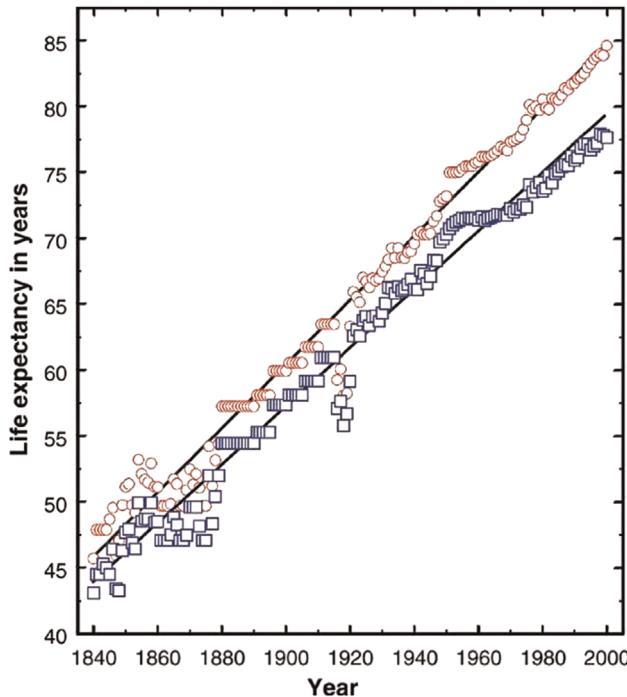


Figure 12.30 Male (blue squares) and female (red circles) life expectancy in the world record holding country between 1840 and 2000 based on the annual data of countries worldwide. (Reproduced with permission from Ref. [61].)

vulnerability (μ_0), actuarial aging rate (β), and environmental risk (γ). From this equation, an expression for the survivorship function can be derived (see Section 12.4.1 and [64]), which shows that the number of remaining survivors depends on all three parameters and consequently a change of the median life expectancy (time until 50% of the population has died) can be caused by a modification of any of those parameters. Analyzing the survivorship data of the last 100 years more closely, it becomes clear that the remarkable increase in life

expectancy was achieved exclusively by changes of intrinsic vulnerability and environmental risk. However, the aging rate, β , remained constant all the time. This is remarkable and important because it is actually β that most strongly controls life expectancy. Because of the drastic social, economical, and political consequences that are brought about by the demographic changes of the age structure of the population, it is now more important than ever to understand what constitutes the biochemical basis for a nonzero aging rate, β . Systems biology might help to achieve this goal.

Why is Aging a Prime Candidate for Systems Biology?

Evolutionary theories of the aging process explain why aging has evolved, but unfortunately they do not predict specific mechanisms to be involved in aging. As a consequence, more than 300 mechanistic ideas exist [65], each centered around different biochemical processes. This is partly due to the fact that even the simplest multicellular organisms are such complex systems that many components have the potential to cause deterioration of the whole system in case of a malfunction. Figure 12.31 shows a small collection of the most popular mechanistic theories. The spatial arrangement of the diagram intends to reflect the various connections between the different theories. And it is exactly the large number of interactions that makes it so difficult to investigate aging experimentally and renders it ideal for systems biology. To understand this, we will look at a few examples.

The *Telomere Shortening Theory* is an important idea that has gained considerable support in the last 15–20 years. Telomeres are the physical ends of linear eukaryotic chromosomes and vital for the functioning of the cell [66,67]. It has been recognized for a long time that linear DNA has a replication problem, since DNA polymerases can only replicate in the 5'–3' direction and cannot start DNA synthesis *de novo* [68,69]. This inability

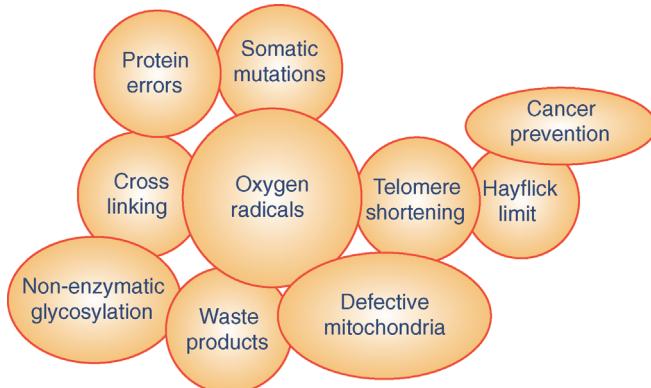


Figure 12.31 Graphical representation of some mechanistic theories of aging. The topology of the diagram reflects logical and mechanistic overlaps and points of interaction between different theories.

leads to a gradual loss of DNA, which was confirmed experimentally for human fibroblasts in 1990 and consequently proposed as being responsible for aging [70,71]. Telomere shortening provides the explanation and connection to the *Hayflick Limit*, the long known phenomenon that most cultured cell types have only a limited division potential [72]. This in turn can be interpreted as evolutionary selected trait that acts as a *cancer prevention system* by preventing unlimited cell division. While the direct cause of telomere shortening is the lack of the enzyme telomerase, there is an interaction to *oxygen radicals* that complicates the mechanism. Oxidative stress was shown to increase the rate of telomere shortening and thus modulating the telomere attrition [73]. As the figure indicates, free oxygen radicals are also the central hub to several other prominent phenomena affecting the aging process. They damage all kinds of macromolecules, leading to *cross-linking* of proteins and the generation of indestructible *waste products* (i.e., lipofuscin) that accumulate in slowly dividing cells [74]. Degraded mitochondria are supposed to be a major fraction of these waste products and certain oxygen radicals are known to damage the mitochondrial membrane. It seems, however, that radical-induced somatic mutations of the mitochondrial DNA (mtDNA) are the main route to *defective mitochondria* that produce less energy, but generate more radicals [75–77]. These mutants are capable of taking over the mitochondrial population of the cell, causing a chronic energy deficiency and maybe aging. It is still unclear what the selection pressure and mechanism is that leads to the accumulation of defective mitochondria, but several suggestions have been made [78–80].

Supporting evidence has been found for all the above-mentioned ideas and the many interdependencies make aging a phenomenon that is very difficult to study experimentally. If a single mechanism is studied in isolation, it is hard to interpret the results that were obtained without the contribution of the other mechanisms. And if a complex system is studied with all involved pathways, it is expensive, technically demanding, and the results are difficult to interpret because of the large number of factors that might have influenced the results.

This is exactly the situation where a systems biological approach is useful. Systems biology aims at investigating the components of complex biochemical networks and their interactions, applying experimental high-throughput and whole genome methods, and integrating computational and mathematical methods with experimental efforts. The growing number of high-throughput techniques that have been developed in the last years is a major driving force behind the wish to utilize computational methods to manage and interpret the high data

output. Modelers, on the other side, are keen to use the generated data to develop quantitative models of systems with complex interactions. Because of the large number of parameters, such models would not be meaningful without sufficient experimental measurements.

In addition, quantitative modeling of complex systems has several benefits. First of all, it requires that each aspect of a verbal hypothesis is being made specific. Before a computational model can be developed, the researcher has to define each component and how it interacts with all the other components. This is a very useful exercise to identify gaps in current knowledge and in the verbal model. It helps to complete the conceptual model, respectively motivates experiments to collect the missing experimental information. To understand complex systems with components that produce opposing effects, it is essential to have a model with quantitative predictions. Purely qualitative models (such as verbal arguments) are not sufficient to decide how a system develops over time if it contains nonlinear opposing sub-components. Computational models are also a convenient way to explore easily and cheaply “what if” scenarios that were difficult or impossible to test experimentally. What if a certain reaction would not exist? What if a certain interaction would be ten times stronger? What if we are interested in time spans too short or too long to observe in an experiment? First, we will study a model that deals with the evolution of the aging process (Section 12.4.1) followed by models that make use of stochastic modeling (Section 12.4.2) or use delay differential equations (Section 12.4.3) to understand the accumulation of damaged mitochondria.

12.4.1 Evolution of the Aging Process

It is not obvious why organisms should grow old and die. What is the selective advantage of this trait? And if aging is advantageous, why have different species such widely differing life spans?

The first attempt to explain the evolution of aging was made by Weismann [81]. He proposed that aging is beneficial by removing crippled and worn-out individuals from the population and thus making space and resources available for the next generation. This type of reasoning is very similar to other suggestions like the prevention of overcrowding or the acceleration of evolution by decreasing the generation time.

These ideas suggest that aging itself confers a selective advantage and that the evolution of genes that bring life to an end is an adaptive response to selective forces. All these theories have in common that they rely on group

selection, the selection of a trait that is beneficial for the group, but detrimental to the individual. However, group selection only works under very special circumstances like small patch size and low migration rates [82].

The weaknesses of adaptive theories have been recognized for some time and newer theories are no longer based on group selection, but on the declining force of natural selection with time. This important concept refers to the fact that, even in the absence of aging, individuals in a population are always at risk of death due to accidents, predators, and diseases. For a given cohort, this leads to an exponential decline over time in the fraction of individuals that are still alive. Events (e.g., biochemical processes) that occur only in chronologically old individuals will therefore only affect a small proportion of the whole population. The later the onset of the events, the smaller the involved fraction of the population.

Medawar [83] was the first to present a theory for the evolution of the aging process based on this idea. His "Mutation Accumulation" theory states that aging might be caused by an accumulation of deleterious genes that are only expressed late in life. Because of the declining force of natural selection, only a small part of the population would be affected by this type of mutation and the resulting selection pressure to remove them would only be very weak. Mutations with a small selection pressure to be removed can persist in a mutation-selection balance [84] and thus explain the emergence of an aging phenotype.

Another theory of this kind is the "Antagonistic Pleiotropy" theory [85]. Genes that affect two or more traits are called pleiotropic genes, and effects that increase fitness through one trait at the expense of a reduced fitness of another trait are antagonistic. Now, consider a gene that improves the reproductive success of younger organisms at the expense of the survival of older individuals. Because of the declining force of natural selection, such a gene will be favored by selection and aging will occur as a side effect of the antagonistic pleiotropy property of this gene. Possible candidate genes might be found in males and females. Prostate cancer appears frequently in males at advanced ages, but can be prevented by administration of female hormones or castration. It seems to be a consequence of long-term exposure to testosterone, which is necessary for male sexual and thus reproductive success. In older females, osteoporosis is mediated by estrogens that are essential for reproduction in younger women. In both cases, gene effects that are beneficial at younger ages have negative consequences later in life.

Genes that trade long-term survival against short-term benefit are probably the strongest candidates to explain

the aging process. A specific version of this hypothesis that connects evolutionary concepts with molecular mechanisms is the "Disposable Soma" theory [86,87]. The theory realizes that organisms have a finite energy budget (food resources) that must be distributed among different tasks like growth, maintenance, and reproduction. Energy spent for one task is not available for another. Organisms have to solve this resource allocation problem such that evolutionary fitness is maximized. On the basis of quite general assumptions, a mathematical model can be constructed that describes the relationship between investment in maintenance and fitness. We are going to have a closer look at this model as an example how to formulate a mathematical description of such a qualitative idea.

To get started, we need a mathematical concept of fitness. A standard measure that is often used in population genetics is the intrinsic rate of natural increase, r , (also called Malthusian parameter) that can be calculated by numerically solving the Euler–Lotka equation (12.36). To calculate r for a given genotype, the survivorship function, $l(t)$, and the fertility function, $m(t)$, have to be known. $l(t)$ denotes the probability that an individual survives to age t and $m(t)$ is the expected number of offspring produced by an individual of age t .

$$\int_0^{\infty} e^{-r \cdot t} \cdot l(t) \cdot m(t) dt = 1. \quad (12.36)$$

If the value of r that solves this equation is negative, it implies a shrinking population, if it is positive, the population grows. Thus, the larger r , the higher is the fitness. An exact derivation of the Euler–Lotka equation is outside the scope of this book, but can be found in Maynard Smith [88] or Stearns [89]. Investment in somatic maintenance and repair will affect both, survivorship and fertility, and the question remains whether there is an optimal level of maintenance that maximizes fitness. Unfortunately, the precise physiological trade-offs are unknown, so we have to develop some qualitative relationship. It was already mentioned that in many species mortality increases exponentially according to the Gompertz–Makeham equation (12.37) [63]. μ_0 , β , and γ represent basal vulnerability, actuarial aging rate, and age-independent environmental mortality, respectively.

$$\mu(t) = \mu_0 \cdot e^{\beta \cdot t} + \gamma. \quad (12.37)$$

Mortality and survivorship are connected via the relation $dl/dt = -\mu(t) \cdot l(t)$. By solving this equation, we obtain an expression for $l(t)$ that depends on two

parameters that are influenced by the level of maintenance, μ_0 and β (12.38). We now define the variable ρ to be the fraction of resources that are allocated for maintenance and repair, $\rho=0$ corresponding to zero repair and $\rho=1$ corresponding to the maximum that is physiologically possible. We also make the assumption that above a critical level of repair, ρ^* , damage does not accumulate and the organism has reached a nonaging state. The rational for this postulation is the idea that aging is caused by the accumulation of some kind of damage and by investing more in repair the accumulation rate is slowed down until finally the incidence rate is equal to the removal rate, in which case the physiological steady state can be maintained indefinitely. The modifications to μ_0 (12.39) and β (12.40) are only one way to implement the desired trade-off (decreasing μ_0 and β with increasing ρ), but in qualitative models like this, the main results are often very robust with regard to the exact mathematical expression used.

$$l(t) = e^{(1-e^{\beta t})\mu_0/\beta - \gamma^* t}, \quad (12.38)$$

$$\mu_0 = \mu_{\min}/\rho, \quad (12.39)$$

$$\begin{aligned} \beta &= \beta_0 \left(\frac{\rho^*}{\rho} - 1 \right) & \rho \leq \rho^*, \\ \beta &= 0 & \rho > \rho^*. \end{aligned} \quad (12.40)$$

The level of maintenance does also influence fertility, $m(t)$. It is assumed that the age at maturation, a , will increase with rising ρ and that the initial reproductive rate, f , is a decreasing function of ρ . We furthermore assume that fertility declines due to age-related deterioration with the same Gompertzian rate term as survivorship. From these conditions, equations can be derived for fertility (12.41), age at maturation (12.42), and initial reproductive rate (12.43).

$$m(t) = f \cdot e^{(e^{\beta a} - e^{\beta t})\mu_0/\beta}, \quad (12.41)$$

$$a = a_0/(1-\rho), \quad (12.42)$$

$$f = f_{\max} \cdot (1-\rho). \quad (12.43)$$

Now we have all the information that are necessary to solve the Euler–Lotka equation for different values of repair, ρ , (Figure 12.32). The calculations confirm that there exists an optimal level of maintenance, ρ_{opt} , which results in a maximal fitness. Depending on the ecological niche (represented by environmental mortality, γ), the optimal amount of maintenance varies. In a risky environment, the optimum is shifted toward lower maintenance and a niche with low external mortality, selects for individuals with a relatively high investment in maintenance.

This fits quite nicely with biological observation. Species like mice or rabbits live in a high-risk environment and, as predicted, they invest heavily in offspring but have

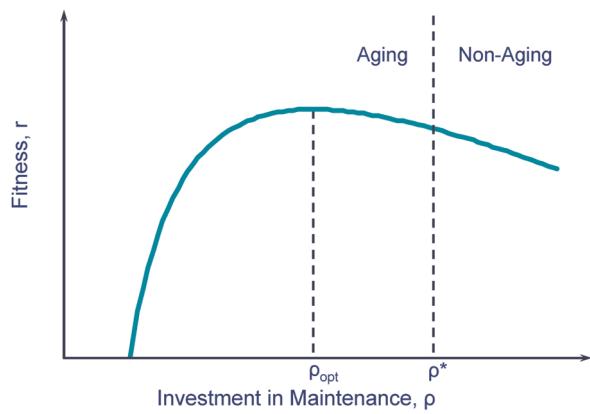


Figure 12.32 The disposable soma theory predicts that the optimal investment in maintenance is always less than what would be required to achieve a nonaging state. The exact position of this optimum (which maximizes fitness) depends on the environmental risk a species is exposed to over evolutionary times. Organisms living in a niche with heavy external mortality should invest less in maintenance than organisms that are exposed to little external mortality.

little left for maintenance. Consequently, their aging rate is high and life expectancy low (even under risk-free laboratory conditions). Humans or elephants, by contrast, inhabit a low-risk environment, spend fewer resources for offspring, and invest more in repair. Especially instructive are birds, which live two to three times as long as mammals of comparable body weight. Again, this long life span can be predicted by the enormous reduction of external mortality that accompanies the ability to fly and thus escape predators or starvation.

Another important result of the model is that the optimal level of maintenance and repair is always below the critical level, ρ^* , that is required for the evolution of a nonaging organism. This result can also be understood intuitively. Since all species have an external mortality that is above zero, they have a finite life expectancy, even without aging. This means, even though it might be physiologically possible to have such an efficient repair system that damage does not accumulate with time, resulting in a potentially immortal organism, this never results in a maximal fitness value. Only in the limit of $\gamma=0$, ρ_{opt} approaches ρ^* .

12.4.2 Using Stochastic Simulations to Study Mitochondrial Damage

The Biological Problem

It was already mentioned that defective mitochondria accumulate in old cells. These organelles developed

roughly two billion years ago from free-living prokaryotic ancestors [90]. This endosymbiotic origin also explains the fact that mitochondria still contain their own genetic material in form of a small circular DNA, named mtDNA. During the course of evolution most mitochondrial genes were transferred to the nucleus, but the genes for 13 proteins are still located on the human mtDNA [91]. Mitochondria are involved in several essential processes like apoptosis, calcium homeostasis, and fatty acid degradation, but their most important task is the generation of ATP via oxidative phosphorylation at the inner mitochondrial membrane. All of the proteins encoded on the mtDNA are involved in this process. Since ATP is essential for thousands of biochemical reactions, it could very well be that damage to the “powerhouses of the cell” is a major contributor to the aging process.

Many studies have shown that damage to mtDNA accumulates with age in various mammalian species such as rats, monkeys, and humans [92–97]. The most prominent type of damage seems to be deletion mutations, which have lost up to 50% of the total mtDNA sequence. These mutants then accumulate in a cell, overtaking the wild-type population of mitochondria. Cells containing damaged mitochondria can be identified by staining for enzymatic activity and Figure 12.33 shows a characteristic result of a muscle cross-section of a 38-months-old rat. Affected cells are typically negative for the mitochondrial encoded enzyme cytochrome-c oxidase (COX) and overexpress the nuclear-encoded enzyme succinate dehydrogenase (SDH). It can be seen that the cross-section displays a mosaic pattern with most cells being healthy and interspersed a few cells with effectively no COX activity at all. Furthermore, the experimental studies have shown that in

each COX negative cell, it is a single type of deletion mutant that is clonally expanded, and different COX negative cells are overtaken by different clonally expanded deletion mutants.

The challenge is now to identify a biochemical mechanism that can explain these experimental findings. That is, it has to explain (i) why deletion mutants accumulate in the presence of wild-type mtDNA, (ii) why it is always a single mutant and not several that are present in an affected cell (low heteroplasmy), and (iii) how such a mechanism can work for species with life spans ranging from 3 to 100 years. An interesting hypothesis that has been put forward and that we want to study in more detail is the idea that pure random drift might be sufficient to explain this clonal expansion [98,99]. Elson *et al.* [98] simulated a population of 1000 mtDNAs with a 10 day half-life and a deletion probability between 10^{-6} and 10^{-4} per replication event. They defined COX negative cells as those that contain more than 60% mutant mtDNA at the end of the simulation and showed that, after 120 years, between 1 and 10% COX negative cells had a degree of heteroplasmy compatible with experimental observations. However, can such a process also work for short-lived animals like rodents, which show a similar pattern of accumulation as in humans but on a greatly accelerated time scale [92,96]?

Modeling Approach

How can such an idea be modeled mathematically? This is an interesting problem, since the standard approaches are not applicable in this special case. Most often the time course of the biological entities to be modeled (molecules, cells, and organisms) is described by differential equations, which are then solved analytically or more often

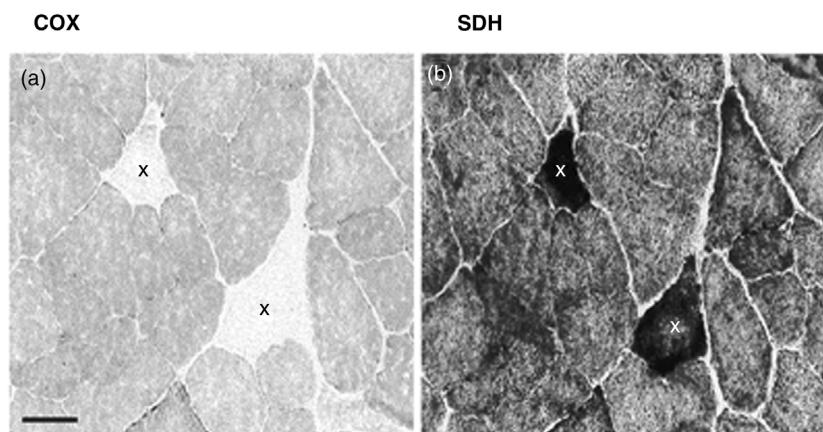


Figure 12.33 Enzymatic staining for COX and SDH activity of a muscle cross section from a 38-month-old rat. The muscle fibers marked by “X” show a COX⁻ (a) and SDH⁺⁺ (b) phenotype, while the neighboring cells present the healthy wild-type phenotype. The black bar represents 50 µm. (Reproduced with permission from Ref. [92].)

numerically by a suitable software tool. However, we cannot use differential equations to describe the fate of wild-type and mutant mtDNAs because the hypothesis assumes that all mtDNAs have identical parameter values for growth and degradation. Thus, in the deterministic case, the number of mutant mtDNAs would always remain at their initial value after the mutation event, which is 1. So, we have here a situation where the system must be modeled stochastically, since the proposed explanation relies completely on random effects. Normally we would use one of the simulation tools that are capable to perform stochastic simulations such as Copasi or VirtualCell, but it turns out that this, too, is not possible in our situation. The problem is that we have an undetermined number of variables. Each time a deletion event happens, it creates a new type of mutant, that is, per definition, different from all currently existing ones in the cell. This effectively creates a new variable in our model, whose time course we need to follow. Furthermore, there is no way to know how many of these new variables will be created during the course of the simulation. This behavior cannot be handled by classical tools for stochastic simulations.

Stochastic simulations only generate single trajectories and therefore it is necessary to perform many repetitions to calculate meaningful statistics (e.g., fraction of COX negative cells). At line 121, this outer loop starts, which performs *Nrepeats* repetitions. In the next lines, important program variables are initialized for the next trajectory simulation, which starts at line 130. A central step during program development is the choice of suitable data structures to hold the state variables of the system. In this example, *mitoL* is a simple list of integers that holds the number of wild-type mtDNAs followed by the numbers of one or more mutant mtDNAs. The simulation progresses in fixed time intervals of 1 h until the simulation time is reached or all wild-type mtDNAs are lost (line 130). This is similar to the τ -leap method (see Section 7.2.3) with the difference that the time interval is much shorter than the average time interval of the modeled reactions (mtDNA degradation & replication). In lines 131–137, the program collects and stores information about the number of type of mutants in *ergL*. This happens only every 30 days to keep the amount of data at a manageable size.

```

120 // Do Nrepeats simulations
121 for (r=0; r<Nrepeats; r++) {
122     _guiObj.print(String.format("Repetition: %d\n", r));
123     t = 0;
124     wtGone = false;
125     ergL.clear();
126     mitoL.clear();
127     mitoL.add(mitoNumber); // initialize mtDNA population
128     // Loop until simTime is over or all wt mtDNAs are lost.
129     // Each iteration is equal to 1 hour
130     while (t<simTimeH && !wtGone) {
131         if (t%720==0) { // collect output info every month
132             itmp = mitoNumber-mitoL.get(0);
133             ergL.add(itmp); // total nr of mutants
134             if (itmp > sumMutantsMax) sumMutantsMax = itmp;
135             ergL.add(mitoL.size()); // number of mtDNA types
136             if (mitoL.size() > sumMutantTypesMax) sumMutantTypesMax = mitoL.size();
137     }

```

The only solution in such a case is to write a computer program that is tailor made for the specific problem. Exactly, this has been done by Kowald and Kirkwood [100] and in the following we will have a closer look at the source code. Stochastic simulations are always computationally demanding and thus a suitable programming language such as Java or C/C⁺⁺ should be chosen. In our case, a Java program was developed within the Eclipse IDE. We cannot discuss the whole program, but instead will concentrate and comment on the core number crunching routine.

In the “for loop” starting at line 141, the program iterates through all different types of mtDNA and the small loop from line 145–147 calculates how many molecules should be degraded during this time step and at line 148 the number of existing mtDNAs is reduced accordingly. This is the place where stochasticity enters the simulation. The “if clause” starting at 151 checks if all wild-type mtDNAs have been lost and performs some necessary bookkeeping in that case. Finally, line 164 removes elements from *mitoL* that are zero. These stem from entries for mutant mtDNAs that have been completely eradicated by degradation.

```

139 // Go through mtDNAs and test for degradation. E.g. mitoL = [990,8,2]
140 nDegTotal = 0; // count number of mtDNA that are degraded at this time step
141 for (i = 0; i<mitoL.size(); i++) { // for all mtDNA types
142     nDeg = 0;
143     nOrig = mitoL.get(i);
144     // for all molecules of this type
145     for (n=0; n<nOrig; n++) {
146         if (_rnd.nextDouble() < degProb) nDeg++;
147     }
148     mitoL.set(i, nOrig-nDeg);
149     nDegTotal += nDeg;
150     // did wt go extinct ?
151     if (nOrig-nDeg==0 && i==0) {
152         _guiObj.print(String.format("Months %d: wt extinct\n",t/720));
153         wtGone = true;
154         nWtExtinct++;
155         // extend ergL to normal length
156         itmp = (int)(simTime*360/30)-ergL.size()/2;
157         for (i=0; i<itmp; i++) {
158             ergL.add(mitoNumber);
159             ergL.add(mitoL.size()-1); // take care of zero in mitoL
160         }
161     } // if wt extinct
162 } // for i
163 // Remove zero elements that stem from mutant removal. E.g. [990,0,2]
164 while (mitoL.remove(new Integer(0)));

```

The model implements a very simple mechanism for replacing degraded mtDNAs. Exactly the same number that is degraded, is newly synthesized via replication, such that the total number of mtDNAs is always equal to *mito-Number*. This is a typical example how certain aspects of biology are simplified for mathematical modeling because they are not relevant for the modeled problem or because more details are not known. Within the “for loop” starting at line 168, the new mtDNAs are created by making a

copy of a randomly chosen (line 169) parent mtDNA. Since mtDNAs are arranged by types within *mitoL*, some lines of code (171–175) are necessary to identify the parent type. During the replication process a deletion can happen, creating a new mutant type. This depends on the parameter *mutProb* and is implemented by lines 177–181. That completes the computations for this time step and in line 185 the number of simulated hours is increased by one until *simTimeH* has been reached.

```

166 // Replenish degraded mtDNAs by copying randomly selected parent mtDNA
167 sumMito = mitoNumber-nDegTotal;
168 for (n=0; n<nDegTotal; n++) {
169     parentIdx = _rnd.nextInt(sumMito); // index of parent
170     // Search parent in mitoL. E.g. [990,0,2]
171     i=0;
172     itmp = mitoL.get(i);
173     while (itmp <= parentIdx) {
174         itmp += mitoL.get(++i);
175     }
176     // We found parent, now copy it
177     if (_rnd.nextDouble() < mutProb) { // mutation
178         mitoL.add(1);
179     } else { // no mutation
180         mitoL.set(i, mitoL.get(i)+1);
181     }
182     sumMito++;
183 } // for n
184 ] ]
185 t++;
186 } // while
187 // Single simulation has finished

```

If the wild type has been lost or we have less than 40% wild-type mtDNAs (line 189), we keep track of this by incrementing some counter variables (190–199). Finally, we write the data that were collected on a monthly basis to an output file (204–209). Thus, for each repetition one long line of text is generated with the monthly count of

```

188 // Did this simu result in COXneg cell as defined by Elson01 (>60% mt) ?
189 if (wtGone || mitoL.get(0)<0.4*mitoNumber) {
190     sumCOXneg++;
191     mutTypesInCOXneg += ergL.get(ergL.size()-1);
192     // How freq is most prominent mutant?
193     // If wtGone, wt entry has already been deleted !
194     itmp = 0;
195     for (i=wtGone?0:1; i<mitoL.size(); i++) {
196         if (mitoL.get(i)>itmp) itmp = mitoL.get(i);
197     }
198     mostFreqMutant += itmp;
199 }
200 // add 1 or 0 to ergCOXnegL if simu resulted in COX neg cell
201 for (i=0; i<ergL.size()/2; i++)
202     ergCOXnegL.set(i, ergCOXnegL.get(i) + (int)(ergL.get(2*i)/(0.6*mitoNumber)));
203 // Now write sumMutants & sumMutantTypes in single line
204 out = String.format("sumMutants%03d", r);
205 for (i=0; i<ergL.size()/2; i++) out += String.format(";%d", ergL.get(2*i));
206 fp.println(out);
207 out = String.format("sumMutantTypes%03d", r);
208 for (i=0; i<ergL.size()/2; i++) out += String.format(";%d", ergL.get(2*i+1));
209 fp.println(out);
210 } // for Nrepeats

```

the total number of mutants and another long line with the monthly count of the number of different mutant types. This concludes our inspection of the simulation code and in the next section we look at some of the results that can be obtained from the simulation data.

Results

The output of the simulation program is a collection of trajectories describing how many mutant mtDNAs accumulate with time and to how many different mutant types they belong. Figure 12.34(a) shows three typical trajectories for a simulation lasting for 120 years. During one simulation (black), some mutant mtDNAs appear after approx. 20 years and reach roughly 200 copies, which correspond to 20% of the total mtDNA population. However, random fluctuations cause these mutants to disappear again and after 120 years this simulation ends with a healthy cell, containing no mutant mtDNAs. The simulation shown in red displays a completely different behavior. After about 60 years, there is a sharp rise in mutant mtDNAs that completely replace the wild type

around year 95. Once all wild-type mtDNAs are lost, the simulation has reached an absorbing boundary condition since there is no way to recreate wild-type mtDNAs.

By calculating a large number of trajectories, interesting statistical insights can be obtained. Figure 12.34(b) illustrates the increase in COX negative cells calculated

from 3000 ($P_{\text{mut}}=10^{-4}$ and 10^{-5}) respectively 15 000 trajectories ($P_{\text{mut}}=10^{-6}$). The larger number of simulations for the low mutation rate is necessary to obtain a “smooth” curve. Experimentally, it is known that around 4% of postmitotic human cells are COX negative after 80 years [93,101], which means that a mutation rate of 10^{-5} – 10^{-4} per replication is compatible with these results.

As mentioned earlier an important experimental observation is the low degree of mtDNA heteroplasmy for species with widely different life spans. To study this phenomenon, stochastic simulations were performed for different life spans (3, 10, 40, 80, and 120 years) and the average number of different mtDNA types per COX negative cell is shown in Figure 12.35. Since the level of COX negative cells in old individuals of species like rat and man is similar, the mutation rate was adjusted such that at the end of the simulation time the 3000 trajectories yielded 10% of COX negative cells.

From the diagram, it is immediately obvious that the number of different mtDNA types per COX negative cell depends strongly on the simulated life span. If random drift can act over a time of 120 years, there are on

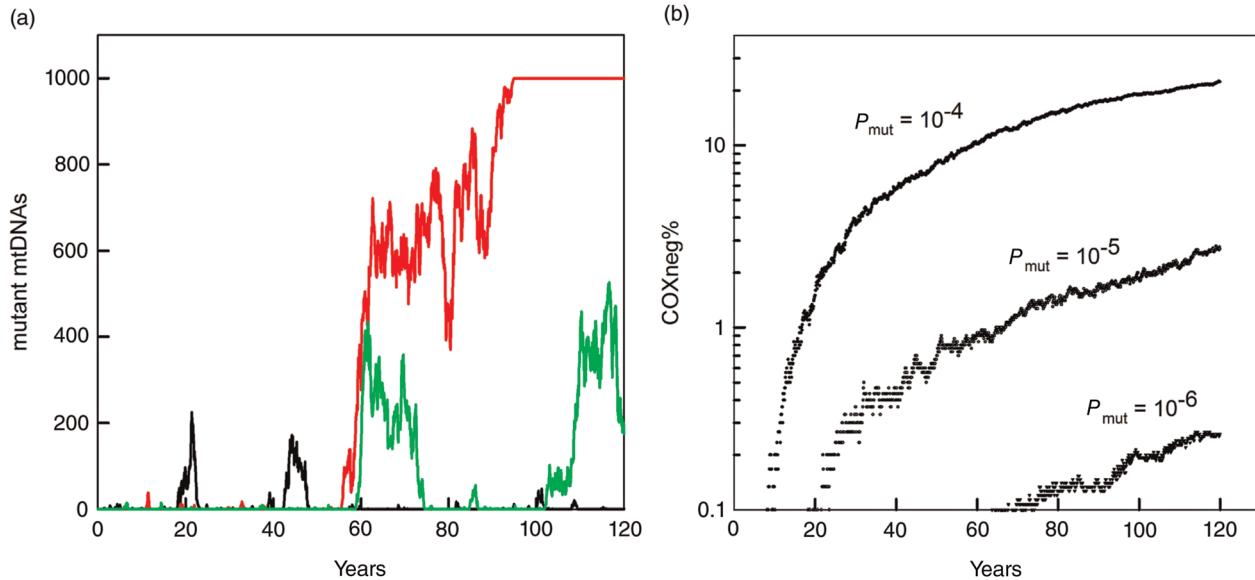


Figure 12.34 (a) Three single trajectories showing the appearance of mutant mtDNAs in a population that initially consists of 1000 wild-type molecules. After 120 years, the mutant mtDNA has either overtaken the population (red), has gone extinct (black) or exists at an intermediate level (green). mtDNA half-life is 10 days and the mutation probability per replication is 4×10^{-5} . (b) Increase in COX negative cells with time for three different mutation rates. A cell is defined as COX negative if it contains more than 60% mutant mtDNAs. The curves for $P_{\text{mut}} = 10^{-4}$ and 10^{-5} are calculated from 3000 trajectories and from 15 000 in case of $P_{\text{mut}} = 10^{-6}$.

average only 1.46 different mutant types per COX negative cell. A value that is in good agreement with experimental results obtained from humans. However, already for a life span of 40 years (e.g., rhesus monkeys), there are

2.8 mutant types per cell, hardly compatible with observation. And finally for a life span of 3 years (e.g., rats), the simulations predict more than 35 different mtDNA mutants per cell, completely incompatible with experimental studies. Thus, the simulation results are very clear and rule out random drift as possible explanation for the accumulation of mtDNA deletion mutants in short-lived species.

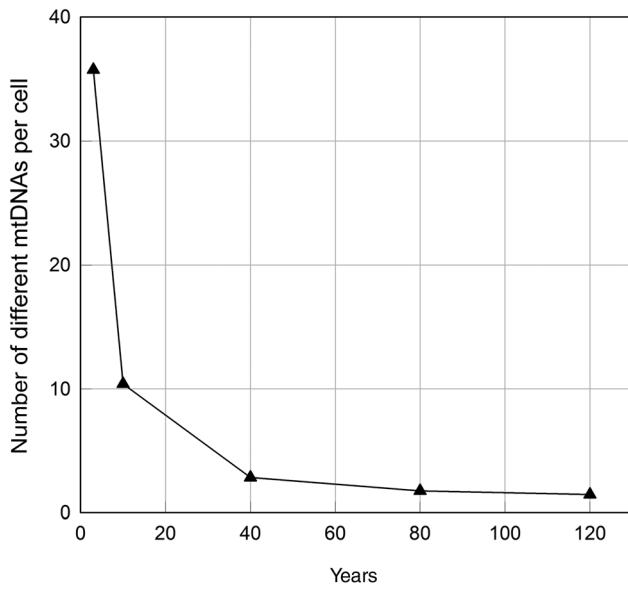


Figure 12.35 Number of different types of mtDNAs per COX negative cell depending on the simulation time (3, 10, 40, 80, and 120 years). For each simulation time a mutation rate was chosen that resulted in 10% COX negative cells. The data points are calculated from 3000 trajectories each.

12.4.3 Using Delay Differential Equations to Study Mitochondrial Damage

If random drift is not a likely explanation for the accumulation of mitochondrial deletion mutants, what else could be the mechanism? An interesting possibility that has been proposed is related to the reduced genome size of the deletion mutants [102,103]. If the reduced size directly translates into a reduced replication time, this should provide the mutants with a selection advantage. However, replication normally only occurs to replace mtDNAs that have been degraded via mitophagy. And since the time necessary for replication (1–2 h) [104] is much shorter than the half-life (1–3 weeks) [105,106], it is unclear if this idea can work. Figure 12.36 provides an overview of the model system. mtDNAs can either be in a “free” or a “busy” state. In the free state, they can respond to cellular signals and enter the busy state by starting a

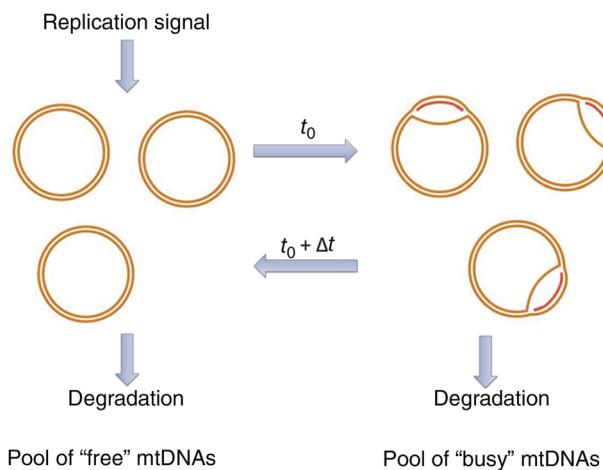


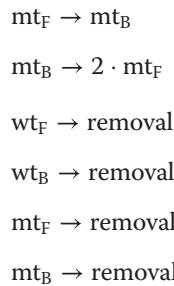
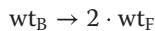
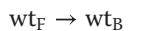
Figure 12.36 Selection advantage via a reduced genome size. If a replication signal is received by the pool of “free” mtDNA, some of them respond by starting replication and entering the “busy” state. After a replication time, Δt , two molecules of mtDNA are generated and return into the “free” pool. It is assumed that “free” as well as “busy” mtDNAs are degraded at the same rate, leading to the same half-life for wild-type and mutant molecules.

round of replication. After a certain delay, Δt , the original molecule plus a copy return into the free pool. Wild-type and mutant form respond with the same probability to replication signals, but deletion mutants will spend a shorter time in the busy state and hence return earlier into the free pool than wild-type mtDNAs. This is the source of the selection advantage.

Kowald *et al.* [107] have investigated this problem using delay differential equations as well as stochastic simulations, but we will concentrate here on the development of the system of DDEs. The model contains four variables that require four differential equations:

- wt_F : number of wild-type mtDNAs in the free pool.
- wt_B : number of wild-type mtDNAs in the busy pool.
- mt_F : number of mutant mtDNAs in the free pool.
- mt_B : number of mutant mtDNAs in the busy pool.

These variables can undergo the following set of reactions, which describe the transfer from one pool into the other, the increase in molecule number caused by replication and the removal of mtDNAs because of degradation.



Differential equations are a standard approach to model biochemical reactions. While ordinary differential equations (ODE) only refer to variable values at one time point, $x(t)$, delay differential equations (DDE) also refer to variable values at some time point in the past, $x(t-\Delta t)$. Because in our model replication needs a certain time span after which molecules reappear from the busy state, DDEs are required instead of ODEs.

Each of the four Eqs. (12.44)–(12.47) consists of three terms describing (i) the transfer of mtDNAs to or from the busy pool, (ii) the transfer of mtDNAs to or from the free pool, and (iii) the degradation process. Let us inspect the second term of (12.44) more closely. It describes the flux of wild-type mtDNAs from the free pool into the busy pool. This depends on the current amount of free mtDNAs, $wtF(t)$, and on how strongly mtDNAs respond to cellular replication signals. It is assumed that the replication probability declines with the total amount of existing mtDNAs. The rational is that some form of negative feedback exists that limits the production of new mtDNAs if already many exist. The mathematical expression that was used to model this behavior is of the form

$$\frac{replS}{(\sum \text{mtDNAs})carryC+1}$$
. This construct has the property that the replication probability is equal to the parameter $replS$ if only very few mtDNAs are present and drops to $replS/2$ when the number of mtDNAs has reached $carryC$. Now, we can also understand the first term that quantifies the flux of mtDNAs from the busy pool back into the free pool at the end of the replication process. This depends on the amount of free wild-type mtDNAs that started replication at $t-\Delta t$, which is given by $c_2 \cdot wtF(t-\Delta t)$. The factor 2 reflects the fact that each mtDNAs that starts replication is doubled in that process. The origin of the exponential expression, however, is not that obvious. It stems from the fact that also mtDNAs in the busy state are subject to degradation. As can be seen from the third term of the equation, a first-order decay is assumed, leading to an exponential decline. Therefore, the quantity of returning mtDNAs has to be diminished by this factor. The only difference between the equations for mutants and wild type is a shorter replication time, expressed as Δt times the size of the deletion (ranging from 0 to 1).

$$\frac{dwtF}{dt} = 2 \cdot c_2 \cdot wtF(t - \Delta t) \cdot e^{-\frac{\ln 2 \cdot \Delta t}{halfL}} - c_1 \cdot wtF(t) - \frac{\ln 2}{halfL} \cdot wtF(t), \quad (12.44)$$

$$\frac{dwtB}{dt} = -c_2 \cdot wtF(t - \Delta t) e^{-\frac{\ln 2 \cdot \Delta t}{halfL}} + c_1 \cdot wtF(t) - \frac{\ln 2}{halfL} \cdot wtB(t), \quad (12.45)$$

$$\frac{dmfF}{dt} = 2 \cdot c_2 \cdot mtF(t - \Delta t \cdot mutS) \cdot e^{-\frac{\ln 2 \cdot \Delta t \cdot mutS}{halfL}} - c_1 \cdot mtF(t) - \frac{\ln 2}{halfL} \cdot mtF(t), \quad (12.46)$$

$$\frac{dmfB}{dt} = -c_2 \cdot mtF(t - \Delta t \cdot mutS) \cdot e^{-\frac{\ln 2 \cdot \Delta t \cdot mutS}{halfL}} + c_1 \cdot mtF(t) - \frac{\ln 2}{halfL} \cdot mtB(t), \quad (12.47)$$

$$c_1 = \frac{replS}{(wtF(t) + wtB(t) + mtF(t) + mtB(t))/carryC + 1}, \quad (12.48)$$

$$c_2 = \frac{replS}{(wtF(t - \Delta t \cdot mutS) + wtB(t - \Delta t \cdot mutS) + mtF(t - \Delta t \cdot mutS) + mtB(t - \Delta t \cdot mutS))/carryC + 1}. \quad (12.49)$$

Results

Please note that the DDE model does not contain *de novo* mutations. It describes the competition between existing mutant and wild-type molecules. The original publication [107] also contains a stochastic model to take mutations into account, but we will restrict ourselves to the results of the deterministic DDE model. The software Mathematica was used to solve the equations numerically and Figure 12.37a shows the time course of wild-type

(wt) and deletion mutant (mt) for a set of parameters based on literature values. The simulation was started with 999 molecules of wild-type and a single mutant molecule, reflecting the situation immediately after a mutation event. The main conclusions that can be drawn are that the shorter replication time does indeed provide a selection advantage for the deletion mutant, but it takes almost 100 years before the wild type goes extinct (drops below one copy).

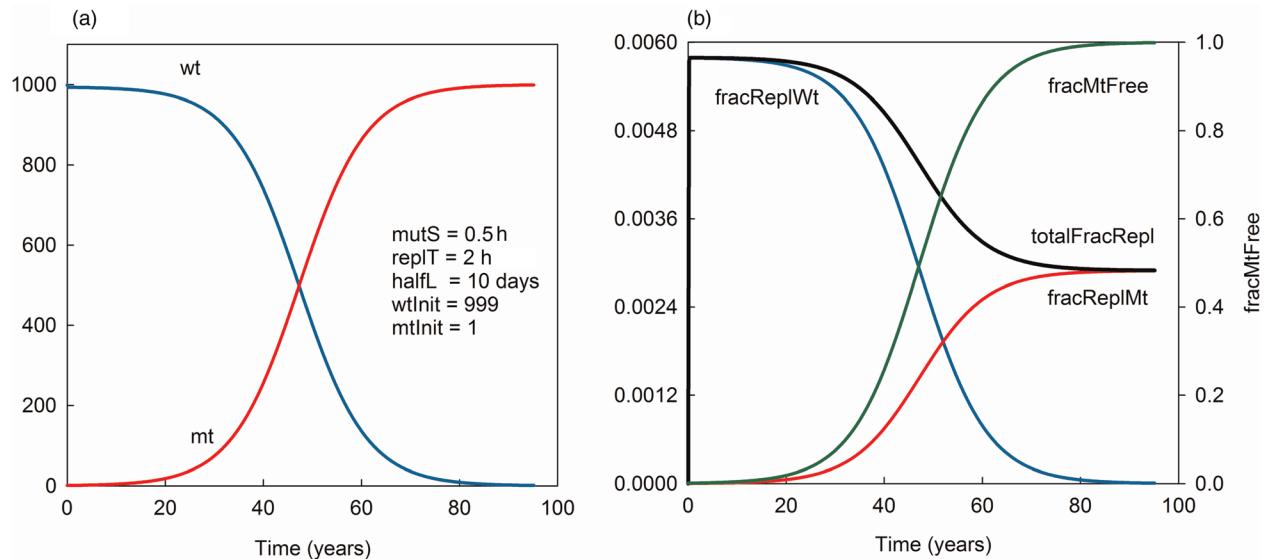


Figure 12.37 (a) Numerical solutions of the set of delay differential equations, showing the time course of wild-type (wt) and mutant (mt) mtDNA (aggregated values of free and busy pool). (b) Additional information such as the fraction of replicating wild-type molecules (fracReplWt), the fraction of replicating mutant molecules (fracReplMt), the total fraction of replicating mtDNAs (totalFracRepl), and the fraction of mutant molecules in the free pool (fracMtFree) are given.

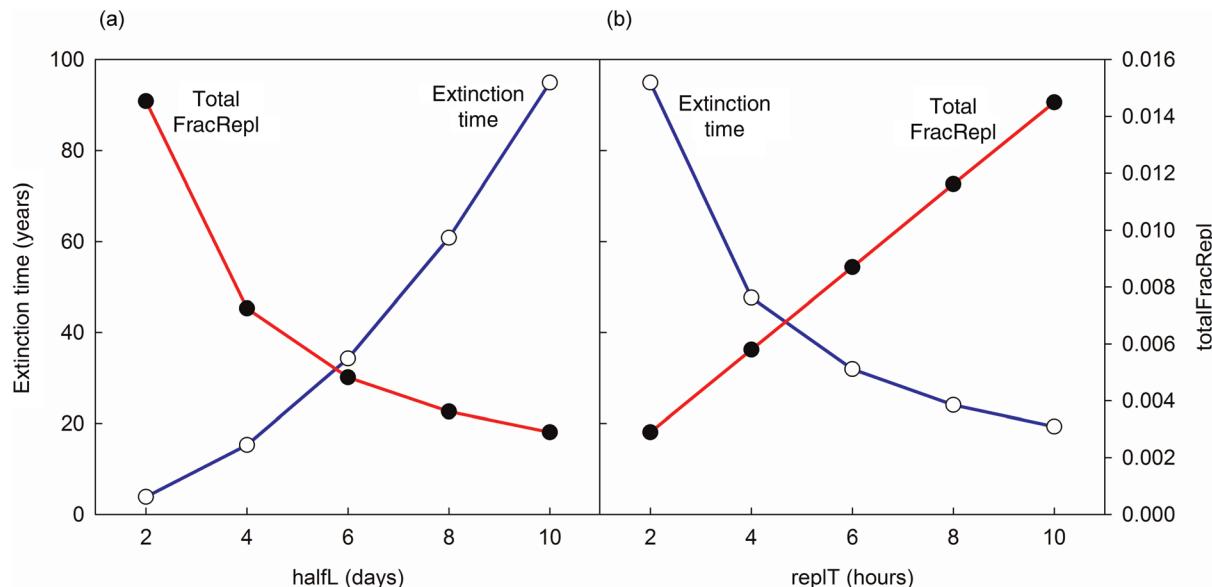


Figure 12.38 (a) Effects of varying the half-life (halfL) on the extinction time of the wild-type and the total fraction of replicating molecules. A replication time of 2 h was used. (b) Effects of varying the replication time (replT) on the extinction time of the wild-type and the total fraction of replicating molecules. A half-life of 10 days was used. Simulations were started with 999 wild-type and a single mutant mtDNA with 50% of the wild-type size.

To better understand the model behavior, it is necessary to see how extinction times depend on parameters such as the replication time or the half-life of mtDNAs. To make the results comparable between simulations, it is important that the steady-state level of the total mtDNA population remains the same, since it is very likely that extinction times also depend on the population size. One way to achieve this is to adjust the free parameter *carryC*, which controls the replication probability, accordingly. The relationship between the mtDNA steady-state level, M_{ss} , and the model parameters can be obtained by setting the left-hand side of the DDE system to zero and then solving it for the steady-state values of mtDNAs in the free and busy state. For this, it is important to realize that under steady-state conditions it holds that $x(t)=x(t-\Delta t)$. The sum of these values is M_{ss} and given by the following expression:

$$M_{ss} = -\frac{carryC \cdot (halfL \cdot (replS \cdot (1 - 2^{1-\Delta t/halfL})) + \ln 2)}{\ln 2},$$

And thus *carryC* is equal to:

$$carryC = \frac{-M_{ss} \cdot \ln 2}{halfL \cdot (replS \cdot (1 - 2^{1-\Delta t/halfL})) + \ln 2}.$$

Figure 12.38 shows the effects of varying half-life or replication time while maintaining a constant total population of mtDNAs. The simulations reveal that either shortening the half-life or increasing the time required for replication leads to a drastic reduction of the extinction time. Thus, the larger the ratio between half-life and replication time, the longer is the resulting extinction time. The reduced size of the mutant is only an advantage in the busy state (via a faster exit time) and consequently the selection advantage dwindles if the total fraction of replicating mtDNAs is decreasing. And as can be seen from Figure 12.38 increasing the half-life or decreasing the replication time diminishes this important parameter.

Thus, mathematical modeling confirms that it is very unlikely that mitochondrial deletion mutants gain a selection advantage from a faster replication time based on their reduced size.

Exercises

- 1) Implement an alternative model for glycolysis as shown in Figure 12.2. However, use Michaelis–Menten kinetics instead of mass actions as has been used for the simulations shown in Figure 12.3. How would the behavior change?
- 2) Calculate the flux control coefficients for the model of the threonine synthesis pathway. If required, use a computational tool providing the necessary functions.
- 3) Consider the Ras activation cycle shown in Figure 12.14 with the parameters given there. The concentration of GAP be 0.1. GEF gets activated according to $GEF = \begin{cases} 0, t < 0 \\ e^{-0.2t}, t \geq 0 \end{cases}$. Calculate the signaling time $\tau_{Ras_{GTP}}$ and the signal duration $\vartheta_{Ras_{GTP}}$ (Eqs. (12.29) and (12.30)).
- 4) MAP kinase cascades comprise kinases and phosphatases. How would such a cascade behave if there were no phosphatases?
- 5) In MAP kinase cascades, proteins have typically multiple phosphorylation sites. How would the model described in Figure 12.17 and Eqs. (12.14)–(12.22) change if we would include all potential phosphorylation sites and all sequences of phosphorylation and dephosphorylation events?
- 6) Signaling pathways often regulate the expression of genes coding for their own components. What kind of feedback does this impose? On which time scale would we expect an effect?

References

- 1 Olivier, B.G. and Snoep, J.L. *JWS Online: Online Cellular Systems Modelling [updated 20 August 2004]*. Available from: jjj.biochem.sun.ac.za.
- 2 Snoep, J.L. and Olivier, B.G. (2002) Java Web Simulation (JWS); a web based database of kinetic models. *Mol. Biol. Rep.*, 29 (1–2), 259–263.
- 3 Stanford, N.J., Lubitz, T., Smallbone, K., Klipp, E., Mendes, P., and Liebermeister, W. (2013) Systematic construction of kinetic models from genome-scale metabolic networks. *PLoS One*, 8 (11), e79195.
- 4 Hynne, F., Dano, S., and Sorensen, P.G. (2001) Full-scale model of glycolysis in *Saccharomyces cerevisiae*. *Biophys. Chem.*, 94 (1–2), 121–163.
- 5 Rizzi, M., Baltes, M., Theobald, U., and Reuss, M. (1997) *In vivo* analysis of metabolic dynamics in *Saccharomyces cerevisiae*: II. Mathematical model. *Biotechnol. Bioeng.*, 55 (4), 592–608.
- 6 Theobald, U., Mailinger, W., Baltes, M., Rizzi, M., and Reuss, M. (1997) *In vivo* analysis of metabolic dynamics in *Saccharomyces cerevisiae*: I. Experimental observations. *Biotechnol. Bioeng.*, 55 (2), 305–316.
- 7 Tiger, C.F., Krause, F., Cedersund, G., Palmer, R., Klipp, E., Hohmann, S. et al. (2012) A framework for mapping, visualisation and automatic model creation of signal-transduction networks. *Mol. Syst. Biol.*, 8, 578.
- 8 Yi, T.M., Kitano, H., and Simon, M.I. (2003) A quantitative characterization of the yeast heterotrimeric G protein cycle. *Proc. Natl. Acad. Sci. USA*, 100 (19), 10764–10769.
- 9 Dohlman, H.G. (2002) G proteins and pheromone signaling. *Annu. Rev. Physiol.*, 64, 129–152.
- 10 Neer, E.J. (1995) Heterotrimeric G proteins: organizers of transmembrane signals. *Cell*, 80 (2), 249–257.
- 11 Blumer, K.J. and Thorner, J. (1991) Receptor-G protein signaling in yeast. *Annu. Rev. Physiol.*, 53, 37–57.
- 12 Buck, L.B. (2000) The molecular architecture of odor and pheromone sensing in mammals. *Cell*, 100 (6), 611–618.
- 13 Dohlman, H.G., Thorner, J., Caron, M.G., and Lefkowitz, R.J. (1991) Model systems for the study of seven-transmembrane-segment receptors. *Annu. Rev. Biochem.*, 60, 653–688.
- 14 Banuett, F. (1998) Signalling in the yeasts: an informational cascade with links to the filamentous fungi. *Microbiol. Mol. Biol. Rev.*, 62 (2), 249–274.
- 15 Wang, P. and Heitman, J. (1999) Signal transduction cascades regulating mating, filamentation, and virulence in *Cryptococcus neoformans*. *Curr. Opin. Microbiol.*, 2 (4), 358–362.
- 16 Dohlman, H.G., Song, J., Apanovitch, D.M., DiBello, P.R., and Gillen, K.M. (1998) Regulation of G protein signalling in yeast. *Semin. Cell Dev. Biol.*, 9 (2), 135–141.
- 17 Offermanns, S. (2000) Mammalian G-protein function *in vivo*: new insights through altered gene expression. *Rev. Physiol. Biochem. Pharmacol.*, 140, 63–133.
- 18 Dohlman, H.G. and Thorner, J.W. (2001) Regulation of G protein-initiated signal transduction in yeast: paradigms and principles. *Annu. Rev. Biochem.*, 70, 703–754.
- 19 Meigs, T.E., Fields, T.A., McKee, D.D., and Casey, P.J. (2001) Interaction of Galpha 12 and Galpha 13 with the cytoplasmic domain of cadherin provides a mechanism for beta-catenin release. *Proc. Natl. Acad. Sci. USA*, 98 (2), 519–524.
- 20 Dohlman, H.G. and Thorner, J. (1997) RGS proteins and signaling by heterotrimeric G proteins. *J. Biol. Chem.*, 272 (7), 3871–3874.
- 21 Ross, E.M. and Wilkie, T.M. (2000) GTPase-activating proteins for heterotrimeric G proteins: regulators of G protein signaling (RGS) and RGS-like proteins. *Annu. Rev. Biochem.*, 69, 795–827.
- 22 Siderovski, D.P., Strockbine, B., and Behe, C.I. (1999) Whither goest the RGS proteins? *Crit. Rev. Biochem. Mol. Biol.*, 34 (4), 215–251.
- 23 Burchett, S.A. (2000) Regulators of G protein signaling: a bestiary of modular protein binding domains. *J. Neurochem.*, 75 (4), 1335–1351.
- 24 Takai, Y., Sasaki, T., and Matozaki, T. (2001) Small GTP-binding proteins. *Physiol. Rev.*, 81 (1), 153–208.

- 25** Francke, C., Postma, P.W., Westerhoff, H.V., Blom, J.G., and Peletier, M.A. (2003) Why the phosphotransferase system of *Escherichia coli* escapes diffusion limitation. *Biophys. J.*, 85 (1), 612–622.
- 26** Rohwer, J.M., Meadow, N.D., Roseman, S., Westerhoff, H.V., and Postma, P.W. (2000) Understanding glucose transport by the bacterial phosphoenolpyruvate:glycose phosphotransferase system on the basis of kinetic measurements *in vitro*. *J. Biol. Chem.*, 275 (45), 34909–34921.
- 27** Postma, P.W., Broekhuizen, C.P., and Geerse, R.H. (1989) The role of the PEP: carbohydrate phosphotransferase system in the regulation of bacterial metabolism. *FEMS Microbiol. Rev.*, 5 (1–2), 69–80.
- 28** Postma, P.W., Lengeler, J.W., and Jacobson, G.R. (1993) Phosphoenolpyruvate:carbohydrate phosphotransferase systems of bacteria. *Microbiol. Rev.*, 57 (3), 543–594.
- 29** Klipp, E., Nordlander, B., Kruger, R., Gennemark, P., and Hohmann, S. (2005) Integrative model of the response of yeast to osmotic shock. *Nat. Biotechnol.*, 23 (8), 975–982.
- 30** Hohmann, S. (2002) Osmotic stress signaling and osmoadaptation in yeasts. *Microbiol. Mol. Biol. Rev.*, 66 (2), 300–372.
- 31** Wilkinson, M.G. and Millar, J.B. (2000) Control of the eukaryotic cell cycle by MAP kinase signaling pathways. *Faseb J.*, 14 (14), 2147–2157.
- 32** Huang, C.Y. and Ferrell, J.E. Jr. (1996) Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proc. Natl. Acad. Sci. USA*, 93 (19), 10078–10083.
- 33** Kholodenko, B.N. (2000) Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades. *Eur. J. Biochem.*, 267 (6), 1583–1588.
- 34** Force, T., Bonventre, J.V., Heidecker, G., Rapp, U., Avruch, J., and Kyriakis, J.M. (1994) Enzymatic characteristics of the c-Raf-1 protein kinase. *Proc. Natl. Acad. Sci. USA*, 91 (4), 1270–1274.
- 35** Lee, E., Salic, A., Kruger, R., Heinrich, R., and Kirschner, M.W. (2003) The roles of APC and Axin derived from experimental and theoretical analysis of the Wnt pathway. *PLoS Biology*, 1 (1), E10.
- 36** Goentoro, L. and Kirschner, M.W. (2009) Evidence that fold-change, and not absolute level, of beta-catenin dictates Wnt signaling. *Mol. Cell*, 36 (5), 872–884.
- 37** Goentoro, L., Shoval, O., Kirschner, M.W., and Alon, U. (2009) The incoherent feedforward loop can provide fold-change detection in gene regulation. *Mol. Cell*, 36 (5), 894–899.
- 38** Benary, U., Kofahl, B., Hecht, A., and Wolf, J. (2013) Modeling Wnt/beta-catenin target gene expression in APC and wnt gradients under wild type and mutant conditions. *Front. Physiol.*, 4, 21.
- 39** Thomas, R. and Kaufman, M. (2002) Conceptual tools for the integration of data. *C. R. Biol.*, 325 (4), 505–514.
- 40** Tyson, J.J. (1975) Classification of instabilities in chemical-reaction systems. *J. Chem. Phys.*, 62 (3), 1010–1015.
- 41** Schaber, J., Baltanas, R., Bush, A., Klipp, E., and Colman-Lerner, A. (2012) Modelling reveals novel roles of two parallel signalling pathways and homeostatic feedbacks in yeast. *Mol. Syst. Biol.*, 8, 622.
- 42** Heinrich, R., Neel, B.G., and Rapoport, T.A. (2002) Mathematical models of protein kinase signal transduction. *Mol. Cell*, 9 (5), 957–970.
- 43** Komarova, N.L., Zou, X., Nie, Q., and Bardwell, L. (2005) A theoretical framework for specificity in cell signaling. *Mol. Syst. Biol.*, 1, 0023.
- 44** Schaber, J., Kofahl, B., Kowald, A., and Klipp, E. (2006) A modeling approach to quantify dynamic crosstalk between the pheromone and the starvation pathway in baker's yeast. *FEBS J.*, 273 (15), 3520–3533.
- 45** Kaizu, K., Ghosh, S., Matsuoka, Y., Moriya, H., Shimizu-Yoshida, Y., and Kitano, H. (2010) A comprehensive molecular interaction map of the budding yeast cell cycle. *Mol. Syst. Biol.*, 6, 415.
- 46** Hartwell, L.H. (1974) *Saccharomyces cerevisiae* cell cycle. *Bacteriol. Rev.*, 38 (2), 164–198.
- 47** Hartwell, L.H., Culotti, J., Pringle, J.R., and Reid, B.J. (1974) Genetic control of the cell division cycle in yeast. *Science*, 183 (120), 46–51.
- 48** Nurse, P. (1975) Genetic control of cell size at cell division in yeast. *Nature*, 256 (5518), 547–551.
- 49** Pardee, A.B. (1974) A restriction point for control of normal animal cell proliferation. *Proc. Natl. Acad. Sci. USA*, 71 (4), 1286–1290.
- 50** Nurse, P. and Bissett, Y. (1981) Gene required in G1 for commitment to cell cycle and in G2 for control of mitosis in fission yeast. *Nature*, 292 (5823), 558–560.
- 51** Goldbeter, A. (1991) A minimal cascade model for the mitotic oscillator involving cyclin and cdc2 kinase. *Proc. Natl. Acad. Sci. USA*, 88 (20), 9107–9111.
- 52** Goldbeter, A. and Koshland, D.E. Jr. (1984) Ultrasensitivity in biochemical systems controlled by covalent modification. Interplay between zero-order and multistep effects. *J. Biol. Chem.*, 259 (23), 14441–14447.
- 53** Chen, K.C., Calzone, L., Csikasz-Nagy, A., Cross, F.R., Novak, B., and Tyson, J.J. (2004) Integrative analysis of cell cycle control in budding yeast. *Mol. Biol. Cell.*, 15 (8), 3841–3862.
- 54** Chen, K.C., Csikasz-Nagy, A., Gyorffy, B., Val, J., Novak, B., and Tyson, J.J. (2000) Kinetic analysis of a molecular model of the budding yeast cell cycle. *Mol. Biol. Cell.*, 11 (1), 369–391.
- 55** Novák, B., Tóth, A., Csikász-Nagy, A., Györffy, B., Tyson, J.J., and Nasmyth, K. (1999) Finishing the cell cycle. *J. Theor. Biol.*, 199 (2), 223–233.
- 56** Tyson, J.J., Novak, B., Odell, G.M., Chen, K., and Thron, C.D. (1996) Chemical kinetic theory: understanding cell-cycle regulation. *Trends Biochem. Sci.*, 21 (3), 89–96.
- 57** Spiesser, T.W., Muller, C., Schreiber, G., Krantz, M., and Klipp, E. (2012) Size homeostasis can be intrinsic to growing cell populations and explained without size sensing or signalling. *FEBS J.*, 279 (22), 4213–4230.
- 58** Spiesser, T.W., Uschner, F., Adler, S., Münzner, U., Krantz, M., and Klipp, E. (2015) A yeast cell cycle model integrating stress signaling and physiology, in *Systems Biology of the Cell Cycle: Towards Integration with Cell Physiology* (ed. M. Barberis), Springer.
- 59** Adrover, M.A., Zi, Z., Duch, A., Schaber, J., Gonzalez-Novo, A., Jimenez, J. et al. (2011) Time-dependent quantitative multi-component control of the G(1)-S network by the stress-activated protein kinase Hog1 upon osmostress. *Sci. Signal.*, 4 (192), ra63.
- 60** Fall, C.P., Marland, E.S., Wagner, J.M., and Tyson, J.J. (2002) *Computational Cell Biology*, Springer-Verlag New York, Inc.
- 61** Oeppen, J. and Vaupel, J.W. (2002) Broken limits to life expectancy. *Supramol. Sci.*, 296, 1029–1031.
- 62** Gompertz, B. (1825) On the nature of the function expressive of the law of human mortality and on a new mode of determining life contingencies. *Philos. Trans. R. Soc.*, 2, 513–585.
- 63** Makeham, W.H. (1867) On the law of mortality. *J. Inst. Actuaries*, 13, 325–358.
- 64** Kowald, A. (2002) Lifespan does not measure ageing. *Biogerontology*, 3 (3), 187–190.
- 65** Medvedev, Z.A. (1990) An attempt at a rational classification of theories of ageing. *Biol. Rev.*, 65, 375–398.
- 66** Lundblad, V. and Szostak, J.W. (1989) A mutant with a defect in telomere elongation leads to senescence in yeast. *Cell*, 57, 633–643.

- 67** Yu, G.-L., Bradley, J.D., Attardi, L.D., and Blackburn, E.H. (1990) *In vivo* alteration of telomere sequences and senescence caused by mutated *Tetrahymena telomerase* RNAs. *Nature*, 344, 126–131.
- 68** Olovnikov, A.M. (1973) A theory of Marginotomy. *J. Theor. Biol.*, 41, 181–190.
- 69** Watson, J.D. (1972) Origin of concatameric T4 DNA. *Nature*, 239, 197–201.
- 70** Harley, C.B. (1991) Telomere loss: mitotic clock or genetic time bomb? *Mut. Res.*, 256, 271–282.
- 71** Harley, C.B., Vaziri, H., Counter, C.M., and Allsopp, R.C. (1992) The telomere hypothesis of cellular aging. *Exp. Gerontol.*, 27, 375–382.
- 72** Hayflick, L. (1965) The limited *in vitro* lifetime of human diploid cell strains. *Exp. Cell Res.*, 37, 614–636.
- 73** von Zglinicki, T., Saretzki, G., Docke, W., and Lotze, C. (1995) Mild hyperoxia shortens telomeres and inhibits proliferation of fibroblasts. A model for senescence. *Exp. Cell Res.*, 220 (1), 186–193.
- 74** Hirsch, H.R., Coomes, J.A., and Witten, M. (1989) The waste-product theory of aging: transformation to unlimited growth in cell cultures. *Exp. Gerontol.*, 24 (2), 97–112.
- 75** Miquel, J., Economos, A.C., Fleming, J., and Johnson, J.E. (1980) Mitochondrial role in cell ageing. *Exp. Gerontol.*, 15, 575–591.
- 76** Linnane, A.W., Baumer, A., Maxwell, R.J., Preston, H., Zhang, C., and Marzuki, S. (1990) Mitochondrial gene mutation: the ageing process and degenerative diseases. *Biochem. Int.*, 22 (6), 1067–1076.
- 77** Linnane, A.W., Marzuki, S., Ozawa, T., and Tanaka, M. (1989) Mitochondrial DNA mutations as an important contributor to ageing and degenerative diseases. *Lancet*, 333, 642–645.
- 78** de Grey, A.D.N.J. (1997) A proposed refinement of the mitochondrial free radical theory of aging. *BioEssays*, 19 (2), 161–166.
- 79** Kowald, A., Jendrach, M., Pohl, S., Bereiter-Hahn, J., and Hammerstein, P. (2005) On the relevance of mitochondrial fusions for the accumulation of mitochondrial deletion mutants: a modelling study. *Aging Cell*, 4 (5), 273–283.
- 80** Kowald, A. and Kirkwood, T.B. (2014) Transcription could be the key to the selection advantage of mitochondrial deletion mutants in aging. *Proc. Natl. Acad. Sci. USA*, 111 (8), 2972–2977.
- 81** Weismann, A. (1891) *Essays Upon Heredity and Kindred Biological Problems*, 2nd edn (ed.): Clarendon Press, Oxford.
- 82** Maynard Smith, J. (1976) Group selection. *Q. Rev. Biol.*, 51, 277–283.
- 83** Medawar, P.B. (1952) *An Unsolved Problem of Biology: An Inaugural Lecture Delivered at University College*, H.K. Lewis, London.
- 84** Stearns, S.C. and Hoekstra, R.F. (2000) *Evolution, An Introduction*, Oxford University Press.
- 85** Williams, G.C. (1957) Pleiotropy, natural selection and the evolution of senescence. *Evolution*, 11, 398–411.
- 86** Kirkwood, T.B.L. and Holliday, R. (1986) Ageing as a consequence of natural selection, in *The Biology of Human Ageing* (eds A.H. Bittles and K.J. Collins), Cambridge University Press, p. 1–15.
- 87** Kirkwood, T.B.L. and Rose, M.R. (1991) *Evolution of senescence: Late survival sacrificed for reproduction*, Philosophical Transactions of the Royal Society, London B, 332, pp. 15–24.
- 88** Maynard Smith, J. (1989) *Evolutionary Genetics*, Oxford University Press, Oxford.
- 89** Stearns, S.C. (1992) *The Evolution of Life Histories*, Oxford University Press, Oxford.
- 90** Schwartz, R.M. and Dayhoff, M.O. (1978) Origins of prokaryotes, eukaryotes, mitochondria, and chloroplasts. *Science*, 199, 395–403.
- 91** Anderson, S., Bankier, A.T., Barrell, B.G., Debruijn, M.H.L., Coulson, A.R., Drouin, J. et al. (1981) Sequence and organization of the human mitochondrial genome. *Nature*, 290, 457–465.
- 92** Cao, Z., Wanagat, J., McKiernan, S.H., and Aiken, J.M. (2001) Mitochondrial DNA deletion mutations are concomitant with ragged red regions of individual, aged muscle fibers: analysis by laser-capture microdissection. *Nucleic acids Res.*, 29 (21), 4502–4508.
- 93** Brierley, E.J., Johnson, M.A., Lightowers, R.N., James, O.F., and Turnbull, D.M. (1998) Role of mitochondrial DNA mutations in human aging: implications for the central nervous system and muscle. *Ann. Neurol.*, 43 (2), 217–223.
- 94** Krupko, K., Boddyak, N., Thilly, W.G., van Orsouw, N.J., Zhang, X., Coller, H.A. et al. (1999) Cell by cell scanning of whole mitochondrial genomes in aged human heart reveals a significant fraction of myocytes with clonally expanded deletions. *Nucleic Acids Res.*, 27 (11), 2434–2441.
- 95** Gokey, N.G., Cao, Z., Pak, J.W., and Lee, D., McKiernan, S.H., McKenzie, D. et al. (2004) Molecular analyses of mtDNA deletion mutations in microdissected skeletal muscle fibers from aged rhesus monkeys. *Aging Cell*, 3 (5), 319–326.
- 96** Herbst, A., Pak, J.W., McKenzie, D., Bua, E., Bassiouni, M., and Aiken, J.M. (2007) Accumulation of mitochondrial DNA deletion mutations in aged muscle fibers: evidence for a causal role in muscle fiber loss. *J. Gerontol. A Biol. Sci. Med. Sci.*, 62 (3), 235–245.
- 97** McKiernan, S.H., Colman, R., Lopez, M., Beasley, T.M., Weindruch, R., and Aiken, J.M. (2009) Longitudinal analysis of early stage sarcopenia in aging rhesus monkeys. *Exp. Gerontol.*, 44 (3), 170–176.
- 98** Elson, J.L., Samuels, D.C., Turnbull, D.M., and Chinnery, P.F. (2001) Random intracellular drift explains the clonal expansion of mitochondrial DNA mutations with age. *Am. J. Hum. Genet.*, 68 (3), 802–806.
- 99** Chinnery, P.F. and Samuels, D.C. (1999) Relaxed replication of mtDNA: a model with implications for the expression of disease. *Am. J. Hum. Genet.*, 64 (4), 1158–1165.
- 100** Kowald, A. and Kirkwood, T.B. (2013) Mitochondrial mutations and aging: random drift is insufficient to explain the accumulation of mitochondrial deletion mutants in short-lived animals. *Aging Cell*, 12 (4), 728–731.
- 101** Müller-Höcker, J. (1990) Cytochrome c oxidase deficient fibres in the limb muscle and diaphragm of man without muscular disease: an age-related alteration. *J. Neurol. Sci.*, 100 (1–2), 14–21.
- 102** Wallace, D.C. (1992) Mitochondrial genetics: a paradigm for aging and degenerative diseases? *Science*, 256 (5057), 628–632.
- 103** Lee, C.M., Lopez, M.E., Weindruch, R., and Aiken, J.M. (1998) Association of age-related mitochondrial abnormalities with skeletal muscle fiber atrophy. *Free Radic. Biol. Med.*, 25 (8), 964–972.
- 104** Clayton, D.A. (1982) Replication of animal mitochondrial DNA. *Cell*, 28 (4), 693–705.
- 105** Huemer, R.P., Lee, K.D., Reeves, A.E., and Bickert, C. (1971) Mitochondrial studies in senescent mice - II. Specific activity, buoyant density, and turnover of mitochondrial DNA. *Exp. Gerontol.*, 6, 327–334.
- 106** Korr, H., Kurz, C., Seidler, T.O., Sommer, D., and Schmitz, C. (1998) Mitochondrial DNA synthesis studied autoradiographically in various cell types *in vivo*. *Braz. J. Med. Biol. Res.*, 31 (2), 289–298.
- 107** Kowald, A., Dawson, M., and Kirkwood, T.B. (2014) Mitochondrial mutations and aging: can mitochondrial deletion mutants accumulate via a size based replication advantage? *J. Theor. Biol.*, 340, 111–118.

Part Two

Reference Section

Summary

A basic characteristic of life is that organisms are composed of cells that can grow, differentiate, and reproduce. Molecular interactions and molecular structures define structural and functional properties of a cell. Electrostatic interactions and different classes of biological molecules, such as carbohydrates, lipids, proteins, and nucleic acids, are the major components that govern the characteristics of biological systems such as their structural organization and physiological processes. In contrast to simple prokaryotic cells, eukaryotic cells are compartmentalized and have organelles that fulfill specific cellular functions, for example, the nucleus holds the cell's genome organized in chromosomes, the cytosol is the compartment where protein biosynthesis and major metabolic processes, such as glycolysis, take place, mitochondria act as cellular power plants, and the endoplasmic reticulum and the Golgi complex are central components of protein sorting and their posttranslational modification. The expression of the genetic information in eukaryotes is a complex process that comprises gene regulation, transcription, processing, translation, posttranslational modification, and protein sorting.

This section gives a brief overview of biology and related subjects, such as biochemistry, with a focus on molecular biology, since the latter is most relevant to current systems biology. We will review several basics on biochemistry, and introduce fundamental knowledge of biology. The basics are required for the setup of all models for biological systems, and the meaningful interpretation of simulation results and analysis. For a broader and more detailed introduction to biology, it is recommended to consult books by Alberts *et al.* [1] and Reece *et al.* [2].

Biology is the science that deals with living organisms and their interrelationships between each other and their environment in light of their evolutionary origin. Some of the main characteristics of organisms are the following:

13.1 The Origin of Life

13.2 Molecular Biology of the Cell

- Chemical Bonds and Forces Important in Biological Molecules
- Functional Groups in Biological Molecules
- Major Classes of Biological Molecules

13.3 Structural Cell Biology

- Structure and Function of Biological Membranes
- Nucleus
- Cytosol
- Mitochondria
- Endoplasmic Reticulum and Golgi Complex
- Other Organelles

13.4 Expression of Genes

- Transcription
- Processing of the mRNA
- Translation
- Protein Sorting and Posttranslational Modifications
- Regulation of Gene Expression

Exercises

References

Further Reading

- *Physiology*: All living organisms assimilate nutrients, extract energy from these nutrients, produce substances themselves, and excrete the remains.
- *Growth and reproduction*: All living organisms grow and reproduce their own species.
- *Cellular composition*: Cells are the general building blocks of organisms.

Biology is divided into several disciplines, such as physiology, morphology, cytology, ecology, developmental biology, behavioral and evolutionary biology, molecular biology, biochemistry, and classical and molecular genetics. Biology tries to explain characteristics such as the shape and structure of organisms and their change during

time, as well as phenomena of their regulatory, individual, or environmental relationships. This section gives a brief overview about this scientific field with a focus on biological molecules, fundamental cellular structures, and molecular biology and genetics.

13.1 The Origin of Life

The earliest development on earth began 4½ billion years ago. Massive volcanism released water (H_2O), methane (CH_4), ammonia (NH_3), hydrogen sulfide (H_2S), and molecular hydrogen (H_2), which formed a reducing atmosphere and the early ocean. By loss of hydrogen into space and gas reactions, an atmosphere consisting of nitrogen (N_2), carbon monoxide (CO), carbon dioxide (CO_2), and water (H_2O) was formed. The impact of huge amounts of energy (e.g., sunlight with a high portion of ultraviolet (UV) radiation and electric discharges) onto the reducing atmosphere along with the catalytic effect of solid-state surfaces resulted in an enrichment of simple organic molecules such as amino acids, purines, pyrimidines, and monosaccharides in the early ocean. This is called the prebiotic broth hypothesis and is based on the experiments of Miller and Urey [3]. Another possibility is that the first forms of life formed in the deep sea utilizing the energy of hydrothermal vents, well protected from damaging UV radiation and the unstable environment of the surface [4]. Once simple organic molecules were formed in significant amounts, they presumably assembled spontaneously into macromolecules such as proteins and nucleic acids. By formation of molecular aggregates from these colloidally solved macromolecules, the development of simple compartmented reaction pathways for the utilization of energy sources was possible. Besides this, enzymes appeared that permitted specific reactions to take place in ordered sequences at moderate temperatures, and information systems necessary for directed synthesis and reproduction were developed. The appearance of the first primitive cells – the last common ancestors of all past and recent organisms – was the end of the abiotic (chemical) and the beginning of the biotic (biological) evolution. Later, these first primitive cells evolved into the first prokaryotic cells (prokaryotes). About 3½ billion years ago, the reducing atmosphere was very slowly enriched by oxygen (O_2) due to the rise of photosynthesis that resulted in an oxidative atmosphere (1.4 billion years ago: 0.2% O_2 ; 0.4 billion years ago: 2% O_2 ; today: about 21% O_2).

Prokaryotes (eubacteria and archaeabacteria) are mostly characterized by their size and simplistic structure compared with the more evolved eukaryotes. Table 13.1 summarizes several differences between these groups. The evolutionary origin of the eukaryotic cells is explained by

Table 13.1 Some important differences between prokaryotic and eukaryotic cells.

	<i>Prokaryotes</i>	<i>Eukaryotes</i>
Size	Mostly about 1–10 μm in length	Mostly about 10–100 μm in length
Nucleus	Nucleus is missing; chromosomal region is called nucleolus	Nucleus is separated from the cytoplasm by the nuclear envelope
Intracellular organization	Normally, no membrane-separated compartments and no supportive intracellular skeletal framework are present in the cells' interior	Distinct compartments are present, for example, nucleus, cytosol with a cytoskeleton, mitochondria, endoplasmic reticulum, Golgi complex, lysosomes, plastids (chloroplasts, leucoplasts)
Gene structure	No introns; some polycistronic genes	Introns and exons
Cell division	Simple cell division	Mitosis or meiosis
Ribosome	Consists of a large 50 S subunit and a small 30 S subunit	Consists of a large 60 S subunit and a small 40 S subunit
Reproduction	Parasexual recombination	Sexual recombination
Organization	Mostly single cellular	Mostly multicellular, and with cell differentiation

the formation of a nucleus and several compartments, and by the inclusion of prokaryotic cells, which is described by the endosymbiont hypothesis. This hypothesis states that cellular organelles, such as mitochondria and chloroplasts, are descendants of specialized cells (e.g., specialization for energy utilization) that have been engulfed by the early eukaryotes.

Prokaryotes and these early eukaryotes are single-celled organisms. Later during evolution, single-celled eukaryotes evolved further into multicellular organisms. Their cells are mostly genetically identical, but differentiate into several specialized cell types during development. Most of these organisms reproduce sexually.

The developmental process that takes place by sexual reproduction starts with a fertilized egg (zygote) that divides several times (the cell division underlying this process is discussed in more detail in Section 13.3). For instance, in the frog *Xenopus laevis* – which is a vertebrate and belongs to the amphibians – the development starts with the zygote and passes through several developmental phases, that is, morula (64 cells), blastula (10 000 cells), gastrula (30 000 cells), and neurula (80 000 cells), before forming the tadpole (with a million cells 110 h after fertilization) that develops into the adult frog later on. This process is

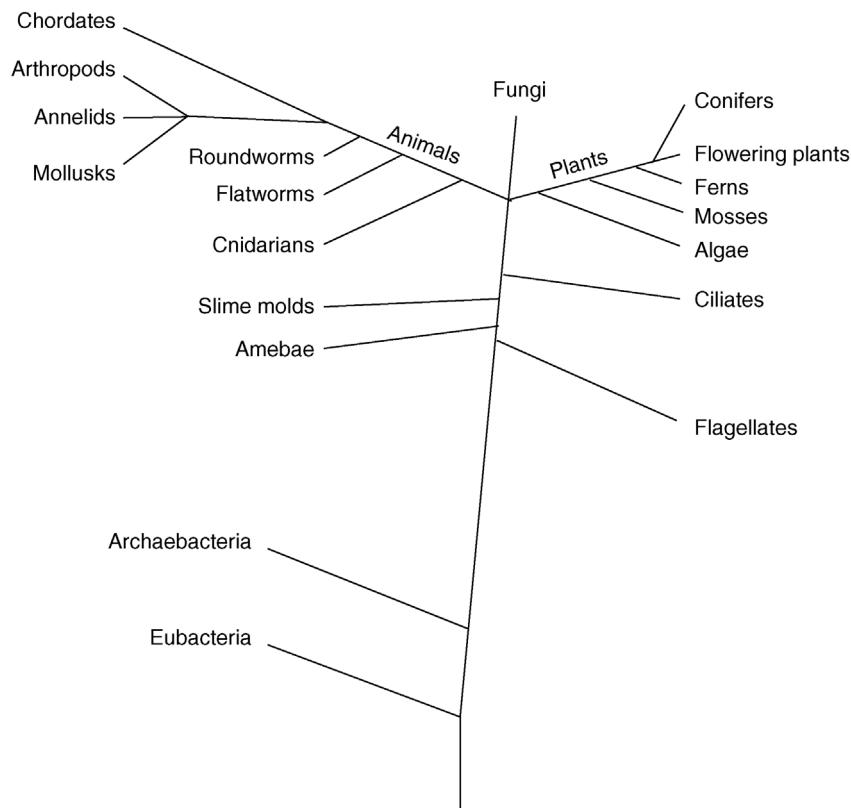


Figure 13.1 The tree of life shows phylogenetic relations between some major groups of organism.

genetically determined and several phases are similar between species that are related close to each other due to their identical evolutionary origin. Figure 13.1 shows a simplified tree of life that illustrates major evolutionary relations.

While most places with moderate aerobic conditions were populated by eukaryotes, the prokaryotic archaeabacteria in particular have specialized to survive under extreme conditions (e.g., thermophilic bacteria, which propagate at temperatures of 85–105 °C in the black smokers of the deep sea, or the halobacteria, which live in high salt concentrations).

Along with organisms that have their own metabolism, parasitic viruses and viroids that utilize cells for reproduction have developed. Viruses consist of a very small genome surrounded by a protein envelope (capsid); viroids are single-stranded circular RNAs. Due to the absence of metabolism and a cellular structure, these parasites are not regarded as living organisms.

The phenotypical diversity of organisms observed is also displayed in the structure of their hereditary information: the size of this genomic information can vary, as can its organization into different elements, that is, plasmids and chromosomes. Table 13.2 summarizes some data acquired from commonly investigated organisms.

Table 13.2 Genome sizes of different organisms from the prokaryotic and eukaryotic kingdom.

Organism	Number of chromosomes (haploid genome)	Genome size (base pairs; genes)
<i>Mycoplasma genitalium</i> (prokaryote)	1 circular chromosome	5.8×10^5 bp; 480 genes
<i>Escherichia coli</i> (prokaryote)	1 circular chromosome	4.6×10^6 bp; 4290 genes
<i>Saccharomyces cerevisiae</i> (budding yeast; eukaryote)	16 chromosomes	12.5×10^6 bp; 6186 genes
<i>Arabidopsis thaliana</i> (flowering plant; eukaryote)	5 chromosomes	100×10^6 bp; ~25 000 genes
<i>Drosophila melanogaster</i> (fruit fly, eukaryote)	4 chromosomes	180×10^6 bp; ~14 000 genes
<i>Mus musculus</i> (mouse, eukaryote)	20 chromosomes	2.5×10^9 bp; ~30 000 genes
<i>Homo sapiens</i> (human, eukaryote)	23 chromosomes	2.9×10^9 bp; ~30 000 genes

Information about further organisms can be found, for example, at <http://www.cbs.dtu.dk/services/GenomeAtlas> and www.ensembl.org.

13.2 Molecular Biology of the Cell

Cellular structures and processes result from a complex interaction network of biological molecules. The properties of these molecules determine possible interactions. Although many of these molecules are highly complex, most fall into one of the following four classes or contain substructures that belong to one of these: carbohydrates, lipids, proteins, and nucleic acids. Along with these four classes, water is essential for all living systems. Molecules are held together by and interact through chemical bonds and forces of different types: ionic, covalent, and hydrogen bonds, nonpolar associations, and van der Waals forces. The following sections will provide a foundation for the understanding of molecular structures, functions, and interactions by giving a brief introduction to chemical bonds and forces, to the most important classes of biological molecules, and to complex macromolecular structures formed by these molecules.

13.2.1 Chemical Bonds and Forces Important in Biological Molecules

The atomic model introduced by Rutherford and significantly extended by Bohr describes the atom as a positively charged nucleus being surrounded by one or more shells (or, more exactly, energy levels) that are filled with electrons. Most significant for the chemical properties of an atom is the number of electrons in its outermost shell. Atoms tend to fill up their outermost shell to obtain a stable state. The innermost or first shell is filled up by two electrons. The second and further shells are filled up by $2n^2$ electrons, where n depicts the number of the shell. However, due to reasons of energetic stability, the outermost shell will not contain more than eight electrons. For example, helium, with two electrons in its single shell, or atoms such as neon or argon, with eight electrons in their outermost shells, are essentially chemically inert. Atoms with a number of electrons near to these numbers tend to lose or gain electrons to attain these stable states. For example, sodium (one electron in its outer shell) and chlorine (seven electrons in its outer shell) can both achieve such a stable state by transferring one electron from sodium to chlorine, thus forming the ions Na^+ and Cl^- . The force holding together the oppositely charged ions in solid state is called the ionic or electrostatic bond (Figure 13.2a). If the number of electrons in the outer shell differs by more than one, atoms tend to share electrons by forming a so-called covalent bond (Figure 13.2b). Atoms held together by covalent bonds are called molecules. If the shared electron pair is equally

distributed between the two involved atoms, this bond is called nonpolar (e.g., for the hydrogen molecule). If one atom has a higher attraction to the shared electron pair, it becomes partially negatively charged. Then the other atom in this polar association becomes partially positively charged, as is the case with the water molecule (H_2O), where the oxygen attracts the shared electron pairs stronger than the hydrogen atoms do. Thus, $-\text{OH}$ and $-\text{NH}$ groups usually form polar regions in which the hydrogen is partially positively charged. A measurement for the affinity of an atom to attract electrons in a covalent bond is given by its electronegativity, which was introduced by Linus Pauling. In addition to single covalent bonds, double and triple bonds also exist. These kinds of bonds are more exactly described by the quantum-mechanical atomic model, in which the electron shells of an atom can be described by one of several differently shaped orbitals that represent the areas where the electrons are located with highest probability (electron clouds). A covalent bond is then described by molecular orbitals, which are derived from atomic orbitals. Furthermore, if single and double bonds are altered in a single molecule or a double bond is in direct vicinity of an atom with a free electron pair, then one electron pair of the double bond and the free electron pair can delocalize across the participating atoms, for example, the three electron pairs in benzol (Figure 13.2b) or the double bond between C and O and the free electron pair of N in a peptide bond (Figure 13.6a). Such electrons are called delocalized π -electrons. For a more detailed description, please consult books about general and inorganic chemistry or introductory books about biochemistry.

Hydrogen atoms with a positive partial charge that are bound to oxygen or nitrogen (as in H_2O or NH_3) are able to interact with free electron pairs of atoms with a negative partial charge. These attractions are called hydrogen bonds and are relatively weak compared with solid-state ionic bonds or covalent bonds. To break a hydrogen bond, only about 4 kJ mol^{-1} is required. Therefore, hydrogen bonds separate readily at elevated temperatures, which is often the reason why proteins such as enzymes lose their function during heating. Likewise, the hydrogen bonds that hold together the double strands of nucleic acids (see Section 13.2.3) can be separated at high temperatures. This fact is utilized for several molecular biological methods, for example, polymerase chain reaction (PCR), or for the radioactive labeling of DNA (deoxyribonucleic acid) fragments (see Chapter 14 for more details). Hydrogen bonds also explain why water is liquid at room temperature and boils at 100°C . Small alcohols, such as methanol or ethanol, are fully soluble in water, due to their hydroxyl group that interacts

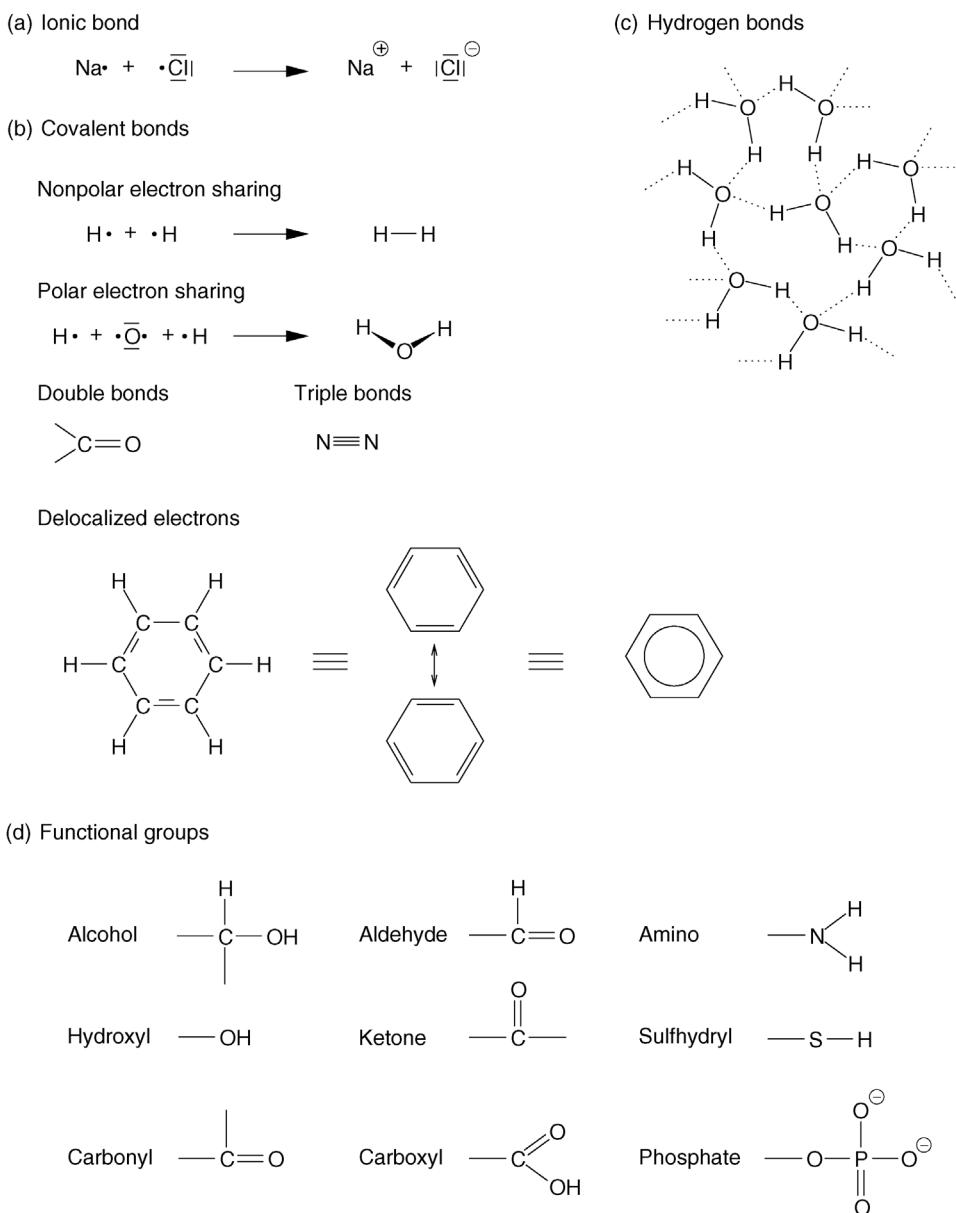


Figure 13.2 Chemical bonds and functional organic groups. Single electrons in the outer shell are visualized by a dot; electron pairs are replaced by a dash. Shared electron pairs are represented by a dash between two atoms. (a) Single charged Na^{+} and Cl^{-} ions are formed by the transition of the single outermost electron of sodium to chlorine. (b) In a covalent bond, electrons are shared between two atoms. If the shared electron pair is attracted more strongly by one of the participating atoms than by the other, this bond is called a polar bond. Depending on the molecular structure, double and triple bonds can occur as well. Sometimes, binding electron pairs might also be delocalized among several atoms, as is the case in benzene. (c) Unequal electron sharing causes the formation of hydrogen bonds (shown by dotted lines) as found in water. (d) The skeleton of organic molecules essentially consists of carbon atoms bound to each other or to hydrogen. Some of these carbons are bound to or are part of functional groups with special chemical characteristics. Hence, these influence the reactivities and physicochemical properties of the molecule.

with the hydrogen bonds of water, whereas larger alcohols, such as hexanol or heptanol, are weakly soluble or insoluble in water due to their longer unpolar carbohydrate tail. As we have seen, polarized functional groups can interact with water, which is why they often

are called hydrophilic (or lipophobic), while nonpolar molecules or molecule parts are called hydrophobic (or lipophilic).

Also critical to structures and interactions of biological molecules are the van der Waals forces. The electron

clouds surrounding atoms that are held together by covalent bonds are responsible for these forces. Momentary inequalities in the distribution of electrons in any covalent bond, due to chance, can make one end of the covalent bond more negative or positive than the other for a short moment, which results in rapid fluctuations in the charge of the electron cloud. These fluctuations can induce opposite fluctuations in nearby covalent bonds, thus establishing a weak attractive force. The closer the electron clouds, the stronger the attractive force, but if the outermost electron orbitals begin to overlap, the negatively charged electrons strongly repel each other. Thus, van der Waals forces can be either attractive or repulsive. Their binding affinity is, at 0.4 kJ mol⁻¹ in water, even lower than that of hydrogen bonds. The optimal distance for maximum van der Waals forces of an atom is called its van der Waals contact radius. The van der Waals repulsions have an important influence on the possible conformations of a molecule.

13.2.2 Functional Groups in Biological Molecules

As outlined before, one major characteristic of life are physiological processes in which nutrients from the outside are converted by the organism to maintain a thermodynamically open system with features such as development or behavior. These physiological processes are realized on the metabolic level by myriads of reactions in which specific molecules are converted into others. These intra- or intermolecular rearrangements often take place at specific covalent bonds that can more readily be disturbed than others. Such covalent bonds are often formed by certain intramolecular substructures that are called functional groups. Thus, functional groups often serve as reaction centers converting some molecules into others or link some molecular subunits to form larger molecular assemblies, for example, polypeptides or nucleic acids. The functional groups most relevant in biological molecules are hydroxyl, carbonyl, carboxyl, amino, phosphate, and sulphydryl groups (Figure 13.2d).

Hydroxyl groups ($-\text{OH}$) are strongly polar and often enter into reactions that link subunits into larger molecular assemblies in which a water molecule is released. These reactions are called condensations. The reverse reaction, in which a water molecule enters a reaction by which a larger molecule is split into two subunits, is called hydrolysis. The formation of a dipeptide from two amino acids is an example of condensation, and its reverse reaction is the hydrolysis of the dipeptide (Figure 13.6a). If the hydroxyl group is bound to a carbon atom, which in turn is bound to other hydrogen and/or carbon

atoms, it is called an alcohol. Alcohols can easily be oxidized to form aldehydes or ketones, which are characterized by their carbonyl group (Figure 13.2d). Aldehydes and ketones are particularly important for carbohydrates (such as sugars) or lipids (such as fats). In aldehydes the carbonyl group occurs at the end of a carbon chain, whereas in ketones it occurs in its interior. A carboxyl group is strongly polar and formed by an alcohol group and an aldehyde group. The hydrogen of the hydroxyl part can easily dissociate as H^+ due to the influence of the nearby carbonyl oxygen. In this way, it acts as an organic acid. The carboxyl group ($-\text{COOH}$) is the characteristic group of organic acids such as fatty acids and amino acids. Amino acids are further characterized by an amino group. Amino groups ($-\text{NH}_2$, Figure 13.2d) have a high chemical reactivity and can act as a base in organic molecules. They are, for instance, essential for the linkage of amino acids to form proteins and for the establishment of hydrogen bonds in DNA double strands. Moreover, amino acids carrying NH_2 in their residual group often play a crucial role as part of the catalytic domain of enzymes. Another group that has several important roles is the phosphate group (Figure 13.2d). As part of large organic molecules, this group acts as a bridging ligand connecting two building blocks to each other, as is the case in nucleic acids (DNA, RNA; see Section 13.2.3) or phospholipids. Furthermore, the di- and triphosphate forms in conjunction with a nucleoside serve as a universal energy unit in cells, for example, adenosine triphosphate (ATP, Figure 13.7a). Phosphate groups are also involved in the regulation of the activity of enzymes, for example, MAP kinases, which participate in signal transduction. Sulphydryl groups (Figure 13.2d) are readily oxidized. If two sulphydryl residues participate in an oxidation, a so-called disulfide bond is created (Figure 13.6d). These linkages often occur between sulphydryl residues of amino acids that form a protein. Thus, they are responsible for the stable folding of proteins, which is required for their correct functioning.

13.2.3 Major Classes of Biological Molecules

The structural and functional properties of an organism are based on a vast number of diverse biological molecules and their interplay. The physicochemical properties of a molecule are determined through their functional groups. In the following sections, four major classes of biological molecules that are ubiquitously present and are responsible for fundamental structural and functional characteristics of living organisms will be introduced: carbohydrates, lipids, proteins, and nucleic acids.

Carbohydrates

Carbohydrates function as energy storage molecules and furthermore can be found as extracellular structure mediators, for example, in plants. The chemical formula of carbohydrates is mostly $C_n(H_2O)_n$. The individual building blocks of all carbohydrates are the monosaccharides, which consist of a chain of three to seven carbon atoms. Depending on the number of carbon atoms, they are

categorized as trioses, tetroses, pentoses, hexoses, or heptoses (cf. Figure 13.3a). All monosaccharides can occur in linear form, and with more than four carbons, they exist in equilibrium with a ring form. In the linear form, all carbons of the chain, except for one, carry a hydroxyl group (polyalcohol), which makes the carbohydrates hydrophilic. The remaining carbon carries a carbonyl group, and depending on its position – whether it is an

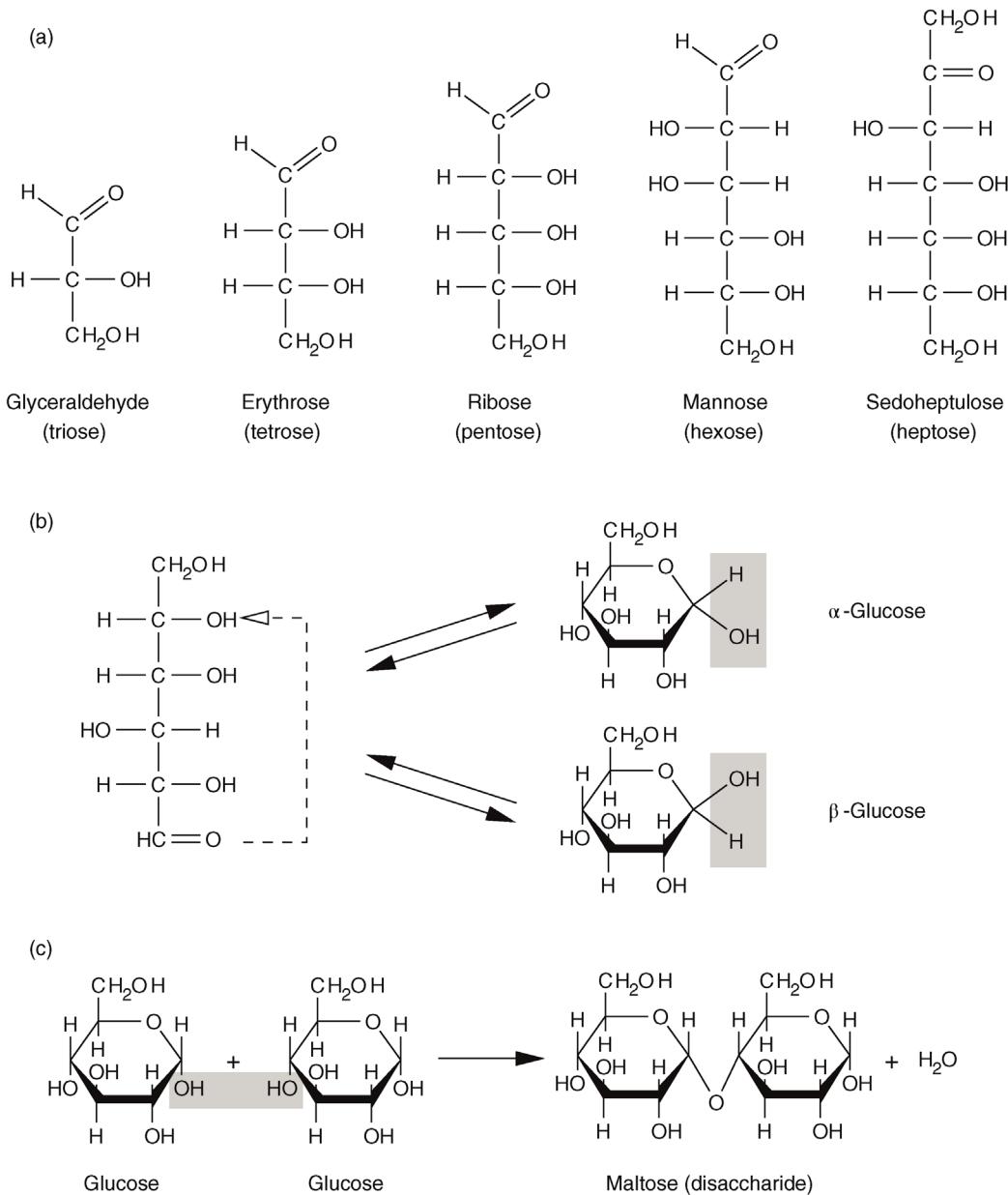


Figure 13.3 Carbohydrates. (a) Some examples of carbohydrates with a backbone of three to seven carbon atoms. (b) Glucose, like other monosaccharides with more than four carbons in their backbone, can form a circular structure, known as hemiacetal, by an intramolecular condensation reaction that can occur in two different conformations. (c) By further condensation reactions, such sugar monomers can form disaccharides or even larger linear or branched molecules called oligomers or polymers depending on the number of monomers involved.

aldehyde or a ketone – it is called an aldose or a ketose. The circular configuration is attained by an intramolecular reaction between the carbonyl group and one of the hydroxyl groups. Such a compound is called a hemiacetal. An example of the ring formation for the six-carbon monosaccharide glucose is given in Figure 13.3b, in which it forms a so-called glucopyranose ring. Depending on the orientation of the hydroxyl group at the 1-carbon, that is, whether it points downward (α -glucose) or upward (β -glucose), two alternate conformations exist. Glucose is one of the most important energy sources for organisms. It is metabolized during glycolysis into ATP and reduction equivalents (e.g., NADH, NADPH, or FADH₂).

The hydroxyl group at the 1-carbon position of the cyclic hemiacetal can react via a condensation with the hydroxyl group of another monosaccharide. This linkage forms a disaccharide from two monosaccharides (Figure 13.3c). If this happens subsequently for several carbohydrates, polysaccharides that occur as linear chains or branching structures are formed.

Lipids

Lipids are a very diverse and heterogeneous group. Since they are made up mostly of nonpolar groups, lipids can be characterized by their higher solubility in nonpolar solvents, such as acetone. Due to their hydrophobic character, lipids tend to form nonpolar associations or membranes. Eventually, these membranes form cellular hydrophilic compartments. Furthermore, such hydrophobic regions offer a local area for reactions that require a surrounding deprived of water. Three different types of lipids are present in various cells and tissues: neutral lipids, phospholipids, and steroids. Lipids can also be linked covalently to proteins or carbohydrates to form lipoproteins or glycolipids, respectively.

Neutral lipids are generally completely nonpolar and are commonly found as storage fats and oils in cells. They are composed of the alcohol glycerol (an alcohol with three hydroxyl groups), which is covalently bound to fatty acids. A fatty acid is a linear chain of 4–24 or more carbon atoms with attached hydrogens (molecules like this are well known as hydrocarbons) and a carboxyl group at one end (Figure 13.4a). Most frequent are chains with 16 or 18 carbons. Fatty acids can be either saturated or unsaturated (polyunsaturated). Unsaturated fatty acids contain one or more double bonds in their carbon chain and have more fluid character than do saturated ones. Linkage of the fatty acids to glycerol results from a condensation reaction of the carboxyl group with one of the alcohol groups of glycerol; this is called an ester binding. If all three sites of the glycerol bind a fatty acid, it is called a triglyceride, which is the most frequent neutral

lipid in living systems. Triglycerides – in form of fats or oils – mostly serve as energy reserves.

Phospholipids are the primary lipids of biological membranes (cf. Section 13.3.1). Their structure is very similar to the neutral lipids. However, the third carbon of glycerol binds a polar residue via a phosphate group instead of a fatty acid. Polar subunits commonly linked to the phosphate group are ethanolamine, choline, glycerol, serine, threonine, or inositol (Figure 13.4c). Due to their polar and apolar parts, phospholipids have dual-solubility properties termed amphipathic or amphiphilic. This property enables phospholipids to form a so-called bilayer in an aqueous environment, which is the fundamental design principle of biological membranes (Figure 13.9a). Polar and nonpolar parts of the amphipathic molecules are ordered side by side in identical orientation and form a one molecule thick layer (monolayer) with a polar and a nonpolar side; the aqueous environment forces the lipophilic sides of two such layers to each other, thus creating the mentioned bilayer.

Steroids are based on a framework of four condensed carbon rings that are modified in various ways (Figure 13.4d). Sterols – the most abundant group of steroids – have a hydroxyl group linked to one end of the ring structure, representing the slightly polar part of the amphiphilic molecule; a nonpolar carbon chain is attached to the opposite end. The steroid cholesterol plays an important part in the plasma membrane of animal cells. Among other things, cholesterol loosens the packing of membrane phospholipids and maintains membrane fluidity at low temperatures. Other steroids act as hormones (substances that regulate biological processes in tissues far away from their own place of production) in animals, and they are, for example, involved in regulatory processes concerning sexual determination or cell growth.

In glycolipids, the lipophilic part is constituted of fatty acids bound to the 1-carbon and 2-carbon of glycerol, as is the case with phospholipids. The 3-carbon is covalently attached to one or more carbohydrate groups that confer an amphiphilic character to the molecule. Glycolipids do occur, for example, in the surface-exposed parts of the plasma membrane bilayer of animal cells that are subject to physical or chemical stress. Furthermore, among several other things, they are responsible for the ABO blood system of humans.

Proteins

Proteins fulfill numerous highly important functions in the cell, only a few of which can be mentioned here. They build up the cytoskeletal framework, which forms the cellular structure and is responsible for cell movements (motility). Proteins are also part of the extracellular

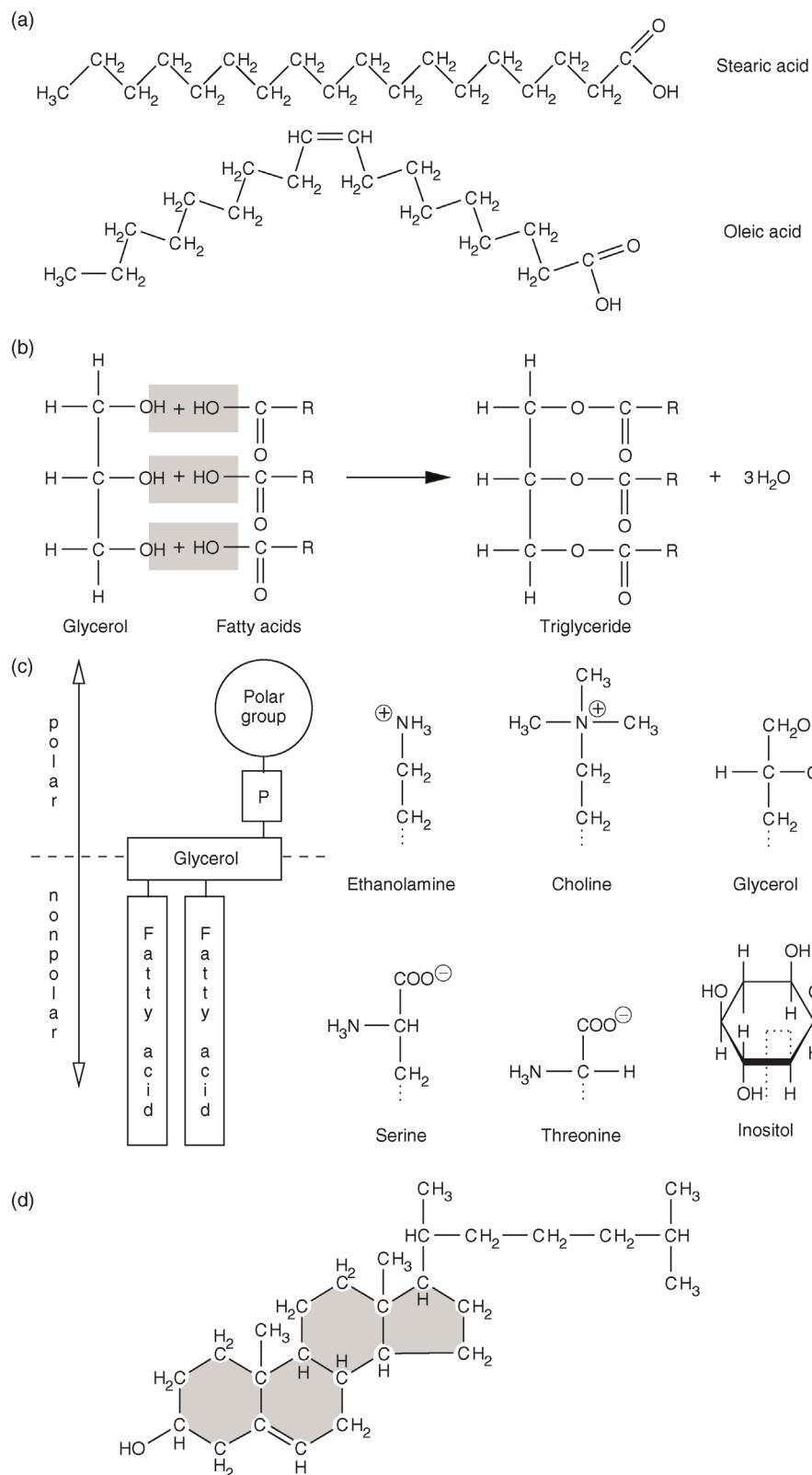


Figure 13.4 (a) Fatty acids represent one part of fats and phospholipids. They are either saturated or unsaturated. (b) Triglycerides are formed by condensation reactions of glycerol and three fatty acids. (c) In phospholipids, the third carbon of glycerol is bound to a polar group via a phosphate group (P), which usually is ethanolamine, choline, glycerol, serine, threonine, or inositol. (d) Steroids constitute another major lipid class. They are formed by four condensed carbon rings. Cholesterol, shown here, is important, for example, for membrane fluidity of eukaryotic cells.

supportive framework (extracellular matrix), for example, as collagen in animals. As catalytic enzymes for highly specific biochemical reactions, they rule and control the metabolism of a single cell or whole organism. Furthermore, proteins regulated by transient modifications are relevant for signal transduction, for example, proteins controlling cell division such as cyclin-dependent protein kinases (CDKs). A further highly important function of proteins is their ability to control the transcription and translation of genes as well as the degradation of proteins (see Section 13.4).

Proteins consist of one or more polypeptides. Each polypeptide is composed of covalently linked amino acids; these covalent bonds are called peptide bonds. Such a bond is formed by a condensation reaction between the amino group of one amino acid and the carboxyl group of another (Figure 13.6a). The primary structure of a polypeptide is coded by the genetic information that defines in which order amino acids – chosen from a set of 20 different ones – do appear. Figure 13.5 shows the chemical structures of these amino acids. Common to all amino acids is a central carbon (α -carbon), which carries an amino group (except for proline where this is a ring-forming imino group), a carboxyl group, and a hydrogen. Furthermore, it carries a residual group with different physicochemical properties, due to which the amino acids can be divided into different groups, such as amino acids that carry (i) nonpolar residues that can grant lipophobic characteristics, (ii) uncharged polar residues, (iii) residues containing a carboxyl group, which are negatively charged at physiological pH and thus act as acids, and (iv) residues that are usually positively charged at common pH ranges of living cells and thus show basic characteristics. Due to the combination of possibilities of these amino acids, proteins are very diverse. Usually proteins are assembled from about 50 to 1000 amino acids, but they might be much smaller or larger. Except for glycine, the α -carbon of amino acids binds four different residues and therefore amino acids can occur in two different isoforms that behave like an image and its mirror image. These two forms are called the L-isoform and the D-isoform, of which only the L-isoform is used in naturally occurring proteins. Furthermore, amino acids of proteins are often altered posttranslationally. For instance, proline residues in collagen are modified to hydroxyproline by addition of a hydroxyl group.

The primary structure of a protein is given by the sequence of the amino acids linked via peptide bonds. This sequence starts at the N-terminus of the polypeptide and ends at its C-terminus (cf. Figure 13.6a). In the late 1930s, Linus Pauling and Robert Corey elucidated the exact structure of the peptide bond. They found that the

hydrogen of the substituted amino group almost always is in opposite position to the oxygen of the carbonyl group, so that both together, with the carbon of the carbonyl group and the nitrogen of the amino group, build a rigid plane. This is due to the fact that the bond between carbon and nitrogen does have a partial double bond character. In contrast to this, both the bonds of the α -carbon with the nitrogen of the substituted amino group and the carbon of the carbonyl group are flexible since they are pure single bonds. The free rotation around these two bonds is limited only by steric interactions of the amino acid residuals. Based on this knowledge, Pauling and Corey proposed two very regular structures: the α -helix and the β -strand. Both are very common in proteins. They are formed by the polypeptide backbone and are supported and stabilized by a specific local amino acid sequence composition. Such regular arrangements are called secondary structures. An α -helix (Figure 13.6b) has a cylindrical helical structure in which the carbonyl oxygen atom of each residue (n) accepts a hydrogen bond from the amide nitrogen four residues further in sequence ($n + 4$). Amino acids often found in α -helices are Glu, Ala, Leu, Met, Gln, Lys, Arg, and His. In a β -sheet, parallel peptide strands – β -strands that may be widely separated in the linear protein sequence – are linked side by side via hydrogen bonds between hydrogen and oxygen atoms of their backbone (Figure 13.6c). The sequence direction (always read from the amino/N-terminal to the carboxyl/C-terminal of the polypeptide) of pairing β -strands can be either parallel or antiparallel. The residual groups of the amino acids point up and down from the β -sheet. Characteristic amino acids of β -sheets are Val, Ile, Tyr, Cys, Trp, Phe, and Thr. The regular secondary structure elements fold into a compact form that is called the tertiary structure of a protein. Its surface topology enables specific interactions with other molecules. Figure 13.6e shows a model of the three-dimensional structure of the superoxide dismutase (SOD), which detoxifies aggressive superoxide radicals ($O_2^{•-}$). Sometimes the tertiary structure is stabilized by posttranslational modifications such as disulfide bridges (Figure 13.6d) or metal ions such as calcium (Ca^{2+}) or zinc (Zn^{2+}). Some proteins are fibrous, that is, they form filamentous structures (e.g., the keratin of hair). But most proteins fold into globular, compact shapes. Larger proteins often fold into several independent structural regions: the domains. Domains frequently consist of 50–350 residues and are often capable of folding stably enough to exist on their own. Often proteins are composed of assemblies of more than one polypeptide chain. Such a composition is termed the quaternary structure. The subunits can be either identical or different in sequence and the protein is thus referred to as a homo-

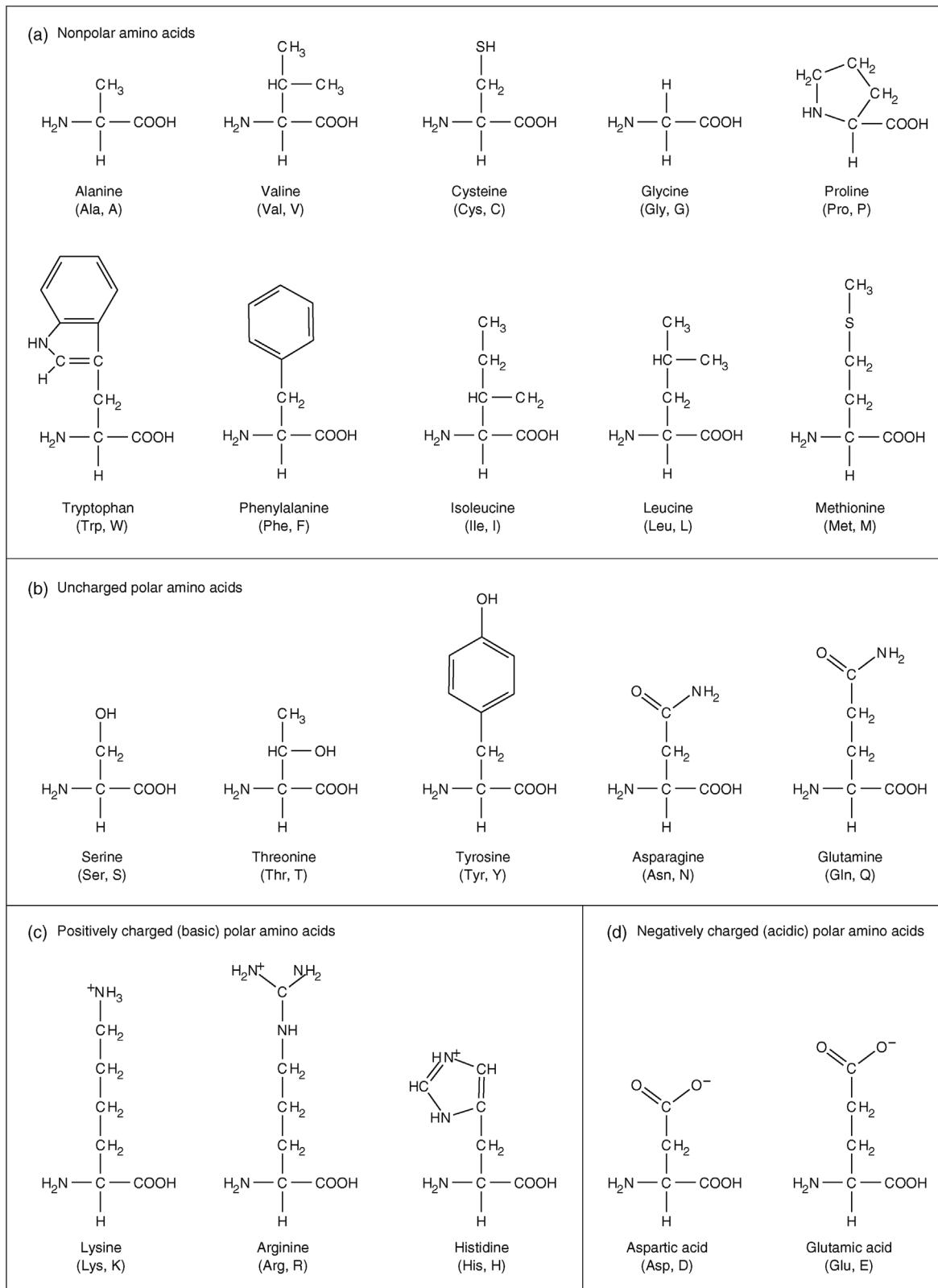


Figure 13.5 Amino acids are formed by carbon that is bound to an amino group, a carboxyl group, a hydrogen, and a residual group. Depending on the physicochemical characteristics of the residual group, they can be categorized as (a) nonpolar, (b) uncharged polar, (c) basic, or (d) acidic amino acids.

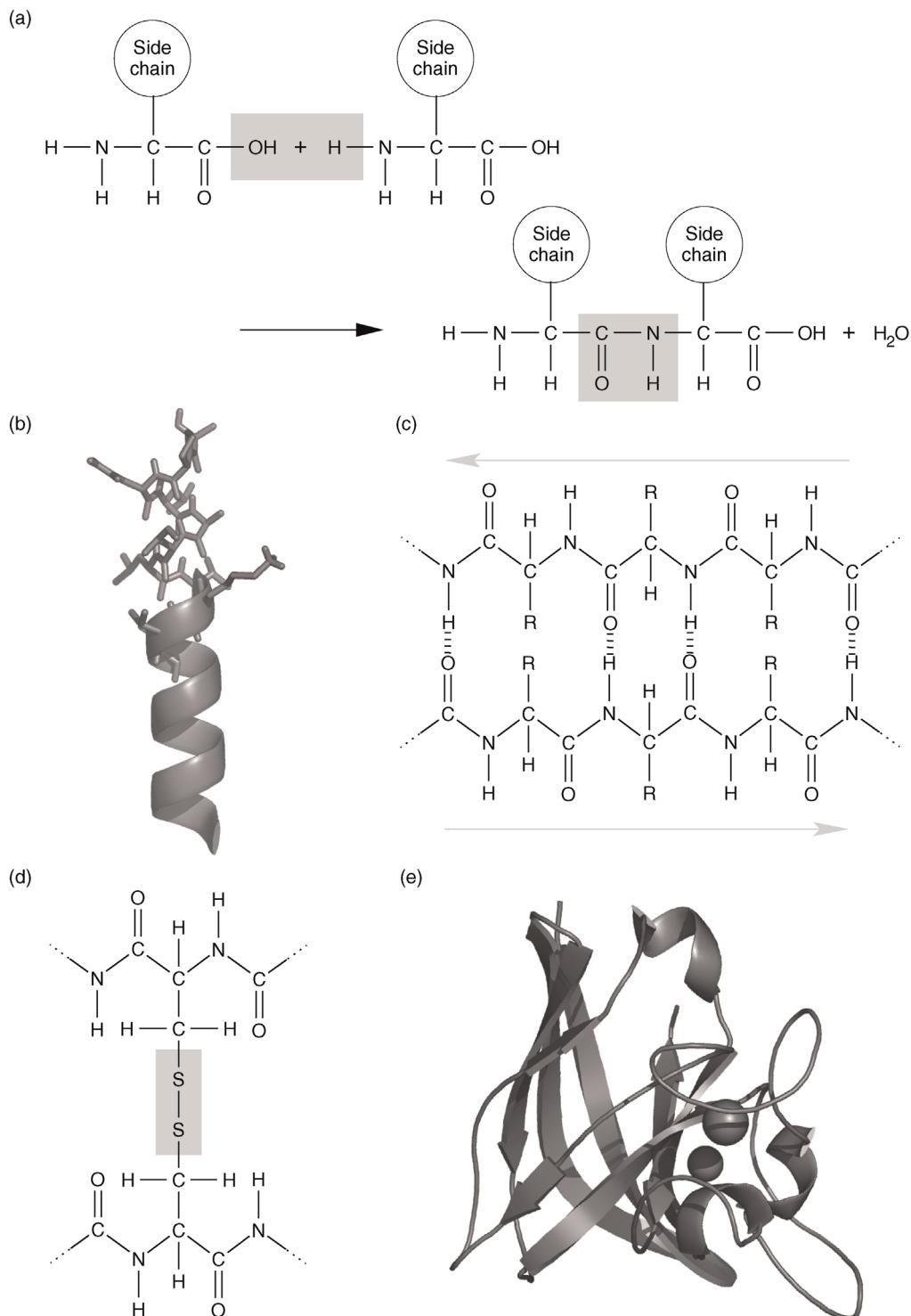


Figure 13.6 (a) Formation of a peptide linkage by a reaction between the carboxyl group of one amino acid and the amino group of the other amino acid. (b) The molecular structure of an α -helix, as shown in the upper part of the image, is often illustrated by a simple helical structure as shown below. (c) An antiparallel β -sheet. (d) A disulfide bridge is formed by oxidation of the SH groups of cysteine residues belonging to either the same or different polypeptides. (e) Three-dimensional illustration of the copper–zinc superoxide dismutase (CuZnSOD) of *E. coli* (PDB: 1EOS). α -Helices are depicted as helical structures and β -strands illustrated by arrows. The two metal ions are shown as spheres.

or heteromer; for example, a protein composed of four identical subunits such as the *lac* repressor is called a homotetramer.

Nucleic Acids

Deoxyribonucleic acid (DNA) is present in all living organisms and is the molecule storing the heredity information, that is, the genes. Another molecule, the ribonucleic acid (RNA), takes part in a vast number of processes. Among these, the transfer of the hereditary information leading from DNA to protein synthesis (via transcription and translation; see Section 13.4) is the most important. Both DNA and RNA are nucleic acids. Nucleic acids are polymers built up of covalently bound mononucleotides. A nucleotide consists of three parts: (i) a nitrogen-containing base, (ii) a pentose, and (iii) one or more phosphate groups (Figure 13.7a). Bases are usually pyrimidines such as cytosine (C), thymine (T), or uracil (U), or purines such as adenine (A) or guanine (G) (Figure 13.7b). In RNA, the base is covalently bound to the first carbon (1'-carbon) of the circular pentose ribose. In DNA, it is bound to the 1'-carbon of deoxyribose, a pentose that lacks the hydroxyl group of the 2'-carbon. A unit consisting of these parts – a base and a pentose – is named nucleoside. If it furthermore carries a mono-, di-, or triphosphate, it is called a nucleotide. Nucleotides are named according to their nucleoside, for example, adenosine monophosphate (AMP), adenosine diphosphate (ADP), or adenosine triphosphate (ATP); prepending *deoxy* to the name (or *d* in the abbreviation) indicates the deoxy form (e.g., deoxyguanosine triphosphate or dGTP). Nucleotides are not only relevant for nucleic acid construction but also responsible for energy transfer in several metabolic reactions (e.g., ATP and ADP) or play certain roles in signal transduction pathways, such as 3'-5' cyclic AMP (cAMP), which is synthesized by the adenylyl cyclase and is involved, for instance, in the activation of certain protein kinases.

In DNA and RNA, the 3'-carbon of a nucleotide is linked to the 5'-carbon of the next nucleotide in sequence via a single phosphate group. These alternating sugar and phosphate groups form the backbone of the nucleic acids. Both DNA and RNA can carry the bases adenine, guanine, and cytosine. In DNA, thymine can also be present, which is replaced by uracil in RNA. The sequence of the different bases has a direction – because of the 5'-3' linkage of its backbone – and is used in living organisms for the conservation of information. DNA contains millions of nucleotides; for example, a single DNA strand of human chromosome 1 is about 246 million nucleotides long. Each base of the sequence is able to pair with a so-called complementary base by hydrogen bonds. Due to the number and steric arrangement of hydrogen bonds,

only two different pairing types are possible (Figure 13.7b): adenine can bind thymine (A–T, with two hydrogen bonds) and guanine can bind cytosine (G–C, with three hydrogen bonds). In RNA, thymine is replaced by uracil. In 1953, Watson and Crick proposed a double strand for the DNA, with an antiparallel orientation of the backbones. Each of the bases of one strand binds to its complementary base on the other strand, and together they form a helical structure (Figure 13.7d). This so-called double helix is the usual conformation of DNA in cells. RNA usually occurs as a single strand. Occasionally, it is paired to a DNA single strand, as during the mRNA synthesis (Section 13.4.1), or complementary bases of the same molecule are bound to each other, for example, as in tRNA.

13.3 Structural Cell Biology

This section gives a general introduction to the structural elements of eukaryotic cells. Fundamental differences between prokaryotic and eukaryotic cells have already been mentioned and are summarized in Table 13.1.

The first microscopic observations of cells were done in the seventeenth century by Robert Hooke and Anton van Leeuwenhoek. The general cell theory was developed in the 1830s by Theodor Schwann and Matthias Schleiden. It states that all living organisms are composed of nucleated cells, which are the functional units of life, and that cells arise only from preexisting cells by a process of division. Today, we know that this is true not only for nucleated eukaryotic cells but also for prokaryotic cells lacking a nucleus. The interior of a cell is surrounded by a membrane that separates it from its external environment. This membrane is called the cell membrane or plasma membrane and it is semipermeable; that is, the traffic of substances across this membrane in either orientation is restricted to some specific molecular species or specifically controlled by proteins of the membrane that handle the transport. Fundamental to eukaryotic cells – in contrast to prokaryotic cells – is their subdivision by intracellular membranes into distinct compartments. Figure 13.8 illustrates the general structure of a eukaryotic cell as found in animals. Generally, one distinguishes between the storage compartment of the DNA, the nucleus, and the remainder of the cell interior that is located in the cytoplasm. The cytoplasm contains further structures that fulfill specific cellular functions and that are surrounded by the cytosol. Among these cytoplasmic organelles are the endoplasmic reticulum (ER), which forms a widely spread intracellular membrane system; the mitochondria, which are the cellular power plants; the Golgi complex; transport vesicles;

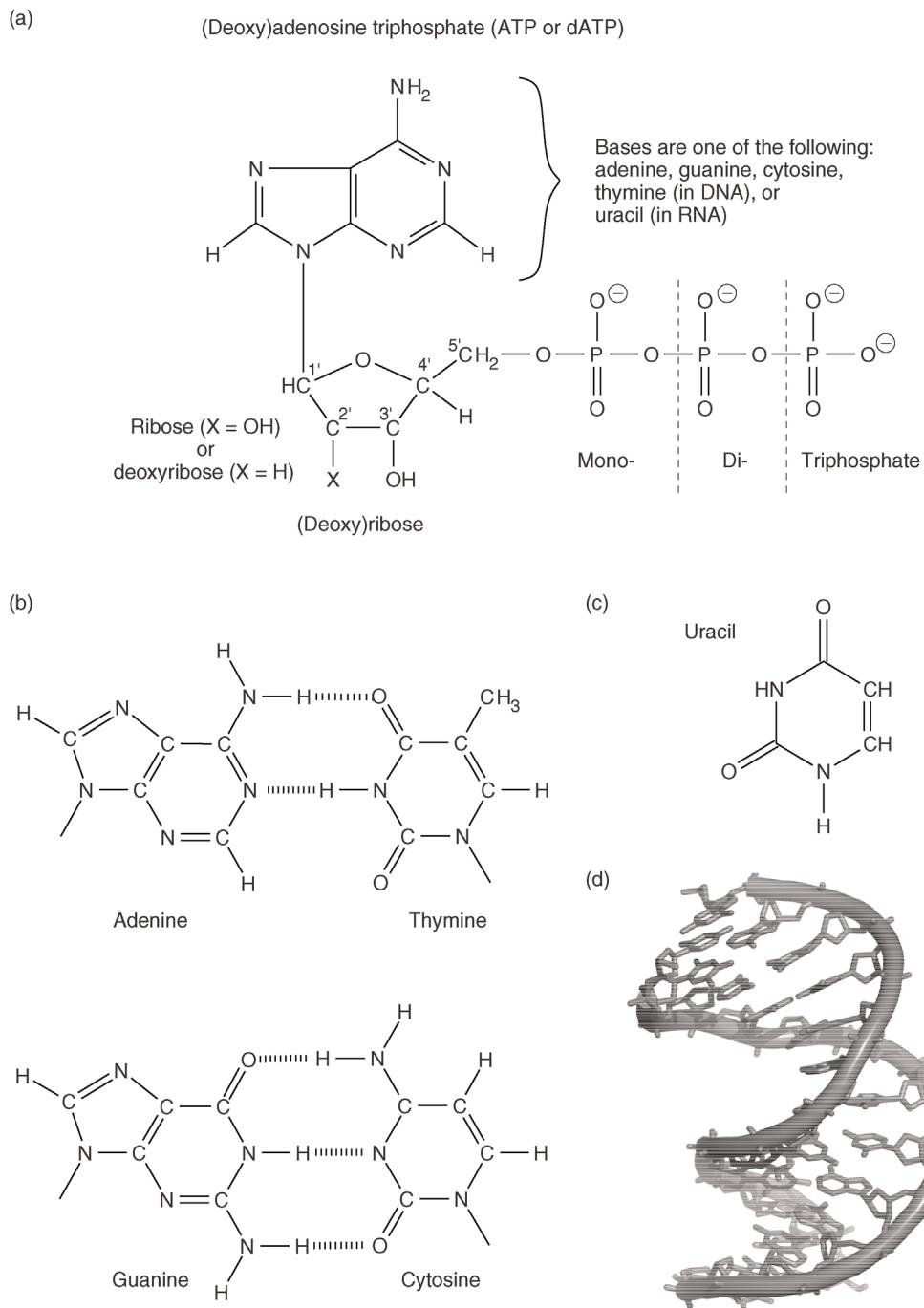


Figure 13.7 (a) Nucleoside phosphates are composed of a ribose or deoxyribose that is linked at its 1'-position to a purine or pyrimidine base. Purines are adenine and guanine, and pyrimidines are thymine, cytosine, or uracil. (b) In DNA, adenine is bound to its complementary base thymine by two hydrogen bonds, and guanine is bound to cytosine by three hydrogen bonds. (c) In RNA, thymine is replaced by uracil. (d) The DNA double helix (PDB: 140D).

peroxisomes; and, additionally in plant cells, chloroplasts, which act as sunlight harvesting systems performing photosynthesis, and the vacuole. In the following sections, we

will describe the structure and function of biological membranes and the most important cellular compartments that are formed by them.

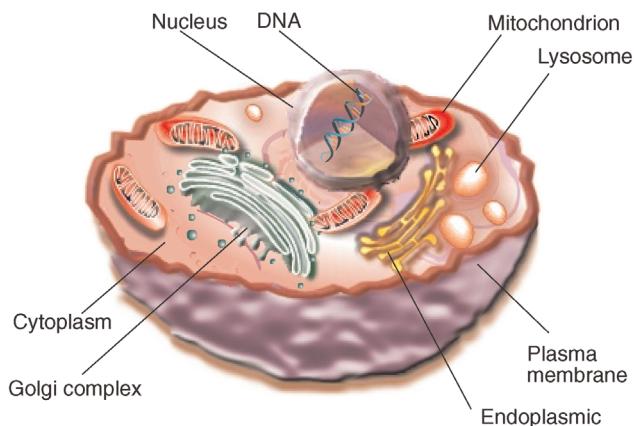


Figure 13.8 Schematic illustration of an animal cell with its major organelles.

13.3.1 Structure and Function of Biological Membranes

All cells are surrounded by a plasma membrane. It not only separates the cell plasma from its surrounding environment but also acts as a selective filter for nutrients and by-products. By active transport of ions, for which the energy source ATP is usually utilized, a chemical and/or electrical potential can be established across the membrane that is essential, for example, for the function of nerve cells. Furthermore, receptor proteins of the plasma membrane enable the transmission of external signals that enable the cell to react to its environment. As already mentioned, eukaryotes additionally possess an intracellular membrane system acting as a boundary for different essential compartments.

The assembly of a bilayer, which is the fundamental structure of all biological membranes, is described in the section about lipids (Section 13.2.3; cf. also Figure 13.9a). Biological membranes are composed of this molecular bilayer of lipids (mainly phospholipids, but also cholesterol and glycolipids) and membrane proteins that are inserted and held in the membrane by noncovalent forces. Besides integral membrane proteins, proteins can also be attached to the surface of the membrane (peripheral proteins). This model of biological membranes is known as the fluid mosaic model and was introduced by Singer and Nicolson [5] (Figure 13.9b). Furthermore, they proposed a possible asymmetric arrangement of adjoining monolayers caused by different lipid composition and orientation of integral proteins, as well as specific occurrence of peripheral proteins in either of the monolayers.

In the plasma membrane, for example, glycolipids always point to the exterior. While an exchange of lipid molecules between the two monolayers – a so-called flip-flop – very rarely occurs by mere chance, lateral movement of lipid molecules takes place frequently. This can also be observed with proteins as long as their movement is not prevented by interaction with other molecules. Lateral movement of lipids depends on the fluidity of the bilayer. The fluidity is strongly enhanced if one of the hydrocarbon chains of the phospholipids is unsaturated and the membrane contains a specific amount of cholesterol.

An important feature of biological membranes is their ability to form a cavity that pinches off as a spherical vesicle, and the reverse process in which the membrane of a vesicle fuses with another membrane and becomes a part of it (Figure 13.9c). This property is utilized by eukaryotic cells for vesicular transport between different intracellular compartments and for the exchange of substances with the exterior. The latter process is termed exocytosis when proteins produced by the cell are secreted to the exterior and endocytosis or phagocytosis when extracellular substances are taken up by the cell.

There are two different kinds of exocytosis. The first one is a constitutive secretion: synthesized proteins packed into transport vesicles at the Golgi complex move to the plasma membrane and fuse with it, thereby delivering their payload to the exterior. This happens, for example, with proteins intended for the extracellular matrix. In the second case termed regulated exocytosis, the proteins coming from the Golgi complex via transport vesicles are enriched in secretory vesicles that deliver their content usually due to an external signal recognized by a receptor and further transmitted via second messengers (e.g., Ca^{2+}). This pathway is common, for example, to neurotransmitters secreted by neurons or digestive enzymes produced by acinar cells of the pancreas.

Vesicular transport is important for large molecules such as proteins. For smaller molecules (e.g., ions or glucose), there are alternative mechanisms. In the case of passive transport, the flux takes place along an osmotic or electrochemical concentration gradient and requires no expenditure of cellular energy. Therefore, either the molecules can diffuse through the membrane or, since especially polar and charged substances cannot pass this hydrophobic barrier, transport is mediated selectively by integral transmembrane proteins. Other transmembrane proteins enable an active transport against a concentration gradient that requires cellular energy (e.g., ATP).

Sensing of exterior conditions and communication with other cells are often mediated by receptors of the cell membrane that tackle the signal transmission.

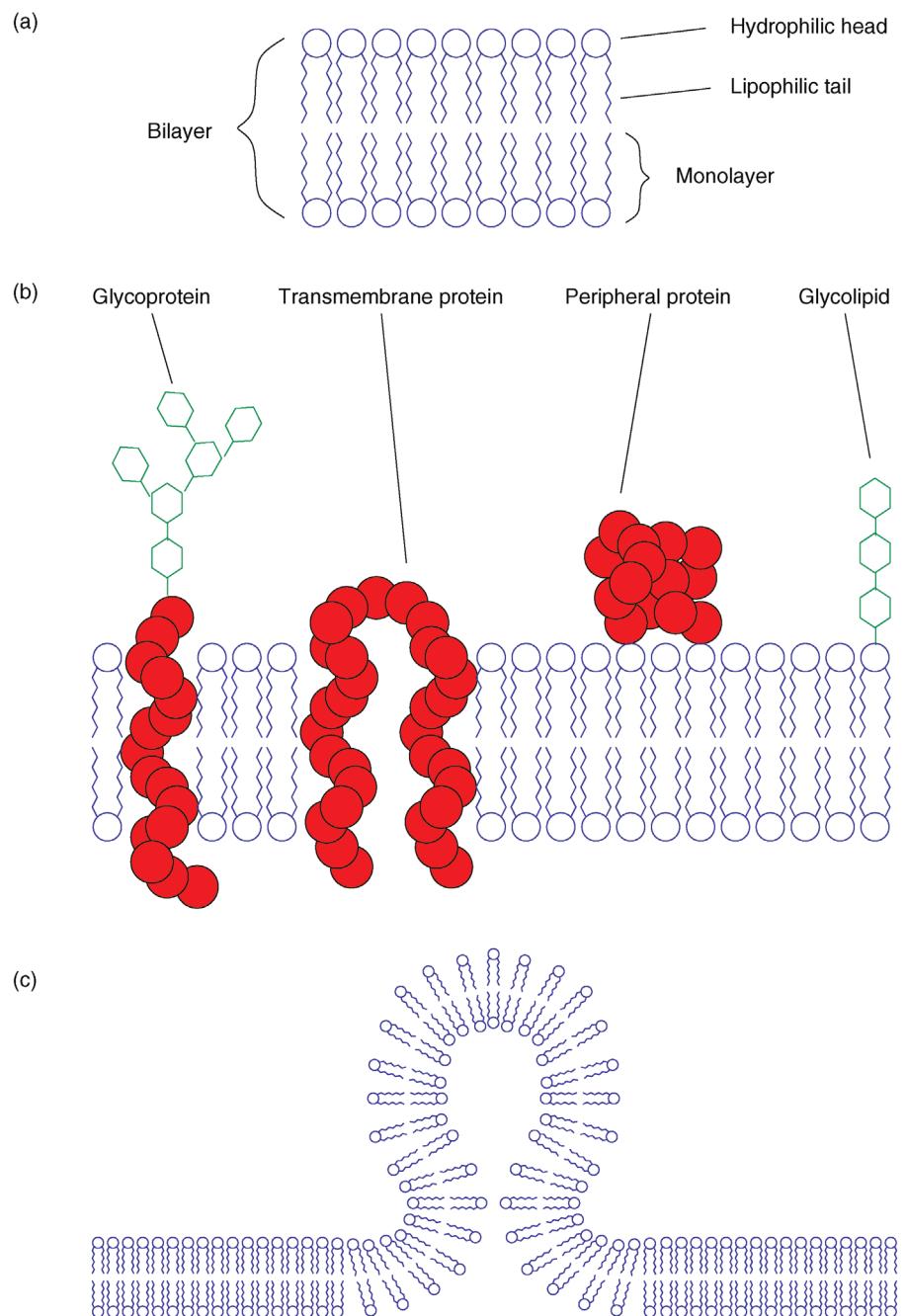


Figure 13.9 (a) In a lipid bilayer, the amphipathic lipids are oriented to both aqueous compartments with their hydrophilic parts. The hydrophobic tails point to the inner membrane space. (b) The fluid mosaic model of a cellular membrane. (c) Formation of a spherical vesicle that is in the process of either pinching off from or fusing with a membrane. Such vesicles are formed during endo- or exocytosis by peripheral proteins inducing the process.

Alternatively, mostly hydrophobic substances such as steroid and thyroid hormones can cross the cell membrane directly and interact with receptors in the cell's interior. A general overview of biochemistry of signal transduction is given, for example, by Krauss [6].

Besides the plasma membrane, plant cells are further surrounded by a cell wall with cellulose, a polysaccharide, as main polymer forming the fundamental scaffold. Prokaryotes also often have a cell wall where different monosaccharides act as building blocks for the polymer.

13.3.2 Nucleus

Prokaryotes store their hereditary information – their genome – in a single, circular, double-stranded DNA (located in a subregion of the cell's interior called the nucleoid) and optionally in one or several small, circular DNAs (the plasmids), which code for further genes. The genome of eukaryotes is located in the cell nucleus and forms the chromatin that is embedded into the nuclear matrix and has dense regions (heterochromatin) and less dense regions (euchromatin). The nucleus occupies about 10% of the cellular volume and is surrounded by the nuclear envelope formed by an extension of the ER that creates a double membrane. The nuclear envelope has several protein complexes that form nuclear pores and that are responsible for the traffic between the nucleus and the cytosol. A subregion of the chromatin in which many repeats of genes encoding ribosomal RNAs (rRNAs) are located appears as a roughly spherical body called nucleolus.

The structure of the chromatin usually becomes optically clearer during cell division, when the DNA strands condense into chromosomes, each consisting of two DNA double strands called chromatids. Both chromatids are joined at the centromere. The ends of the chromatids are called telomeres. At the molecular level, the DNA of a chromosome is highly ordered: The double strand is wound around protein complexes, the histones, and each DNA/histone complex is called a nucleosome.

13.3.3 Cytosol

The cytosol fills the space between the organelles of the cytoplasm. It represents about half of the cell volume and contains the cytoskeletal framework. This fibrous network consists of different protein filaments that constitute a general framework and are responsible for the coordination of cytoplasmic movements. These activities are controlled by three major types of protein filaments: the actin filaments (also called microfilaments), the microtubules, and the intermediate filaments.

The long stretched actin filaments, with a diameter of about 5–7 nm, are built up of globular actin proteins. One major task of actin filaments is the generation of motility during muscle contraction. For the generation of movement, actin filaments slide along another filament type called myosin. This ATP-consuming process is driven by a coordinated interaction of these proteins. Together with other proteins involved in the regulation of muscle activity, these

filaments form very regular structures in muscle cells. Furthermore, in many animal cells, actin filaments associated with other proteins are often located directly under the plasma membrane in the cell cortex and form a network that enables the cell to change its shape and to move.

Another filament type found in eukaryotes are the microtubules. They consist of heterodimers of the proteins α - and β -tubulin, which form unbranched cylinders of about 25 nm in diameter with a central open channel. These filaments are involved, for example, in rapid motions of flagella and cilia, which are hair-like cell appendages. Flagella are responsible for the movement of, for example, sperm and many single-celled eukaryotic protists. Cilia occur, for instance, on epithelial cells of the human respiratory system. The motion of a cilia or flagella is due to the bending of a complex internal structure called axoneme. Almost all kinds of cilia and eukaryotic flagella have nearly the same characteristic structure of the axoneme. This is called the 9+2 structure, because of its appearance: nine doublets that look like two condensed microtubules form a cylinder together with other associated proteins, the center of which contains two further single microtubules. The flexibility of the axoneme is also an ATP-consuming process that is further assisted by the protein dynein.

The third major filament type of the cytoskeleton is the intermediate filament. In contrast to actin filaments and microtubules, which are built of globular proteins, intermediate filaments consist of fibrous proteins. Several subtypes of these filaments are known, for example, keratin filaments in the cytosol of epithelial cells, which make these cells resistant against mechanical influence, or lamin filaments, which are involved in the formation of the nuclear lamina.

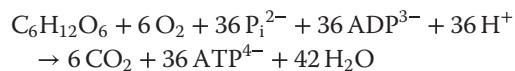
Furthermore, the cytosol contains ribosomes responsible for protein synthesis, and is filled with thousands of metabolic enzymes. A central metabolic pathway that is catalyzed by some of these enzymes is the glycolysis. Substrates of this pathway are glucose or some similar six-carbon derivatives of it. These substrates are converted by several reactions into two molecules of the three-carbon compound pyruvate. Each metabolized glucose molecule generates two molecules of ATP, and one NAD^+ (the oxidized form of nicotinamide adenine dinucleotide) is reduced to NADH. But via this pathway – which does not involve molecular oxygen – only a small amount of the energy that can be gained through oxidation of glucose is made available. In aerobic organisms, the bulk of ATP is produced from pyruvate in the mitochondria (see the following section).

13.3.4 Mitochondria

Mitochondria have a spherical or elongated shape and are about the size of a bacterium. Their interior is surrounded by two membranes: a highly permeable outer membrane and a selective inner membrane. Therefore, mitochondria have two internal compartments, the intermembrane space and the matrix. The outer membrane is permeable for ions and most of the small molecules due to several transmembrane channel proteins called porins. The inner membrane's surface area is strongly increased by numerous folds and tubular projections into the mitochondrial interior, which are called cristae. Mitochondria are partially autonomous; they possess their own DNA and enzymatic complexes required for protein expression (such as ribosomes and mRNA polymerase). Nevertheless, they depend on the symbiosis with their cell since most genes of mitochondrial proteins left the mitochondrial chromosome during evolution and are encoded by the nuclear DNA today. These mitochondrial proteins are synthesized in the cytoplasm and are then imported into the organelle.

As mentioned above, the bulk of ATP (34 out of 36 molecules per metabolized glucose molecule) is gained in mitochondria; thus, they can be termed the "power plants" of eukaryotic cells. The underlying oxidative process that involves molecular oxygen and yields CO_2 and ATP is driven mainly by pyruvate from the glycolysis and fatty acids. Both pyruvate and fatty acids can be converted into acetyl-CoA molecules. Acetyl-CoA has an acetyl group (CH_3CO^- , a two-carbon group consisting of a methyl group and a carbonyl group) that is covalently linked to coenzyme A (CoA). Cytosolic pyruvate can pass the outer mitochondrial membrane and enter the mitochondrial matrix via a transporter of the inner membrane. Pyruvate is then converted into acetyl-CoA by a huge enzyme complex called pyruvate dehydrogenase. Acetyl-CoA reacts with oxaloacetate and thus enters the citrate cycle, a sequence of several reactions during which two CO_2 molecules and energetic reduction equivalents (mainly NADH, but also FADH_2) are produced. Finally, oxaloacetate is regenerated and thus the cycle is closed. The electrons delivered by the reduction equivalents are further transferred step by step onto O_2 , which then reacts together with H^+ ions to form water. The huge amount of energy provided by this controlled oxyhydrogen reaction is used subsequently for the transfer of H^+ ions out of the mitochondrial matrix, thus establishing a H^+ gradient across the inner membrane. The energy provided by this very steep gradient is used by another protein complex of the inner mitochondrial membrane – the ATP synthase – for the production of ATP inside the mitochondrial matrix by

a flux of H^+ from the intermembrane space back into the matrix. This coupled process of oxidation and phosphorylation is called the oxidative phosphorylation. The complete aerobic oxidation of glucose produces as many as 36 molecules of ATP:



13.3.5 Endoplasmic Reticulum and Golgi Complex

The endoplasmic reticulum is a widely spread cytosolic membrane system that forms tubular structures and flattened sacs. Its continuous and unbroken membrane encloses a lumen that stays in direct contact with the perinuclear space of the nuclear envelope. The ER occurs in two forms: the rough ER and the smooth ER. The rough ER forms mainly flattened sacs and has many ribosomes that are attached to its cytosolic surface; the smooth ER lacks ribosomes and forms mostly tubular structures. Proteins destined for secretion but also intended for the ER itself, the Golgi complex, the lysosomes, or the outer plasma membrane enter the lumen of the ER directly after being synthesized by ribosomes of the rough ER. The total amount of ER membranes of a cell as well as the ratio of smooth and rough ER varies strongly depending on species and cell type. All enzymes required for biosynthesis of membrane lipids, such as phosphatidylcholine, phosphatidylethanolamine, or phosphatidylinositol, are located in the ER membrane, their active centers facing the cytosol. Membrane lipids synthesized by these enzymes are integrated into the cytosolic part of the ER bilayer. Since this would result in an imbalance of lipids in the two layers of the membrane, phospholipid translocators can increase the flip-flop rate for specific membrane lipids; thus, the lipid imbalance can be compensated and the membrane asymmetry concerning specific membrane lipids can be established. Furthermore, the ER can form transport vesicles responsible for the transfer of membrane substance and proteins to the Golgi complex.

The Golgi complex (also called Golgi apparatus), usually located in vicinity of the nucleus, consists of piles of several flat membrane cisternae. ER transport vesicles enter these piles at its *cis*-side. Substances leave the Golgi complex at the opposite *trans*-side. Transport between the different cisternae is mediated by Golgi vesicles. Some modifications of proteins by the addition of a specific oligosaccharide happen in the ER, but further glycosylations of various types take place in the lumen of the Golgi complex. Since such modified membrane proteins

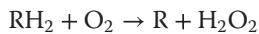
and lipids point to the organelles' inner space, they will be exposed to the cell's outer space when they are transported to the plasma membrane. The synthesis of complex modifications by several additions of carbohydrates requires a special enzyme for each specific addition. Therefore, these reaction pathways become very complex.

13.3.6 Other Organelles

Eukaryotic cells have further compartments for certain functions. Some of these organelles and their major functions will be mentioned briefly here.

Lysosomes are responsible for the intracellular digestion of macromolecules. These vesicular organelles contain several hydrolyzing enzymes (hydrolases), for example, proteases, nucleases, glycosidases, lipases, phosphatases, and sulfatases. All of them have their optimal activity at pH 5. This pH value is maintained inside the lysosomes via ATP-dependent H⁺ pumps (for comparison, the pH of the cytosol is about 7.2).

Peroxisomes (also called microbodies) contain enzymes that oxidize organic substances (R) and use therefore molecular oxygen as an electron acceptor. This reaction produces hydrogen peroxide (H₂O₂).



H₂O₂ is used by peroxidase to further oxidize substances such as phenols, amino acids, formaldehyde, and ethanol, or it is detoxified by catalase (2 H₂O₂ → 2 H₂O + O₂).

In contrast to the ER, the Golgi cisternae, lysosomes, peroxisomes, and vesicles, which are surrounded by a single membrane, chloroplasts, as well as mitochondria, have a double membrane of which the inner one is not folded into cristae as in mitochondria. Instead, a chloroplast has a third membrane that is folded several times and forms areas that look like piles of coins. This membrane contains light harvesting complexes and ATP synthases that utilize the energy of the sunlight for the production of cellular energy and reduction equivalents used for the fixation of carbon dioxide (CO₂) into sugars, amino acids, fatty acids, or starch. Chloroplasts, as well as mitochondria, have own circular DNA and ribosomes.

13.4 Expression of Genes

Classically, a gene is defined as the information encoded by the sequence of a DNA region that is required for the construction of an enzyme or – more generally – of a protein. We will see that this is a simplified definition,

since, for example, mature products of some genes are not proteins but RNAs with specific functions; eukaryotic gene sequences in particular also contain noncoding information. The term gene expression commonly refers to the whole process during which the information of a particular gene is translated into a particular protein. This process involves several steps. First, during transcription (Figure 13.10, ①), the DNA region encoding the gene is transcribed into a complementary messenger RNA (mRNA). In eukaryotic cells, this mRNA is further modified (②) inside the nucleus and transferred to the cytosol (③). In the cytosol, the mRNA binds to a ribosome that uses the sequence as a template for the synthesis of a specific polypeptide that can fold into the three-dimensional protein structure (④). In prokaryotic cells, the mRNA is not further modified and ribosomes can bind to the nascent mRNA during transcription.

In eukaryotic cells, the synthesized proteins can either remain in the cytosol (⑤) or, if they have a specific signaling sequence, be synthesized by ribosomes of the rough ER and enter its lumen (⑦). However, there are several mechanisms of directing each protein to its final destination. During this sorting, proteins are often modified, for example, by cleavage of signaling peptides or by glycosylations.

All the genes of a single organism make up its genome. But only a subset of these genes will be expressed at a particular time or in a specific cell type. Some genes fulfill basic functions of the cell and are always required; these are called constitutive or housekeeping genes. Others are expressed only under certain conditions. The amount of a gene product, for example, a protein, depends mainly on its stability and the number of its mRNA templates. The number of the latter depends on the transcription rate, which is influenced by regulatory regions of the gene and transcription factors that control the initialization of transcription. Thus, quantitative changes in gene expression can be monitored by mRNA and protein concentrations (see Chapter 14 on experimental techniques used for this purpose). Rate changes in any production or degradation step of a specific gene product, which might happen in different cell types or developmental stages, can lead to differential gene expression.

The whole procedure of gene expression, protein sorting, and posttranslational modifications is summarized in Figure 13.10 and will be described in more detail in the following sections.

13.4.1 Transcription

The synthesis of an RNA polymer from ATP, GTP, CTP, and UTP employing a DNA region as a template is called

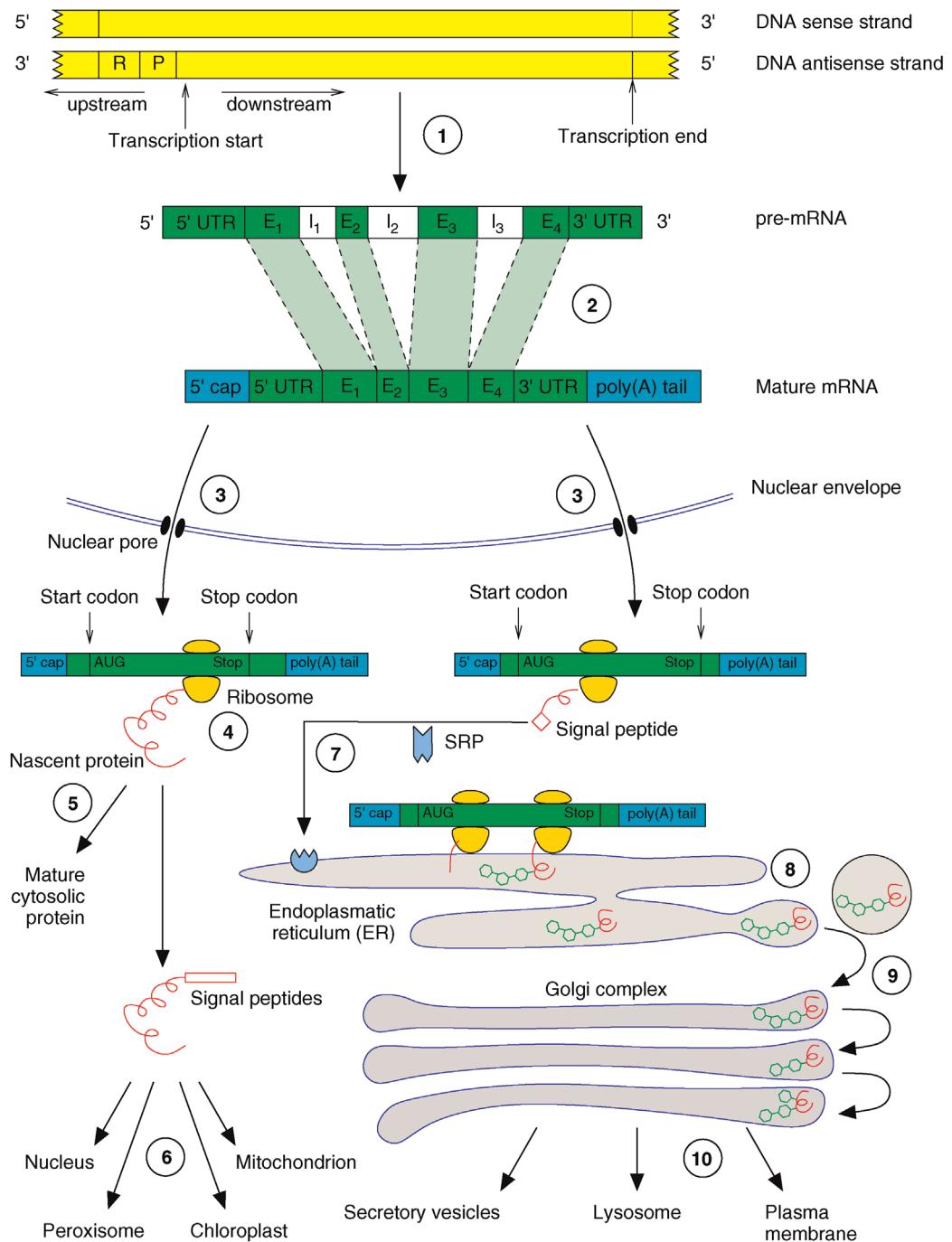


Figure 13.10 Gene expression in eukaryotic cells comprises several steps from the DNA to the mature protein at its final destination. This involves the (①) transcription of the gene, (②) splicing and processing of the pre-mRNA, (③) export of the mature mRNA into the cytosol, (④) translation of the genetic code into a protein, and (⑤–⑩) several steps of sorting and modification. More details are given in the text.

transcription. RNA synthesis is catalyzed by the RNA polymerase. In eukaryotic cells, there are different types of this enzyme that are responsible for the synthesis of different RNA types, including mRNA, rRNA, or transfer RNA (tRNA). In prokaryotic cells, all these different RNA types are synthesized by the same polymerase. This

enzyme has an affinity to a specific DNA sequence, the promoter, that also indicates the first base to be copied. During initiation of transcription, the RNA polymerase binds to the promoter with a high affinity that is supported by further initiation factors. Complete formation of the initiation complex causes the DNA to unwind in

the promoter region. Now the enzyme is ready to add the first RNA nucleoside triphosphate to the template strand of the opened DNA double strand. In the subsequent elongation phase, the RNA polymerase moves along the unwinding DNA and extends the newly developing mRNA continuously with nucleotides complementary to the template strand. During this phase, a moving transient double-stranded RNA–DNA hybrid is established. As the polymerase moves along, the DNA rewinds again just behind it. As RNA synthesis always proceeds in the $5' \rightarrow 3'$ direction, only one of the DNA chains acts as template, the so-called antisense (–) strand. The other one, the sense (+) strand, has the same sequence as the transcribed RNA, except for the thymine nucleotides that are replaced by uracil nucleotides in RNA. As much as the promoter is responsible for initiation of transcription, the terminator – another specific DNA sequence – is responsible for its termination. For the bacterium *Escherichia coli*, two different termination mechanisms are described: the Rho-independent and the Rho-dependent termination. In Rho-independent termination, the transcribed terminator region shows two short GC-rich and self-complementary sequences that can bind to each other and thus form a so-called hairpin structure. This motif is followed by a block of uracil residues that bind the complementary adenine residues of the DNA only weakly. Presumably, this RNA structure causes the RNA polymerase to terminate and release the RNA. In Rho-dependent termination, a protein – the Rho factor – can bind the newly synthesized RNA near the terminator and mediate the RNA release. Termination in eukaryotic cells shows both similarities to and differences from the mechanisms found in bacteria.

13.4.2

Processing of the mRNA

In eukaryotic cells, the primary mRNA transcript (precursor mRNA or pre-mRNA) is further processed before being exported into the cytosol and entering translation (Figure 13.10, ②). The protein-coding sequence lies internally in the mRNA and is flanked on both sides by nucleotides that are not translated. During processing, a so-called 5' cap is attached to the flanking 5' untranslated region (5' UTR, about 10–200 nucleotides) preceding, or lying upstream of, the coding sequence. This 5' cap consists of three nucleotides that are further modified. The 3' untranslated region (3' UTR) of most mRNAs is also modified after transcription by addition of a series of about 30–200 adenine nucleotides that are known as the poly(A) tail. Furthermore, the pre-mRNA is often much longer than the mature RNA because the coding sequence is often interrupted by one or several

intervening sequences called introns, which do not occur in the mature mRNA exported to the cytosol. These intron sequences are removed during processing by a mechanism called splicing. The remaining sequences are called exons. The final coding sequence thus consists of a series of exons joined together. It starts with AUG, which is the first triplet being translated into an amino acid, and it stops with a stop codon (UGA, UAA, or UAG). Via the pores of the nuclear envelope, the mature mRNA is finally exported to the cytoplasm, where the translation process takes place.

13.4.3 Translation

Translation of the genetic information encoded by the mRNA into the amino acid sequence of a polypeptide is done by ribosomes in the cytosol. To encode the 20 different amino acids occurring in polypeptides, at least three bases out of the four possibilities (G, U, T, C) are necessary ($4^3 = 64 > 20$). During evolution, a code developed that uses such triplets of exactly three bases, which are called codons, to code the amino acids and signals for start and end of translation. By using three bases for each codon, more than 20 amino acids can be coded, and hence some amino acids are encoded by more than one triplet. The genetic code is shown in Table 13.3. It is

Table 13.3 The genetic code.

Position 1 (5' end)	Position 2				Position 3 (3' end)
	U	C	A	G	
U	Phe	Ser	Tyr	Cys	U
	Phe	Ser	Tyr	Cys	C
	Leu	Ser	Stop	Stop	A
	Leu	Ser	Stop	Trp	G
C	Leu	Pro	His	Arg	U
	Leu	Pro	His	Arg	C
	Leu	Pro	Gln	Arg	A
	Leu	Pro	Gln	Arg	G
A	Ile	Thr	Asn	Ser	U
	Ile	Thr	Asn	Ser	C
	Ile	Thr	Lys	Arg	A
	Met	Thr	Lys	Arg	G
G	Val	Ala	Asp	Gly	U
	Val	Ala	Asp	Gly	C
	Val	Ala	Glu	Gly	A
	Val	Ala	Glu	Gly	G

Each codon of the genetic code – read in the $5' \rightarrow 3'$ direction along the mRNA – encodes a specific amino acid or a starting or termination signal of translation.

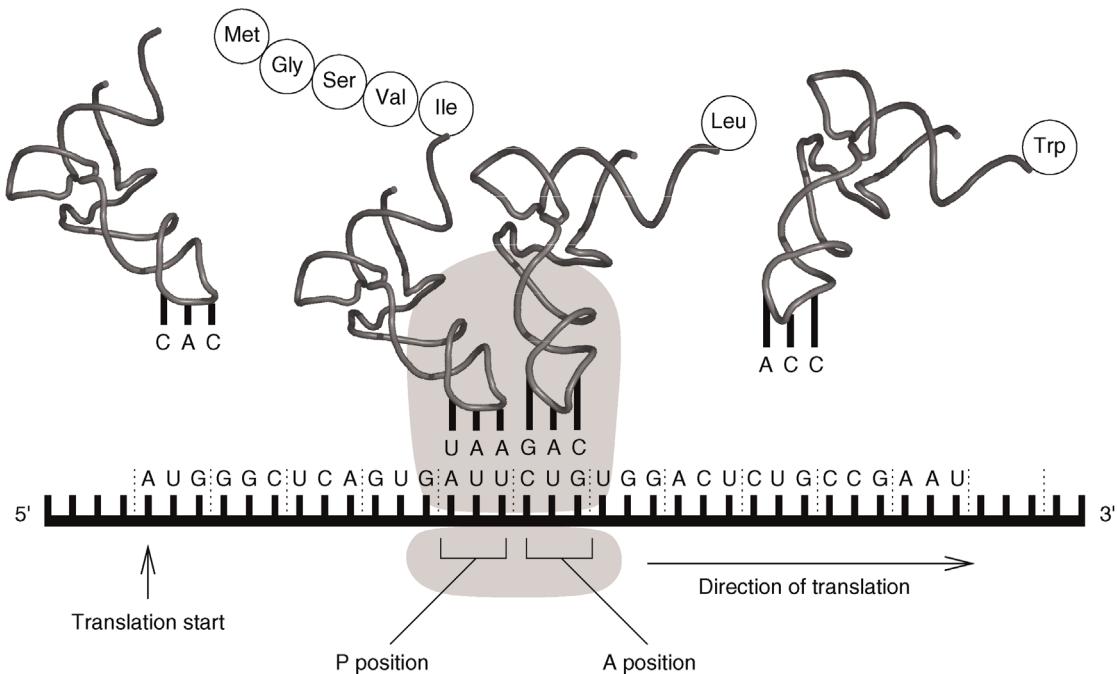


Figure 13.11 During translation, the genetic information of the mRNA is converted into the corresponding polypeptide. More details are given in the text.

highly conserved across almost all prokaryotic and eukaryotic species except for some mitochondria or chloroplasts. For translation of the genetic information, adapter molecules are required. These are the transfer RNAs. They consist of about 80 nucleotides and are folded into a characteristic form similar to an "L". Each tRNA can recognize a specific codon by a complementary triplet, called an anticodon, and it can also bind the appropriate amino acid. For each specific tRNA, a certain enzyme (aminoacyl tRNA synthetase) attaches the right amino acid to the tRNA's 3' end. Such a loaded tRNA is called an aminoacyl tRNA.

During translation (Figure 13.11), the genetic information of the mRNA is read codon by codon in the 5' → 3' direction of the mRNA, starting with an AUG codon. AUG codes for methionine, and therefore newly synthesized proteins always begin with this amino acid at their amino terminus. Protein biosynthesis is catalyzed by ribosomes. Both eukaryotic and prokaryotic ribosomes consist of a large and a small subunit, and both subunits are composed of several proteins and rRNAs. In eukaryotic cells, the small ribosomal subunit first associates with an initiation tRNA (Met-tRNA_i) and binds the mRNA at its 5' cap. Once attached, the complex scans along the mRNA until reaching the start AUG codon. In most cases, this is the first AUG codon in the 5' → 3' direction. This position indicates the translation start and determines the reading frame. Finally, during initiation the

large ribosomal subunit is added to the complex and the ribosome becomes ready for protein synthesis. Each ribosome has three binding sites: one for the mRNA and two for tRNAs. In the beginning, the first tRNA binding site, also called P site, contains the initiation tRNA. The second or A site is free to be occupied by an aminoacyl tRNA that carries an anticodon complementary to the second codon. Once the A site is filled, the amino acid at the P site, which is the methionine, establishes a peptide bond with the amino group of the amino acid at the A site. Now the unloaded tRNA leaves the P site and the ribosome moves one codon further downstream. Thus, the tRNA carrying the dipeptide enters the P site and the A site is open for another aminoacyl tRNA, which is complementary to the third codon in sequence. This cycle is repeated until a stop codon (UAA, UAG, or UGA) is reached. Then the newly synthesized polypeptide detaches from the tRNA and the ribosome releases the mRNA. It is obvious that the addition or alteration of nucleotides of a gene can lead to changes in the reading frame or to the insertion of false amino acids, which might result in malfunctioning proteins. Such changes can happen by mutations, which are random changes of the genomic sequence of an organism that either occur spontaneously or are caused by chemical substances or radiation. A mutation can be either an exchange of a single nucleotide by another or some larger rearrangement. Even the exchange of a single nucleotide

by another might severely influence the function of an enzyme, if it occurs, for example, in the sequence coding for its active site.

13.4.4 Protein Sorting and Posttranslational Modifications

Cells possess a sorting and distribution system that routes newly synthesized proteins to their intra- or extracellular destination. This is mediated by signal peptides – short sequences of the polypeptide occurring at diverse positions. The sorting begins during translation when the polypeptide is synthesized by either a free ribosome or one that becomes attached to the ER membrane. The latter occurs if the growing polypeptide has a signal sequence at its amino terminus that can be recognized by a specific signal recognition particle (SRP) that routes it to a receptor located in the ER membrane (Figure 13.10, ⑦). Such polypeptides are transferred into the ER lumen, where the signal peptide is cleaved off.

Peptides synthesized in the cytosol (Figure 13.10, ④) either remain in the cytosol (⑤), if not possessing a specific signal sequence, or are routed further to a mitochondrion, chloroplast, peroxisome, or the nucleus (⑥). The nuclear localization sequence (NLS) is usually located inside the primary sequence of the protein and is not found terminally; thus, it is not cleaved from the protein as happens with many other signal peptides. Similarly, some transmembrane proteins synthesized by ribosomes of the rough ER have internal signal peptides that are required for correct routing to the membrane.

Polypeptides entering the ER after synthesis are usually further modified by glycosylations, where oligosaccharides are bound to specific positions of the newly synthesized proteins (⑧). Most proteins entering the ER do not remain in the ER but are transferred via transport vesicles to the Golgi complex (⑨), where further modifications of the bound oligosaccharides and additional glycosylations take place. If the proteins are not intended to remain in the Golgi complex, they are further transferred into lysosomes or secretory vesicles or they become transmembrane protein complexes of the plasma membrane (⑩).

13.4.5 Regulation of Gene Expression

The human genome presumably contains about 20 000–25 000 protein-coding genes, with an average coding length of about 1400 base pairs (bp) and an average genomic extent of about 30 kb (1 kb = 1000 bp). This would mean that only about 1.5% of the human genome

consists of coding sequences and only one-third of the genome would be transcribed in genes [7,8]. Besides coding sequences, also regulatory sequences are known that play important roles in particular through control of replication and transcription. The remaining noncoding genomic DNA that does not yet appear to have any function is often referred to as “junk DNA.”

Since only a small subset of all the genes of an organism must be expressed in a specific cell (e.g., detoxification enzymes produced by liver cells are not expressed in epidermal cells), there must be regulatory mechanisms that repress or specifically induce the expression of genes. This includes mechanisms that control the level of gene expression.

In 1961, François Jacob and Jacques Monod proposed a first model for the regulation of the *lac* operon, a genetic region of the *E. coli* genome that codes for three genes required for the utilization of the sugar lactose by this bacterium. These genes are activated only when glucose is missing but lactose, as an alternative carbon source, is present in the medium. The transcription of the *lac* genes is under the control of a single promoter, which overlaps with a regulatory region lying downstream called operator to which a transcription factor, a repressor, can bind. Jacob and Monod introduced the term operon for such a polycistronic gene. (The term cistron is defined as the functional genetic unit within which two mutations cannot complement. The term is often used synonymous with gene and describes the region of DNA that encodes a single polypeptide [or functional RNA]. Thus, the term polycistronic refers to a DNA region encoding several polypeptides. Polycistronic genes are known only for prokaryotes.)

Besides the negative regulations or repressions mediated by a repressor, positive regulations or activations that are controlled by activators are also known. An activator found in *E. coli* that is also involved in the catabolism of alternative carbon sources is the catabolite activator protein (CAP). Since the promoter sequence of the *lac* operon shows only low agreement to the consensus sequence of normal *E. coli* promoters, the RNA polymerase has only a weak affinity to it. (The consensus sequence of a promoter is a sequence pattern that shows highest sequence similarity to all promoter sequences to which a specific RNA polymerase can bind.) The presence of CAP, which indicates the lack of glucose, enhances the binding affinity of RNA polymerase to the *lac* promoter and thus supports the initiation of transcription.

The regulation of gene expression in eukaryotic cells is more complicated than in prokaryotic cells. In contrast to the bacterial RNA polymerase that recognizes specific DNA sequences, the eukaryotic enzymes require a protein/DNA complex that is established by general

Exercises

- 13.1. What are the different structures and conformations a protein can have and by which properties is the protein conformation defined?
- 13.2. Why is it necessary that the protein sequences are encoded in the DNA by nucleotide triplets and not by nucleotide duplets?
- 13.3. Why are proteins of thermophilic bacteria not rapidly denatured by the high temperatures these organisms are exposed to?
- 13.4. What is the purpose of posttranslational modifications? List six functional groups that are used for posttranslational modifications.
- 13.5. What is the purpose of introns and why do eukaryotes have introns but prokaryotes do not?
- 13.6. What is the benefit of cellular compartments?
- 13.7. Why do most transmembrane proteins have their N-terminus outside and the C-terminus inside?
- 13.8. If a eukaryotic cell has lost all its mitochondria (let's say during mitosis one daughter cell got none), how long does it take to regrow them?

transcription factors. One of these transcription factors (TFIIB) binds the so-called TATA-box – a promoter sequence occurring in most protein-coding genes with the consensus sequence TATAAA. Besides these general transcription factor binding sites, most genes are further regulated by a combination of sequence elements lying in the vicinity of the promoter and enhancer sequence elements located up to 1000 nucleotides or more upstream of the promoter.

Regulation of gene expression not only is carried out by transcriptional control but can also be controlled during processing and export of the mRNA into the cytosol, by the translation rate, by the decay rates of the mRNA and the protein, and by control of the protein activity.

References

- 1 Alberts, B. *et al.* (2008) *Molecular Biology of the Cell*, Garland Science.

- 2 Reece, J.B. *et al.* (2013) *Campbell Biology*, Pearson.
- 3 Miller, S.L. and Urey, H.C. (1959) Organic compound synthesis on the primitive earth. *Science*, 130, 245–251.
- 4 Wächtershäuser, G. (1988) Before enzymes and templates: theory of surface metabolism. *Microbiol. Rev.*, 52, 452–484.
- 5 Singer, S.J. and Nicolson, G.L. (1972) The fluid mosaic model of the structure of cell membranes. *Science*, 175, 720–731.
- 6 Krauss, G. (2003) *Biochemistry of Signal Transduction and Regulation*, 3rd edn, Wiley-VCH Verlag GmbH, Weinheim.
- 7 Lander, E.S. *et al.* (2001) Initial sequencing and analysis of the human genome. *Nature*, 409, 860–921.
- 8 Venter, J.C. *et al.* (2001) The sequence of the human genome. *Science*, 291, 1304–1351.

Further Reading

- Alberts, B., Johnson, A., Lewis, J., Morgan, D., Raff, M., Roberts, K., and Walter, P. (2014) *Molecular Biology of the Cell*, 6th edn, Garland Science.

Experimental Techniques

14

Summary

The development of experimental techniques for the purification, amplification, and investigation of biologically relevant molecules is of crucial importance for our understanding of how living cells and organisms work.

With the discovery of restriction endonucleases and ligases in the 1960s and 1970s, it was possible to cut DNA at specific positions and join pieces together in new combinations. Since then, a large collection of methods have been developed to manipulate DNA in almost arbitrary ways. With the help of the polymerase chain reaction (PCR), pieces of DNA can be amplified billion-fold, and together with next-generation sequencing techniques, whole-genome sequencing projects are becoming a routine procedure.

The separation and analysis of mixtures of proteins is also of great importance and the researcher can nowadays choose from a diverse variety of gel and chromatographic techniques. A technologically very advanced and demanding technique for the identification of peptides and proteins is mass spectrometry (MS). Using a database approach, it is possible to identify known proteins and also to characterize post-translational modifications.

An important driving force of current developments is miniaturization, leading to high-throughput methods. Prominent representatives of this approach are DNA and protein chips. DNA microarrays and high-throughput sequencing allow us to measure simultaneously the expression of thousands of different genes, and protein chips can be used to study interactions of proteins with antibodies, other proteins, DNA, or small molecules. Large numbers of transcription factor (TF) binding sites can be measured *in vivo* using ChIP-on-Chip, and single-cell methods allow us to measure heterogeneities that exist in organisms at the microscopic level.

14.1 Restriction Enzymes and Gel Electrophoresis

14.2 Cloning Vectors and DNA Libraries

14.3 1D and 2D Protein Gels

14.4 Hybridization and Blotting Techniques

- Southern Blotting
- Northern Blotting
- Western Blotting
- *In Situ* Hybridization

14.5 Further Protein Separation Techniques

- Centrifugation
- Column Chromatography

14.6 Polymerase Chain Reaction

14.7 Next-Generation Sequencing

14.8 DNA and Protein Chips

- DNA Chips
- Protein Chips

14.9 RNA-Seq

14.10 Yeast Two-Hybrid System

14.11 Mass Spectrometry

14.12 Transgenic Animals

- Microinjection and ES Cells
- Genome Editing Using ZFN, TALENs, and CRISPR

14.13 RNA Interference

14.14 ChIP-on-Chip and ChIP-PET

14.15 Green Fluorescent Protein

14.16 Single-Cell Experiments

14.17 Surface Plasmon Resonance

Exercises

References

Exercises

In this chapter, we will provide a description of elementary experimental techniques used in modern molecular biology. In the same way as Chapters 13 and 15 are only introductions to biology and mathematics, this chapter is only an introduction to the large arsenal of experimental techniques that are used and is not meant to be a comprehensive overview. We felt, however, that for readers without experimental background, it might be interesting and helpful to get a basic idea of the techniques that are used to actually acquire the immense biological knowledge that is nowadays available. A basic understanding of the techniques is also indispensable for understanding experimental scientific publications or simply discussing experiments with colleagues.

The order in which the different techniques are presented corresponds roughly to their historical appearance and complexity. Some of these techniques are of special interest, because they are able to generate large quantities of data (high-throughput techniques) that can be used for quantitative modeling in systems biology.

14.1 Restriction Enzymes and Gel Electrophoresis

We have seen in the last chapter that the genes, which code for the proteins of a cell, are all located on very long pieces of DNA, the chromosomes. To isolate individual genes, it is therefore necessary to break up the DNA and isolate the fragment of interest. However, until the early 1970s, this was a very difficult task. DNA consists of only four different nucleotides, making it a very homogeneous and monotonous molecule. In principle, the DNA can be broken into smaller pieces by mechanical shear stress. However, this results in random fragments, which are not useful for further processing.

This situation began to change when the first restriction endonucleases were isolated from bacteria at the end of the 1960s. These enzymes recognize specific short sequences of DNA and cut the molecule only at these positions (type II restriction enzymes). Restriction enzymes are part of a bacterial defense system against bacteriophages (prokaryote-specific viruses). The

recognition sequences are typically between 4 and 8 bp long. Methylases form the second part of this defense system. They modify DNA molecules at specific sequences by adding methyl groups to the nucleotides of the target sequence. The DNA of the bacterium is methylated by the methylase, which protects it against the nuclease activity of the restriction enzyme. But the DNA of a phage that enters the cell is not methylated and hence is degraded by the restriction enzyme.

Most restriction enzymes cut the double helix in one of three different ways, as depicted by Figure 14.1. Some produce blunt ends, but others cut the DNA in a staggered way resulting in short stretches of single-stranded DNA (ssDNA) (here 4 bp), called sticky ends. Sometimes the cutting site is some distance away from the recognition site (18 bp in case of *MmeI*), which is the basis for recent experimental techniques (see Section 14.14).

Restriction enzymes generate reproducibly specific fragments from large DNA molecules. This is a very important advantage over the random fragments that can be generated by shear forces. If the restriction fragments that result from an enzymatic digestion are separated according to size, they form a specific pattern, which represents a fingerprint of the digested DNA. Changes of this fingerprint indicate that the number or position of the recognition sites of the used restriction enzyme has changed. Restriction enzyme patterns can, therefore, be used to characterize mutational changes or to compare orthologous genes from different organisms.

The size separation of digested DNA is also a prerequisite to isolate and clone specific fragments. If the sequence of the DNA is known, the number and size of the restriction fragments for a given restriction enzyme can be predicted. By choosing the right enzyme from the large number of available restriction enzymes, it is often possible to produce a fragment that contains the gene, or region of DNA, of interest. This fragment can then be separated from the others and cloned into a vector (see Section 14.2) for further investigation.

Electrophoresis is one of the most convenient and most often used methods of molecular genetics to separate molecules that differ in size or charge. Many different forms of electrophoresis exist, but they all work by applying an electrical field to the charged molecules. Since each nucleotide

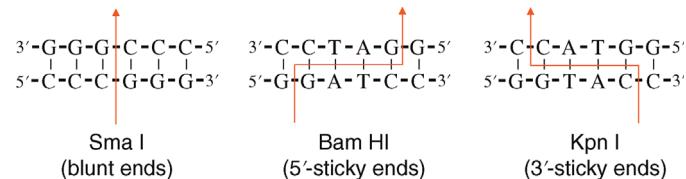


Figure 14.1 Restriction enzymes recognize short stretches of DNA that often have a palindromic structure. The enzyme then cuts the DNA into one of three different ways, producing either blunt ends or sticky ends. This behavior is shown here for the type II enzymes Sma I, Bam HI, and Kpn I. The red line indicates where the enzymes cut the double helix.

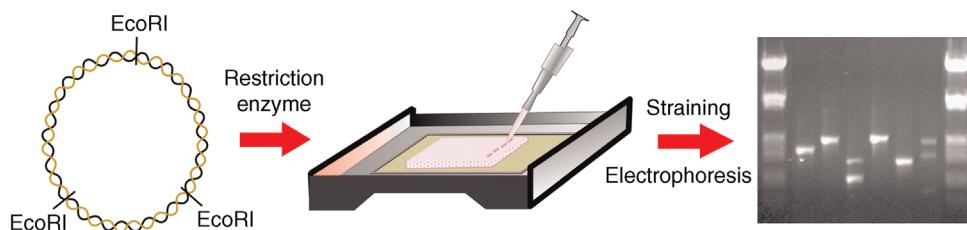


Figure 14.2 Agarose gel electrophoresis of DNA restriction fragments. A plasmid containing several recognition sites for a restriction enzyme (here Eco RI) is digested by the enzyme and the resulting fragments are placed on an agarose gel (middle). An applied electrical field drives the charged molecules through the gel (here from top to bottom) and separates them according to size. After staining, the individual fragments appear under UV light as bright bands (right). (Courtesy of Dr. P. Weingarten, Protagen AG.)

of DNA (or RNA) carries a negative charge, nucleic acids move from the anode to the cathode. The separation is carried out in a gel matrix to prevent convection currents and to present a barrier to the moving molecules, which causes a sieving effect. Size, charge, and shape of the molecules decide how fast they move through the gel. Generally, it holds that the smaller the molecule, the faster it moves. For a typical restriction fragment, that is between 0.5 and 20 kb, agarose gels are used. Agarose is a linear polysaccharide that is extracted from seaweed. For DNA fragments smaller than 500 bp, agarose gels are not suitable. In this case, polyacrylamide gels are used that have smaller pores. For such small molecules size differences of a single base pair can be detected. Another problem represent very large DNA molecules that are completely retarded by the gel matrix. These fragments are not separated by the usual type of electrophoresis. In this case, a special form of electrophoresis, the so-called pulse field electrophoresis, can be used, which allows the separation of DNA molecules of up to 10^7 bp. This technique varies the direction of the electric field periodically, so the molecules follow a zigzag path through the gel. Very long DNA fragments move head-on through the gel, which results in a velocity that is independent of size. Because of the oscillating field, the molecules have to reorient themselves. This is easier for the smaller fragments, so the larger ones lag behind. A typical application of this technique is the separation of whole chromosomes of microorganisms (mammalian chromosomes are even for this technique too large).

Whatever be the type of electrophoresis or gel used, the DNA is invisible unless it is especially labeled or stained. A commonly used dye for staining is ethidium bromide that intercalates between DNA bases. In the intercalated state, ethidium exposed to UV light fluoresces in bright orange.

Figure 14.2 sketches the different processing steps from the DNA to the size-separated restriction fragments on an agarose gel. On the gel (right part Figure 14.2) different lanes can be seen, where different DNA probes are separated in parallel. The concentrated solution of DNA fragments is filled in the pockets at the top of the gel and the fragments migrate during electrophoresis to the bottom. The smallest fragments move fastest and appear at the

bottom of the gel. The lanes on the left and right sides contain fragments of known length that serve as size markers.

14.2 Cloning Vectors and DNA Libraries

In the last section, we discussed how restriction enzymes generate DNA fragments by cutting the DNA at short, specific recognition sites. In this section we will see how the generated fragments can be used to generate billions of identical copies (clones).

For the actual cloning (amplification) step, a restriction fragment has to be inserted into a self-replicating genetic element. This can, for instance, be a virus or a plasmid. Plasmids are small circular rings of DNA that occur naturally in many bacteria. They often carry a few genes for resistance to antibiotics or to enable the degradation of unusual carbon sources and are normally only a few thousand base pairs long (Figure 14.3). Genetic elements that are used in the laboratory to amplify DNA fragments of interest are called cloning vectors and the amplified DNA is said to be cloned. In the following, we will concentrate on the use of plasmids as vectors. The actual insertion process requires that the DNA to be cloned and the vector are being cut with the same restriction enzyme and that the vector has only one recognition site for this enzyme. This gives a linearized plasmid that has the same type of sticky ends as the DNA that is to be cloned. If the linearized vector and the digested DNA are now mixed at the right concentration and temperature, the complementary sticky ends base pair and form a new recombinant DNA molecule. Initially, the resulting molecule is only held together by hydrogen bonds. This is made permanent using the enzyme DNA ligase that forms covalent bonds between the phosphodiester backbones of the DNA molecules. This procedure allows combining DNA from arbitrary sources.

Finally, the vector is introduced into bacterial cells, which are then grown in culture. Every time the bacteria double (approximately every 30 min), the recombinant plasmids also double. Each milliliter of the growth medium

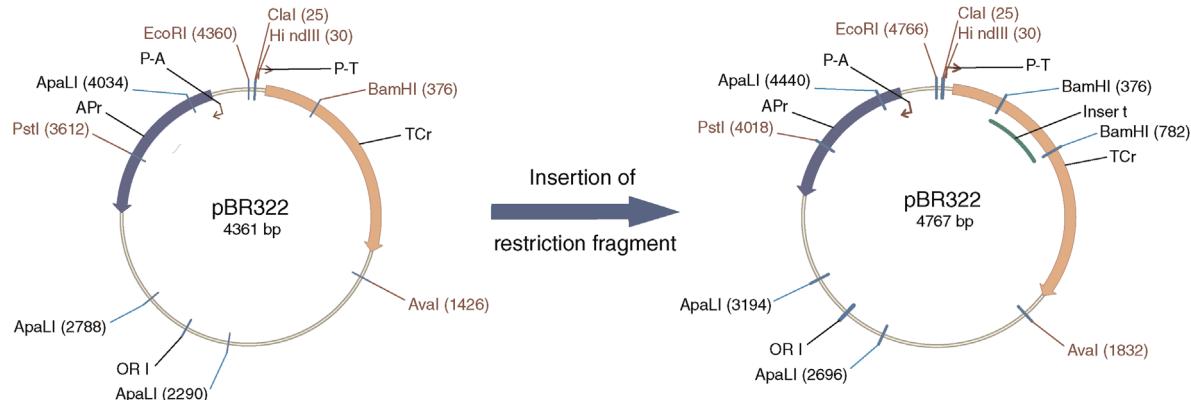


Figure 14.3 pBR322 is a circular plasmid of 4.3 kb that is a classic cloning vector. The diagram on the left shows several important genetic elements of the plasmid. ORI (origin of replication) marks a region of DNA that controls the replication of the plasmid. The boxes represent genes that confer resistance for the antibiotics ampicillin (APr) and tetracycline (TCr). P-A and P-R are the promoters of the resistance genes and the lines and text mark recognition sites for the corresponding restriction enzyme. The right part shows pBR322 after a restriction fragment has been inserted into the Bam HI restriction site.

can finally contain up to 10^7 bacteria! The actual process of introducing the vector into the bacteria is called transformation. For this end, the cells are especially treated so that they are temporarily permeable for the DNA molecules.

But loss occurs at all steps of this genetic engineering. Not all vector molecules will have received an insert, because it is possible that the sticky ends of some vectors self-ligate without insert. Furthermore, not all bacteria used in the transformation step will have received a vector molecule. It can therefore be that there is only a small proportion of cells that contain a vector with insert in the growing cell population. Normally, selection markers are used to cope with this problem. Figure 14.3 shows pBR322, a typical cloning vector. Apart from a DNA sequence that enables the cell machinery to replicate the plasmid (ori), it also contains two genes for resistance against the antibiotics ampicillin and tetracycline. If the DNA fragment is cloned into a restriction site that lies within one of the resistance genes, for instance, the Bam HI site, simple selection steps can be used to end up with cells that contain the desired construct. For this purpose, the bacteria are grown in a medium that contains ampicillin so that only cells that carry the plasmid can survive. The next step is more complicated since we are interested in all those bacteria that contain a vector with a nonfunctional tetracycline gene (caused by an insert). The cells are plated in high dilution on the surface of an agar plate, where each individual cell forms a colony. After the colonies become visible, some bacteria of each colony are copied onto a second agar plate by a stamping technique (which preserves the spatial arrangement of colonies). This second plate contains tetracycline and so only those cells with intact resistance gene can grow. By comparing the colony pattern of the two plates, it is now possible to identify those colonies that exist on the first plate, but not on the second. These are the colonies that we are interested in.

Unfortunately, there is an upper size limit for the DNA one can clone into a plasmid vector. Above 10 kb, the cloning efficiency declines so much that other vectors are required. Lambda is a linear bacteriophage of approximately 48 kb and up to 20 kb of the original phage DNA can be replaced by foreign DNA. Cosmids are artificial constructs that combine some features of the phage lambda and of classical plasmids. The advantage is that fragments up to 45 kb can be cloned. For really large fragments of up to 1 million base pairs, yeast artificial chromosomes (YACs) have been developed [1], which are now gradually replaced by bacterial artificial chromosomes (BACs) [2]. BACs are based on the naturally occurring F-plasmid, which itself is around 100 kb in length. While the copy number per cell of most smaller plasmids is rather large, the F-plasmid and the derived BACs are maintained at only one or two copies per cell. This reduces the risk of unwanted recombination events between different copies and contributes to the stability of such large inserts.

Since the average fragment size of restriction enzymes is much smaller than 1 Mb, the enzymatic reaction is allowed to proceed only for a very short time. This time is not long enough to cut all recognition sites and therefore the resulting fragments are much longer. This technique is called a partial digest.

So far we have discussed the situation where we wanted to clone a specific fragment after a restriction digest. The DNA was separated on an agarose gel, the desired fragment was excised from the gel and cloned into an appropriate vector. However, often the situation is different insofar that we do not know in advance the fragment that contains our gene of interest. In this case, we can construct a DNA library by simply cloning all fragments that result from a digest into vectors. Such a library is maintained in a population of bacteria, each containing vectors with a

different insert. Bacteria with different inserts are either kept together or are separated, so that the library consists of thousands of clones (each kept in a separate plastic tube) each carrying a specific fragment. This is, for instance, important for the construction of DNA chips (see Section 14.7). This strategy is also known as shotgun cloning.

There are two basic types of DNA libraries that are extensively used in molecular genetics. The first type are genomic DNA libraries that were described in the last section. They are directly created from the genetic material of an organism. But restriction enzymes cut DNA irrespectively of the start and end points of genes and hence there is no guarantee that the gene of interest does completely fit on a single clone. Furthermore, in Chapter 13, we have seen that the genome of most higher organisms contains large amounts of junk DNA. These sequences also end up in the genomic DNA library, which is not desired.

A different type of library, the cDNA library, circumvents these problems. This technique does not use DNA as source material, but starts from the mRNA pool of the cells or tissue of interest. The trick is that the mRNA molecules are a copy of exactly those parts of the genome that are normally the most interesting. They represent the coding regions of the genes and contain neither introns nor inter-gene junk DNA. Using the enzyme reverse transcriptase that exists in some viruses, mRNA can be converted into DNA. The resulting DNA is called complementary DNA (cDNA) because it is complementary to the mRNA. cDNA libraries are in several important points different from genomic libraries: (i) They contain only coding regions. (ii) They are tissue specific since they represent a snapshot of the current gene expression pattern. (iii) Because they are an image of the expression pattern, the frequency of specific clones in the library is an indicator of the expression level of the corresponding gene. cDNA libraries have many different applications. By sequencing cDNA libraries, it is possible to experimentally determine the intron–exon boundaries of eukaryotic genes. Constructing cDNA libraries from different tissues helps understand which genes are expressed in which parts of the body. A derivative of the cDNA library are expression libraries. This type of library is constructed in such a way that it contains a strong promoter in front of the cloned cDNAs. This makes it possible not only to amplify the DNA of interest but also to synthesize the protein that is encoded by this DNA insert.

14.3 1D and 2D Protein Gels

The basic principle of electrophoresis works for all charged molecules. This means not only nucleic acids but also other kinds of cellular macromolecules, like proteins,

can be separated by electrophoresis. But the distribution of charges in a typical protein is quite different from the distribution in nucleic acids. DNA molecules carry a negative charge that is proportional to the length of the DNA, since the overall charge is controlled by the phosphodiester backbone. The net charge of proteins, however, varies from protein to protein, since it depends on the amount and type of charged amino acids that are incorporated into the polypeptide chain. If proteins are separated in this native form, their velocity is controlled by a function of charge, size, and shape that is difficult to predict.

It was a major improvement when Shapiro *et al.* [3] introduced the detergent sodium dodecylsulfate (SDS) to protein electrophoresis. SDS has a hydrophilic sulfate group and a hydrophobic part that binds to the hydrophobic backbone of polypeptides. This has important consequences: (i) The negative charge of the protein–detergent complex is now proportional to the protein size because the number of SDS molecules that binds to a protein is proportional to the number of its amino acids. (ii) All proteins denature and adopt a linear conformation. (iii) Even very hydrophobic, normally insoluble, proteins can be separated by gel electrophoresis. Under these conditions, the separation speed of proteins is given by a function of their size, as in the case of nucleic acids.

For proteins, a different gel matrix is used than for nucleic acids. Acrylamide monomers are polymerized to give a polyacrylamide gel. During the polymerization step, the degree of cross-linking and thus the pore size of the network can be controlled to be optimal for the size range of interest. Over the past years, SDS polyacrylamide gel electrophoresis (SDS-PAGE) has become an easy-to-use standard technique for separating proteins by size. As in the case of DNA, the gel has to be stained to make the protein bands visible (e.g., using Coomassie blue, silver staining, or specific antibodies). Figure 14.4a sketches the basic steps required for SDS-PAGE. In this example, the outmost lanes contain protein size markers (large proteins at top, small ones at the bottom) and the middle lanes contain different samples of interest.

On the gel shown in Figure 14.4a, only a few protein bands can be seen. This is typically the result after several protein purification steps. However, a cell or subcellular fraction contains hundreds or thousands of different proteins. If such a mixture is used for SDS-PAGE, individual bands overlap and proteins cannot be separated clearly. A solution to this problem is the two-dimensional polyacrylamide gel electrophoresis [4]. The idea is to separate the proteins in a second dimension according to a property different than size.

Isoelectric focusing (IEF) is such a separation technique. The net charge of a protein depends on the number of charged amino acids, but also on the pH of the medium. At a very low pH, the carboxy groups of aspartate and

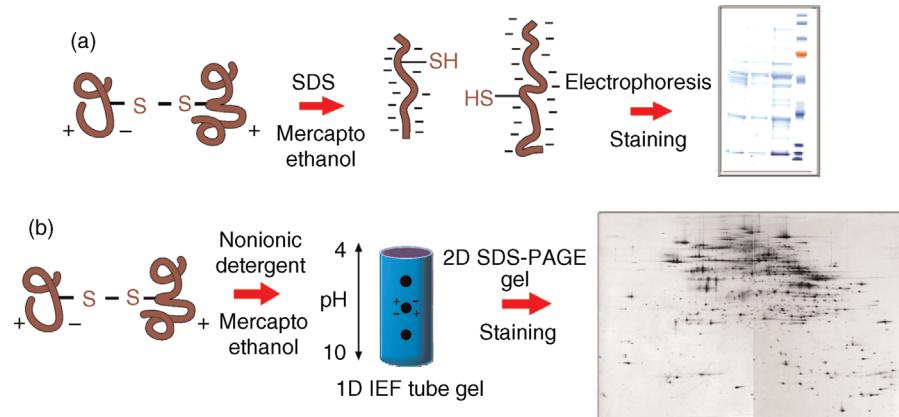


Figure 14.4 (a) SDS-PAGE: Native proteins are treated with the negatively charged detergent SDS and the reducing agent mercaptoethanol to break up disulfide bridges and unfold the protein. After this treatment, even extremely hydrophobic proteins can be separated on a polyacrylamide gel according to their size. (Courtesy of Dr. P. Weingarten, Protagen AG.) (b) 2D gel electrophoresis: For the first dimension, proteins are separated in a rod-like tube gel according to their isoelectric point. To preserve the native charge of the proteins, a nonionic detergent is used to unfold the polypeptides. For the second dimension, the tube gel is placed on top of a standard SDS-PAGE slab gel and the proteins are now separated by size. Up to 2000 proteins can be separated with this technique. (Courtesy of Dr. L. Mao and Prof. J. Klose, Charité Berlin.)

glutamate are uncharged ($-COOH$), while the amino groups of lysine and arginine are fully ionized ($-NH_3^+$), conferring a positive net charge to the protein. At a very basic pH, by contrast, the carboxy groups are charged ($-COO^-$) and the amino groups are neutral ($-NH_2$), resulting in a negative net charge. Consequently, for each protein, there exists a pH that results in an equal amount of negative and positive charges. This is the isoelectric point of the protein where it has no net charge. For isoelectric focusing, the proteins are treated with a nonionic detergent so that the proteins unfold, but retain their native charge distribution (Figure 14.4b). Then they are placed onto a rod-like tube gel, which has been prepared such that it has a pH gradient from one end to the other. After a voltage is applied, the proteins travel until they reach the pH that corresponds to their isoelectric point.

For the second dimension, the tube gel is soaked in SDS and then placed on top of a normal SDS slab gel. A voltage is applied perpendicular to the direction of the first dimension and the proteins are now separated according to size. The result is a two-dimensional (2D) distribution of proteins in the gel, as shown in Figure 14.4b.

DIGE (difference gel electrophoresis) is a variation of this technique that makes 2D gels available for high-throughput experiments [5]. Before the protein samples are separated electrophoretically, they are labeled with fluorescent dyes. Up to three samples are marked with different dyes and the total mixture is then separated on one 2D gel. Three different images are generated from this gel by using different, dye-specific, excitation wavelengths for the scanning process. This approach ensures that identical protein species from the different samples

are located at the same gel position, avoiding the error-prone spot matching step that is necessary when comparing two different 2D gels.

14.4 Hybridization and Blotting Techniques

Hybridization techniques are based on the specific recognition of a probe and target molecule. The aim is to use such techniques to detect and visualize only those molecules in a complex mixture that are of interest to the researcher. The base pairing of complementary single-stranded nucleic acids is the source of specificity for Southern blotting, Northern blotting, and *in situ* hybridization, which are described in the following sections. A short fragment of DNA, the probe, is labeled in such a way that it can later easily be visualized. Originally, radioactive labeling has been used, but in recent years it has often been replaced by fluorescent labels. The probe is incubated with the target sample and after the recognition of probe and target molecules is completed, the location of the probe shows the location and existence of the sought-for target molecule. In principle, 16 nucleotides are sufficient to ensure that the sequence is unique in a typical mammalian genome ($4^{16} \approx 4.29 \times 10^9$), but in practice much longer probes are used. Finally, the Western blot is not a hybridization technique, since it is not based on the formation of double-stranded DNA, RNA, or DNA/RNA hybrids by complementary base pairing. Instead, it is based on the specific interaction between antibody and antigen.

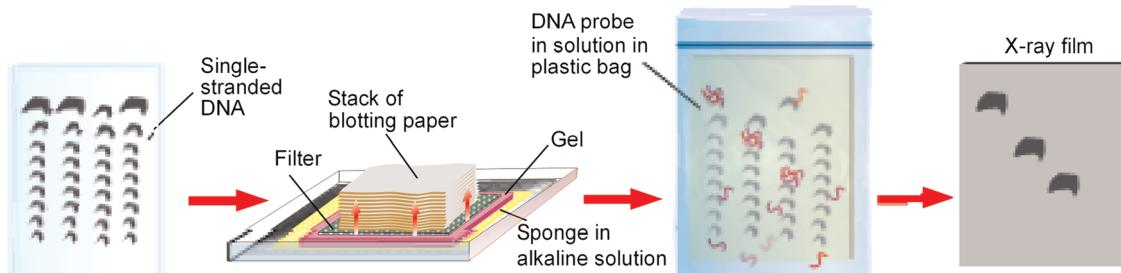


Figure 14.5 Elementary steps required for Southern blotting. Following gel electrophoresis, the DNA fragments are treated with an alkaline solution to make them single stranded. The nitrocellulose or nylon membrane is sandwiched between the gel and a stack of blotting paper and the DNA is transferred onto the membrane through capillary forces. Finally, the membrane is incubated with the labeled DNA probe (here radioactive labeling) and the bands are then visualized by X-ray film exposure. (Courtesy of Dr. P. Weingarten, Protagen AG.)

14.4.1 Southern Blotting

The technique of Southern blotting is used to analyze complex DNA mixtures [6]. Normally, the target DNA is digested by restriction enzymes and separated by gel electrophoresis. If the DNA is of genomic origin, the number of resulting fragments will be so large that no individual bands are visible. If we are interested in a certain gene and know its sequence, small DNA fragments can be synthesized, which are complementary to the gene. However, before the actual hybridization step, the digested DNA has to be transferred from the gel onto the surface of a nitrocellulose or nylon membrane, so that the DNA molecules are accessible for hybridization. This is achieved with the help of a blotting apparatus, as shown in Figure 14.5. Originally, the transfer was achieved by placing the nitrocellulose filter between the gel and a stack of blotting paper. Capillary forces lead to a flow of water and DNA fragments from the gel into the blotting paper. On its way, the DNA gets trapped by a nitrocellulose or a nylon filter. Nowadays more sophisticated blotting machines are used that transfer the DNA by applying a voltage across the gel and membrane. Once the DNA is blotted, the membrane is placed into a plastic bag and incubated for several hours with a solution containing the labeled DNA probe. In case of a radioactive label, the membrane is finally placed against an X-ray film. The radioactive DNA fragments expose the film and form black bands that indicate the location of the target DNA.

With this technique, the presence of the gene of interest can be tested, but also modifications of the gene structure (in case of a mutation) can be studied. By performing several Southern blots with DNA probes that correspond to different regions of the gene, modifications like deletions and insertions can be detected. Point mutations, however, cannot be identified with Southern blotting.

14.4.2 Northern Blotting

Northern blotting is very similar to Southern blotting. The only difference is that mRNA and not DNA is used for blotting. Although the experimental technique is very similar, Northern blotting can be used to answer different questions than Southern blotting. Even though mRNA is only an intermediate product on the way from the gene to the protein, it is normally a reasonable assumption that the amount of mRNA is correlated with the amount of the corresponding protein in the cell. Northern hybridization is therefore used not only to verify the existence of a specific mRNA but also to estimate the amount of the corresponding protein via the amount of mRNA. Since the expression profile of genes varies among tissues, Northern blotting gives different results for different organs in contrast to Southern blotting, which is based on genomic DNA.

14.4.3 Western Blotting

So far we have seen techniques for blotting different types of nucleic acids. The same type of technique exists for proteins, called Western blot. Depending on the problem at hand, 1D or 2D protein gels can be used for blotting. It is more difficult to obtain specific probes for proteins than for nucleic acids. Apart from special cases, antibodies are used that are directed against the desired protein.

Once the proteins are transferred to the nitrocellulose membrane, they are incubated with the primary antibody. In a further step, the membrane is incubated with the so-called secondary antibody, which is an antibody against the primary antibody. If the primary antibody was obtained by immunizing a rabbit, the secondary antibody could be a goat-anti-rabbit antibody. This is an antibody from a goat, which recognizes all rabbit antibodies. The secondary antibody is chemically linked to an enzyme,

like horseradish peroxidase, which catalyzes a chemiluminescence reaction and exposure of an X-ray film finally produces bands, indicating the location of the protein–antibody complex. The intensity of the band is proportional to the amount of protein. The secondary antibody serves as signal amplification step. That is the reason why the enzyme is not linked directly to the primary antibody.

14.4.4

In Situ Hybridization

The described blotting and hybridization techniques are applied to mixtures of nucleic acids or proteins that have been extracted from cells or tissues. During this process, all information about the spatial location is lost. *In situ* hybridization avoids this problem by applying DNA probes directly to cells or tissue slices.

One common application is the location of specific genes on chromosomes. For this purpose, metaphase chromosomes, which have been exposed to a high pH to separate the double strands, are incubated with labeled DNA probes. This makes it possible to directly see where and how many copies of the gene are located on the chromosome. If the label is a fluorescent dye, the technique is called FISH (fluorescent *in situ* hybridization). Not only chromosomes, but also slices of whole tissues and organisms can be hybridized to DNA probes. This can be used to study the spatial and temporal expression patterns of genes by using probes specific to certain mRNAs. This method is often used to study gene expression patterns during embryogenesis. Immunostaining, finally, uses antibodies to localize proteins within cells. Knowledge about the subcellular localization often helps us to better understand the functioning or lack of functioning of the studied proteins.

14.5

Further Protein Separation Techniques

14.5.1

Centrifugation

One of the oldest techniques for the separation of cell components is centrifugation. This technique fractionates molecules (and larger objects) according to a combination of size and shape. However, in general, it holds that the larger the object, the faster it moves to the bottom. A typical low-speed centrifugation collects cell fragments and nuclei in the pellet; at medium speeds, cell organelles and ribosomes are collected and at ultrahigh speeds, even typical enzymes end up in the pellet. The sedimentation rate for macromolecules is measured in Svedberg units, S , after Theodor Svedberg who invented ultracentrifugation in 1925. S is defined by the ratio of the sedimentation

velocity (v) and the centrifugal acceleration ($\omega^2 r$). The sedimentation velocity itself is controlled by the mass (m) and density of the particle (ρ_{par}), as well as the density (ρ_{sol}) and friction (f) of the medium:

$$S = \frac{v}{\omega^2 r}, \quad v = \frac{m(1 - \rho_{\text{sol}}/\rho_{\text{par}})}{f}.$$

The ribosomal subunits, for instance, got their name from their sedimentation coefficient (40S and 60S subunits). Because the friction is controlled not only by the size of the particle but also by its shape, S values are not additive. The complete ribosome (40S plus 60S) sediments at 80S and not at 100S. From the above expressions it follows that the sedimentation rate is zero, if the density of the particle and the surrounding medium are identical. This is the basis for the equilibrium centrifugation method in which the medium forms a stable density gradient caused by the gravitational forces. If the centrifugation is run long enough, the particles move to the position where the density of the medium and the particle are identical and form stable bands there. Thus, equilibrium centrifugation separates the molecules by density, independent of their size.

14.5.2

Column Chromatography

Other classical separation techniques include the different forms of column chromatography (Figure 14.6). A column is filled with a solid carrier material and the protein mixture is placed on top of it. Then buffer is slowly washed through the column and takes along the protein mixture. Different proteins are held back to a different degree by the column material and arrive after different times at the bottom of the column. The eluate is fractionated and tested for the presence of the desired protein. The ratio of desired protein to total protein is a measure of purity. Different column materials are available and the success of a separation often depends critically on the choice of the appropriate material.

The material for ion exchange chromatography (Figure 14.6a) contains negatively or positively charged beads that can be used to separate hydrophilic proteins according to the charge. The binding between the proteins and the beads is also influenced by the salt concentration and pH of the elution buffer. Some proteins possess hydrophobic surfaces that can be used to separate proteins by hydrophobic interaction chromatography (Figure 14.6b). For this purpose, short aliphatic side chains are attached to the surface of the column material. Gel filtration chromatography (Figure 14.6c) is also often used to separate proteins according to the size. The beads contain a range of pores and channels that allow small molecules to enter, which increases their retention times. This allows not only their separation by size but also estimation of the absolute size of a protein or a protein complex. A more recent

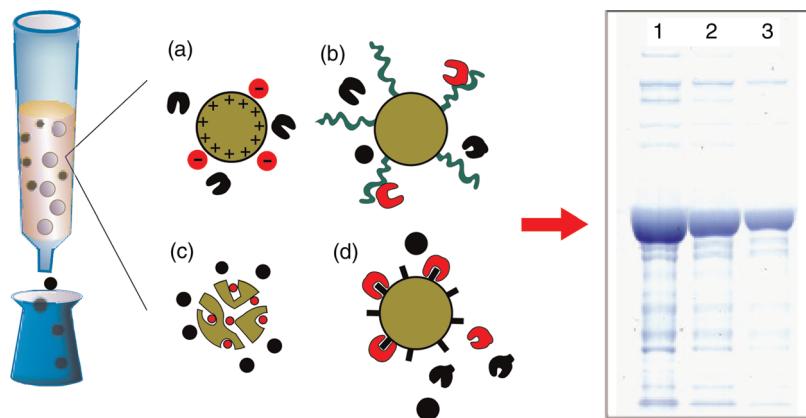


Figure 14.6 In column chromatography, a protein mixture is placed on top of the column material that is then eluted with buffer. Different types of material are available that separate the proteins according to charge (a), hydrophobicity (b), size (c), or affinity to a specific target molecule (d). Often different chromatographic steps have to be used successively to purify the desired protein to homogeneity (lanes 1–3 of the gel). (Courtesy of Dr. P. Weingarten, Protagen AG.)

development is affinity chromatography (Figure 14.6d) that makes use of highly specific interactions between a protein and the column material. This can, for instance, be achieved by chemically linking antibodies to the column material. The proteins of interest bind, while the other proteins pass through the column. In a second step, the elution process is started by using a high salt or pH buffer that recovers the bound protein from the column.

A major improvement regarding speed and separating power was achieved through the development of high-performance liquid chromatography (HPLC). The columns are much smaller and the carrier material is packed more densely and homogeneously. To achieve reasonable buffer flow rates, very high pressures (up to several hundred atmospheres) are needed that are generated using special pumps.

The enrichment factor of a single chromatographic step is normally between 10- and 20-folds. However, since many proteins represent only a tiny fraction of the total protein content of a cell, often different chromatographic columns have to be used consecutively. A notable exception is affinity chromatography, which can achieve enrichments up to 10^4 in a single step. In combination with modern recombinant DNA techniques, affinity chromatography has many applications. Recombinant proteins can be designed to contain special short sequences of amino acids that do not compromise the functionality of the protein but can serve as molecular tags. A short stretch of histidine residues, for instance, is called a His tag and is specifically recognized by a nickel surface or special His antibodies.

directly from genomic DNA [7]. A pair of short oligonucleotides (15–25 bp), the primers, are synthesized chemically such that they are complementary to an area upstream and downstream of the DNA of interest. DNA is made single stranded by heating, and during the cooling phase, primers are added to the mixture, which then hybridize to the single-stranded DNA. In a next step, a DNA polymerase extends the primers, doubling the copy number of the desired DNA fragment. This concludes one PCR cycle. Each additional cycle (denaturation, annealing, and amplification) doubles the existing amount of the DNA that is located between the primer pair. Thus, 30 cycles correspond to a $2^{30} = \sim 10^9$ -fold amplification step (25–35 cycles are typically used). Today, the different steps of the PCR reaction do not have to be performed manually. Small, automated PCR machines, also called thermal cyclers, can perform dozens of PCR reactions in parallel and a single cycle is finished in 3–4 min.

In the last years, sequence information for many complete genomes has become available, which allows using PCR to clone genes directly from genomic DNA without the use of DNA libraries. PCR has revolutionized modern molecular genetics and has many applications. For instance, by combining reverse transcriptase (which makes a DNA copy from RNA) with PCR, it is also possible to clone mRNA with this technique. Furthermore, the extreme sensitivity of PCR makes it the method of choice for forensic studies. Highly variable tandem repeats are amplified and used to test if genetic material that comes from hair follicle cells, saliva, or blood stains belongs to a certain suspect. This is possible because different individuals have tandem repeats of different lengths, which results in amplified DNA fragments of different lengths. By looking at a large number of loci that contain tandem repeats, the chances of a false positive result can be made arbitrarily small. This principle is also the basis of paternity tests.

14.6 Polymerase Chain Reaction

The polymerase chain reaction allows billion-fold amplification of specific DNA fragments (typically up to 10 kbp)

An important, more recent, development is real-time PCR that is especially suited to quantify the amount of template that was initially present. Classical PCR is normally unable to give quantitative results, because of saturation problems during the later cycles. The real-time PCR circumvents these problems by using fluorescent dyes that either intercalate in double-stranded DNA or are bound to sequence-specific oligonucleotides (TaqMan® probe). The increase of fluorescence with time is used, in real time, as an indicator of product generation during each PCR cycle.

14.7 Next-Generation Sequencing

In the 1970s, the first techniques have been developed that allowed the sequencing of short stretches of DNA [8,9]. Continuous improvement and automation of the Sanger method [9] finally allowed the determination of the complete sequence of the human genome [10]. However, the growing demand for ever-increasing sequencing capacities led to the development of cheaper and faster methods, the so-called next-generation sequencing (NGS) techniques. The field is still under active development with the constant development of new methods [11,12]. We will therefore restrict ourselves

here to the description of the two most popular NGS techniques, the pyrosequencing method of Roche/454 and the solid-phase method of Illumina/Solexa.

Both methods start with the preparation of single-stranded DNA molecules, which is normally achieved by random shearing of the source DNA into fragments of a suitable length (approximately 50–500 bp). After separation into single strands, 5' and 3' specific primers are added that are required for the following amplification and sequencing steps as well as for the anchoring to some solid surface. The Roche/454 method uses an emulsion PCR (emPCR) system for amplification of single DNA molecules (Figure 14.7a, center). First, molecules are covalently attached to beads under such conditions that only a single DNA molecule per bead is attached. Next millions of such beads are emulsified with all reagents required for PCR in a water–oil mixture. This effectively creates separate microreactors in which multiple copies of each library fragment are produced. The reaction mixture contains only one free primer, while the other primer is covalently attached to the bead. After the amplification process, this results in beads that are covered with a clonal population of covalently attached copies of a single fragment. Finally, the emulsion is broken and the beads are separated into tiny (picoliter range) wells, again under such conditions that only a single bead fits into a well (Figure 14.7a, right). A solution with a

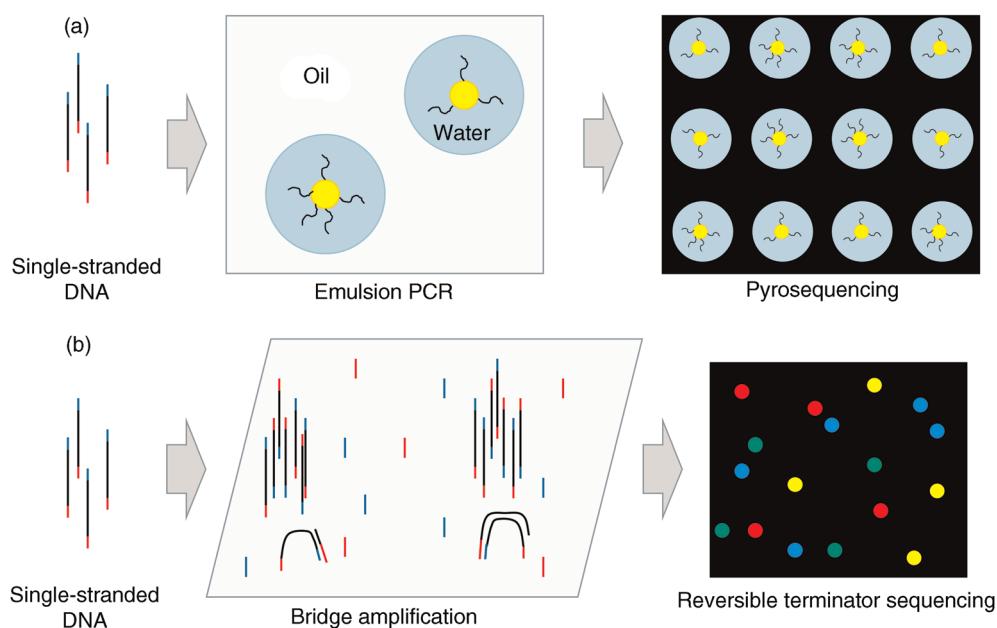


Figure 14.7 Schematic representation of the two most popular next-generation sequencing (NGS) methods. (a) The pyrosequencing method by Roche/454 starts with single-stranded DNA molecules that are amplified on beads in an emulsion PCR system. Sequencing of individual beads then takes place inside wells etched into a fiber-optic slide. (b) The solid-phase method of Illumina/Solexa also starts with single-stranded DNA molecules, but these are then amplified on a solid-phase surface, leading to spatially separated clusters of amplified DNA. Sequencing is performed using fluorescent reversible terminators and an optical analysis of the surface. For details, see the text.

single type of nucleotide is then flowed across the wells and in those wells where the nucleotide is complementary to the template strand, it is incorporated and results in a chemiluminescent light signal recorded by a camera. For the next cycle, the free nucleotides are washed out and another type of nucleotide is added. This technique allows read length of up to 1000 nucleotides and a maximum throughput of up to 0.7 Gb per run [12].

The solid-phase method of Illumina/Solexa also starts with a single-stranded DNA, but the amplification process then takes place on a two-dimensional surface that is covered with covalently attached primers of both types (Figure 14.7b, center). A diluted solution of ssDNA fragments is washed over the surface, which then attach to the primers. At the places of attachment, a local PCR reaction is then performed, called bridge amplification. The DNA strands bend over to hybridize with complementary primers, providing the starting point for the next round of PCR. This results in spatially separated clusters of amplified DNA that grow until the PCR reaction is stopped. Sequencing of these clusters is then performed using reversible terminator nucleotides. These nucleotides contain a fluorophore and prevent the incorporation of any further nucleotide. Therefore, the solution can contain a mixture of all four (differently labeled) terminator nucleotides and in each cluster, only the next nucleotide complementary to the DNA sequence is incorporated. A picture is taken (Figure 14.7b, right) and the colored spots show which nucleotide has been incorporated by a given cluster. The next round of sequencing is started by cleaving and washing out the terminator (i.e. fluorophore) part of the nucleotide, which renders the cluster susceptible to the next terminator nucleotide. In contrast to the Roche/454 method, this technique is also robust against stretches of identical nucleotides, since only a single nucleotide can be incorporated at a time. Read lengths up to 300 nucleotides and a maximum throughput of up to 1800 Gb per run can be achieved. With their newest machine, the HiSeq X Ten, Illumina claims to have reached the \$1000 price barrier for sequencing a human genome [12].

14.8 DNA and Protein Chips

14.8.1 DNA Chips

DNA chips, also called DNA microarrays, are a high-throughput method for the analysis of gene expression [13]. Instead of looking at the expression of a single gene, microarrays allow monitoring the expression of

several thousand genes in a single experiment, resulting in a global picture of the cellular activity.

A microarray experiment starts with the construction of the chip from a DNA library (Section 14.2). The inserts of individual clones are amplified by PCR (a single primer pair can be used, which is specific for the vector that was used to construct the library) and spotted in a regular pattern on a glass slide or nylon membrane. These steps are normally automated and performed by robots. Then total mRNA is extracted from two samples that should be compared (e.g., yeast cells before and after osmotic shock). Using reverse transcriptase, the mRNA is transcribed into cDNA and labeled with a fluorescent dye. It is important that the dyes used for the two samples emit light at different wavelengths. Red and green dyes are commonly used. The cDNAs are now incubated with the chip where they hybridize to the spot that contains the complementary DNA fragment. After washing, the ratio of the fluorescence intensities for red and green are measured and displayed as false color picture. Spots of pure red or green indicate a large excess of mRNA from one or the other sample, while yellow spots show that the amount of this specific mRNA was roughly equal in both samples. Very low amounts of both mRNA samples result in dark spots. These ratios can be quantified numerically and used for further analyses, like the generation of a clustergram. For this purpose, a complete linkage cluster of the genes that were spotted on the chip is generated and the mRNA ratio (represented as color) is displayed in this order. This helps to test if groups of related genes (maybe all involved in fatty acid synthesis) show a similar expression pattern.

A variant of DNA chips, oligonucleotide chips are based on an alternative experimental design. Instead of spotting cDNAs, short oligonucleotides (25–50 mer) are used. Approximately a dozen different and specific oligonucleotides are used per gene. In this case only one probe of mRNA is hybridized per chip and the ratio of fluorescence intensity of different chips is used to estimate the relative abundance of each mRNA. Most commonly, chips from companies like Affymetrix or Agilent are used for this approach.

In the last years, this technique was used to study such diverse problems as the effects of caloric restriction and aging in mice [14], influence of environmental changes on yeast [15], or the consequences of serum withdrawal on human fibroblasts [16].

14.8.2 Protein Chips

Despite the large success of DNA chips, it is clear that the function of genes is realized through proteins and not by

mRNAs. Therefore, efforts are under way to construct chips that consist of spotted proteins instead of DNA. In this case, the starting point is an expression library for obtaining large quantities of the recombinant proteins. The proteins are spotted and fixed on a coated glass slide that can then be incubated with interaction partners. This could be (i) other proteins to study protein complexes, (ii) antibodies to quantify the spotted proteins or to identify the antigen that is recognized by the antibody, (iii) DNA to find DNA binding proteins, or (iv) drugs to identify compounds of pharmaceutical interest [17].

However, the generation of protein chips poses more problems than DNA chips because proteins are not as uniform as DNA. One challenge is to express sufficient amounts of recombinant proteins in a high-throughput approach. Another problem is that the optimal conditions (temperature, ionic strength, etc.) for the interaction with the reaction partner are not identical for different proteins. But academic groups and companies are constantly improving the technique and chips with several thousand different proteins are now used successfully [18].

The main advantage of DNA and protein chips is the huge amount of data that can be gathered in a single experiment. However, this is also a feature that needs careful consideration. The quality of an expression profile analysis based on array data is highly dependent on the number of repeated sample measurements, the array preparation, hybridization, and signal quantification procedure [19]. The large number of samples on the chip also poses a problem regarding multiple testing. If care is not taken, a large number of false positives are to be expected.

14.9 RNA-Seq

The DNA chip technology that was discussed in Section 14.8.1 has been a valuable method for quantifying gene expression levels of individual transcripts in a high-throughput fashion. However, the techniques have several limitations that restrict its usefulness and some of which are also relevant from a modeling viewpoint. First of all, the genomic sequence has to be known for the construction of a DNA chip. Although the number of sequenced species is constantly increasing, it nevertheless restricts the analysis to popular model species. However, the most important limitation of DNA chips is probably its small dynamic range of only a few hundred-fold [20]. This stems from the fact that rare transcripts cannot be detected because of a relatively high fluorescence background (caused by material and/or cross-hybridizations)

and highly abundant transcripts cannot be accurately quantified because of saturation effects.

The advent of next-generation sequencing techniques (Section 14.7) has provided the ground for a new high-throughput quantitative technique for transcriptome profiling called RNA-Seq (RNA-sequencing) [20,21]. The basic idea is quite simple; after the RNA isolation (e.g., total RNA, mRNA, and small RNAs), the molecules are converted to cDNA and then, with or without amplification, sequenced in a high-throughput fashion. The reads are typically between 30 and 400 bp, depending on the used sequencing method. Since large mRNAs have to be broken into smaller fragments to be compatible with most current sequencing methods, the number of counts per transcript is proportional to the expression level and the length of the transcript. To compare expression levels of different transcripts and between libraries with different sequencing depth, the expression level is normally expressed as number of reads per fragment per kilobase per million reads (RPKM). Since RNA-Seq has practically no background, its dynamic range is in principle only limited by the sequencing depth. A dynamic range covering five orders of magnitude was estimated for 40 million sequence reads in mouse [22]. Other advantages of RNA-Seq are that the genomic sequence of the studied organism is not necessary to be known, that it is suitable to differentiate between highly similar mRNA isoforms, and can reveal the exact location of the transcription boundaries. Although RNA-Seq does not suffer from the problems of the hybridization-based approaches, one should keep in mind that it is also subject to some biases originating from the cDNA preparation step. One problem is that all techniques used to generate smaller mRNA fragments, such as sonication, enzymatic digestion, or the use of divalent cations, produce their own specific bias. And another problem is that the random hexamers used as PCR primers do not anneal randomly [21]. But despite these issues, RNA-Seq is a technique with many advantages that will replace DNA-chip-based approaches in many areas of research.

14.10 Yeast Two-Hybrid System

The yeast two-hybrid (Y2H) system is a technique for the high-throughput detection of protein–protein interactions. It rests on the fact that some transcription factors, such as the yeast *Gal4* gene, have a modular design, with the DNA binding domain separated from the activating domain. To test if two proteins, called bait and prey, interact, their genes are fused to the DNA binding or DNA activating domain, respectively, of the TF

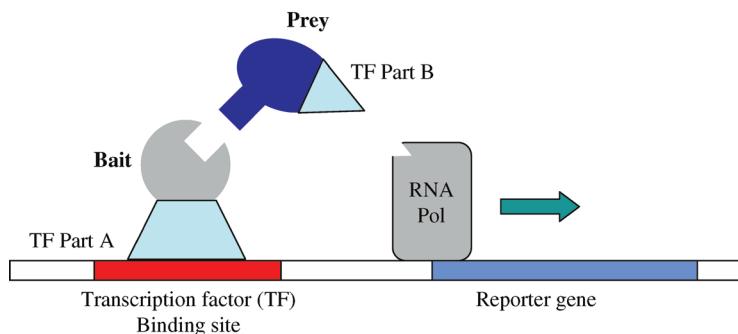


Figure 14.8 The yeast two-hybrid system identifies protein–protein interactions. The genes of the bait and prey proteins are fused to parts of a yeast transcription factor. If bait and prey interact, the different parts of the transcription factor come close enough to activate the expression of a reporter gene.

(Figure 14.8). The bait binds to the DNA via its TF fragment. If bait and prey do interact, the two domains of the transcription factor come close enough to stimulate the expression of a reporter gene. If bait and prey do not interact, the reporter gene is silent. This technique can be used to find all interacting partners of the bait protein. For this purpose, yeast cells are transformed with an expression library containing prey proteins fused to the activating part of the TF. Although the detection occurs in yeast, the bait and prey proteins can come from any organism. This single-bait multiple prey system can be extended even to a multiple bait multiple prey approach, which made it possible to obtain the complete protein interactome of yeast [23]. However, as with most high-throughput techniques, the two-hybrid system is prone to false positives, as indicated by the fact that a second study using the same technique derived a quite dissimilar yeast interactome [24]. To corroborate the interactions obtained with Y2H, affinity chromatography or coimmunoprecipitation can be used.

14.11 Mass Spectrometry

The identification and quantification of individual proteins of a cell is an essential part for studying biological processes. The first step toward identification often involves a separation of the protein broth. Two-dimensional gels (Section 14.3) or the different forms of chromatography (Section 14.5.2) are frequently used for this task. The separated proteins can then be identified by cleaving them enzymatically into specific peptides and determining the exact size and sequence of these fragments using different types of mass spectrometry.

MS has been used for the past 50 years to measure the masses of small molecules with high accuracy. But its

application to large biomolecules has been limited by the fragility and low volatility of these materials. However, the situation changed in the late 1980s with the emergence of the matrix-assisted laser desorption/ionization mass spectrometry (MALDI-MS) [25] technique and the electrospray ionization (ESI) [26]. For MALDI, the polypeptide is mixed with solvent and an excess of low molecular weight matrix material. Polypeptide and matrix molecules cocrystallize and are placed in the spectrometer under vacuum conditions. The probe is then targeted with a laser beam that transfers most of its energy to the matrix material, which heats up and is vaporized. During this process, the intact polypeptide is charged and carried into the vapor phase. Electrospray ionization is, in contrast to MALDI, a continuous method that can be used together with chromatographic separation techniques, such as HPLC. In this case, a spray of small droplets is generated that contains solvent and charged macromolecules. Evaporation of the solvent leads to individual, charged protein molecules in the gas phase.

For both methods, the following steps are very similar. An electrical field accelerates the molecules and the time of flight (TOF) for a specific distance is measured, which depends on the ratio of mass and charge ($\text{TOF} \sim \sqrt{m/z}$). The mass accuracy depends strongly on the used MS technique and type of probe, but accuracies around 1 ppm (part per million) are often achievable. In addition to measuring peptide masses, MS can also be used for obtaining sequence information. Specific peptide ions are selected and subjected to further fragmentation through collision with gas molecules. The sequence information obtained by this tandem mass spectrometry (MS/MS) is then used to identify the protein using powerful search engines such as Mascot (www.matrixscience.com), Sequest (fields.scripps.edu/sequest), Phenyx (<http://www.genebio.com/products/phenyx>), or the free X! Tandem engine (<http://www.thegpm.org/TANDEM>). Many

different types of MS machines and techniques exist for specialized applications and a good review is given by Domon & Aebersold [27].

14.12 Transgenic Animals

Genetic material can be introduced into single-cell organisms (bacteria and yeast) by transformation (Section 14.2) and is then automatically passed down to the offspring during each cell division. But to achieve the same in a multicellular organism is much more complicated. One obvious problem is the separation of soma and germline. To generate animals that carry the transgene in all of their cells, it has to be placed in germline cells (sperm and eggs), which then form all the cells of the new transgenic animal. A second problem is the way that the transgene is inserted into the genome. Ideally, one would like to insert a single copy at a specific place in the genome, because otherwise undesired side effects can occur. Uncontrolled insertion can, for instance, disturb the activity of endogenous genes and properties of the chromatin structure at the insertion point can also influence the expression of the transgene in an unforeseeable way.

14.12.1 Microinjection and ES Cells

The first method that was applied successfully to mammals is DNA microinjection [28]. It is based upon the observation that in mammalian cells, linear DNA fragments are rapidly assembled into tandem repeats, which are then integrated into the genomic DNA. This integration occurs at a single random location within the genome. For this method, a linearized gene construct is injected directly into the pronucleus of a fertilized ovum. After introduction into foster mothers, embryos develop that contain the foreign DNA in some cells of the organism, resulting in a chimera. If the construct is also present in germline cells, some animals of the daughter generation (F_1 generation) will carry the transgene in all of its body cells. A transgenic animal has been created. The advantage of DNA microinjection is that it is applicable to a wide range of species. The disadvantage is that the integration is a random process and so the genomic neighborhood of the insert is unpredictable. This often means that the expression of the recombinant DNA is suppressed by silencers or by an unfavorable chromatin structure. As mentioned, the insertion process can also alter the expression of existing genes if the insert destroys the coding or control region of existing genes. Finally, if a mutant form of an endogenous gene has been introduced,

it has to be considered that, in general, the wild type is also present, which restricts this approach to the investigation of dominant mutants.

This problem can be overcome by using the method of embryonic stem-cell-mediated transfer [29]. In rare cases (approximately 0.1%), the integration of a gene variant into the genome does not occur randomly, but actually replaces the original gene via homologous recombination. This paves the way to modify or inactivate any gene. In the latter case, this results in knockout animals. For this technique, the gene construct is introduced into embryonic stem cells (ES cells), which are omnipotent and can give rise to any cell type. With the help of PCR or Southern blotting, the few ES cells that underwent homologous recombination can be identified. Some of these cells are then injected into an early embryo at the blastocyst stage, which leads to chimeric animals that are composed of wild-type cells and cells derived from the manipulated ES cells. As in the case of DNA microinjection, an F_1 generation of animals has to be bred to obtain genetically homogeneous animals. ES-cell-mediated transfer works particularly well in mice and is the method of choice to generate knockout mice, which are invaluable in deciphering the function of unknown genes.

14.12.2 Genome Editing Using ZFN, TALENs, and CRISPR

In recent years, several new techniques have been developed that make it possible to insert the transgene exactly at the desired location. At the heart of all these techniques is the ability to generate a double-strand break (DSB) at a specific position within the approximately 3 billion base pairs of a mammalian genome.

Zinc finger nucleases (ZFN) are engineered proteins that contain several zinc finger domains, each of which binds specifically to three consecutive bases of DNA [30]. A zinc finger consists of 30 amino acids and several of such domains can be combined in a modular way in a single ZFN. Zinc finger libraries have been created with specificities for almost all of the 64 possible nucleotide triplets, making it possible to custom design zinc finger nucleases that can bind to virtually any sequence in a eukaryotic genome [31]. To complete the design of ZFN, the DNA binding fragment is combined with a nuclease like *Fok1*. Deliberately a nuclease has been chosen that is active only as dimer. For the actual experiment, two different zinc finger nucleases are then designed and used, one that binds the DNA just upstream of the cutting position and another that is specific for the DNA just downstream of it. This allows the *Fok1* domains to form a dimer at the cutting position and generate a double-strand break. The advantage of using two different ZFNs

for a single experiment is to further increase the overall specificity. After the double-strand break has been created, two types of cellular maintenance mechanisms can repair the damage: the fast but unspecific non-homologous end joining (NHE) mediated repair mechanism and homologous DNA recombination. NHEJ repairs the break, but often inserts frameshift mutations, so that the affected gene is no longer functional. This can be used to generate knockout mutants. Homologous recombination, in contrast, repairs the break by inserting DNA sequences if they contain homologous stretches of DNA. Normally, this process is very inefficient in mammals (see also Section 14.12.1), but it was discovered that double-strand breaks strongly increase the activity of this repair mechanism [31]. Thus, codelivering a site-specific nuclease together with a transgene that carries homologies to the target sequence has revolutionized mammalian genome editing.

A recent development that provides an alternative to ZFNs is based on the discovery of DNA binding proteins called TALE (transcription activator like effector) from the plant pathogenic bacteria *Xanthomonas* [32]. These proteins consist of repeats of a domain that is 33–35 amino acids long, with each repeat recognizing a single base pair. The specificity is conferred by just two variable amino acids of the repeat. By combining multiple of these repeat modules, virtually any desired DNA sequence can be targeted [31]. As in the case of ZFNs, these DNA binding proteins are fused to the *FokI* nuclease to form the final tool called TALENs (transcription activator like effector nucleases). Although the single base pair specificity provided by the TALE domains has simplified the design of a target-specific nuclease, it still requires the genetic engineering of the required protein. A very recent technique called CRISPR/Cas has the potential to further simplify this process [33]. Bacteria possess a defense system against foreign DNA that is based on clustered regular interspaced palindromic repeats (CRISPR) in their genome. In these areas, bacteria possess short stretches of DNA that originate from bacteriophages. The CRISPR area is transcribed into RNA and processed to form short RNA molecules, called crRNA, which bind to a nuclease called *Cas9* and act as guide RNA, providing specificity to the nuclease. The target area can be up to 20 base pairs long, sufficient to provide single cut specificity within a mammalian genome.

As discussed earlier, traditional methods to generate transgenic animals produce in a first step chimeric animals and only in the daughter generation pure transgene animals can be obtained. This is routine for mice, but can be a problem if the generation time is much longer or no embryonic stem cells are available for that species. The high efficiency and specificity of new techniques such as

CRISPR/Cas make possible the injection of the transgene plus mRNA for *Cas9* directly into the fertilized egg, resulting directly in complete transgenic animals, without the need for further crossings [34].

14.13 RNA Interference

We have seen that the generation of transgenic animals and the use of homologous recombination to produce knockout animals is one way to elucidate the function of a gene. However, this approach is time consuming, technically demanding, and expensive. A new convenient method for transiently downregulating arbitrary genes makes use of the phenomenon of RNA interference (RNAi). In 1998, it was discovered that the injection of double-stranded RNA (dsRNA) into the nematode *Ceenorhabditis elegans* led to a sequence-specific downregulation of gene expression [35]. It turned out that this effect is part of a natural defense system of eukaryotes against viruses and selfish genetic elements that propagate via long double-stranded RNA intermediates.

The dsRNA is recognized by a cellular endoribonuclease of the RNase III family, called DICER (Figure 14.9a). This cuts the dsRNA into short pieces of 21–23 bp length with a 2 bp single-stranded overhang at the 3'-end. These short fragments are called small interfering RNAs (siRNAs). After phosphorylation, they are assembled (as single strands) into a riboprotein complex called RISC (RNA-induced silencing complex) (Figure 14.9b). If RISC encounters an mRNA complementary to its siRNA, the mRNA is enzymatically cut at a position that corresponds to the middle of the siRNA (Figure 14.9c). In mammals, long dsRNA also induces the interferon response that causes unspecific degradation of mRNA and a general shutdown of translation. This would severely hamper the use of RNA interference as research tool, but luckily artificial siRNAs can also be used to activate the RNA interference machinery [36].

Artificial siRNAs are synthesized chemically and can, in principle, be targeted against any mRNA. However, the search for design rules that would yield the most effective and specific siRNAs is still on [37] and positional effects can lead to suppression levels that vary between 10 and 70% [38]. Because the transfection of siRNA is only transient (it only lasts a few days in mammals), there might also be problems to silence genes that encode for long-lived proteins.

If siRNAs are expressed from plasmids or viral vectors, their effects are more persistent and they can be used as therapeutic agents by downregulating specific disease genes. Applications to HIV and other targets have been

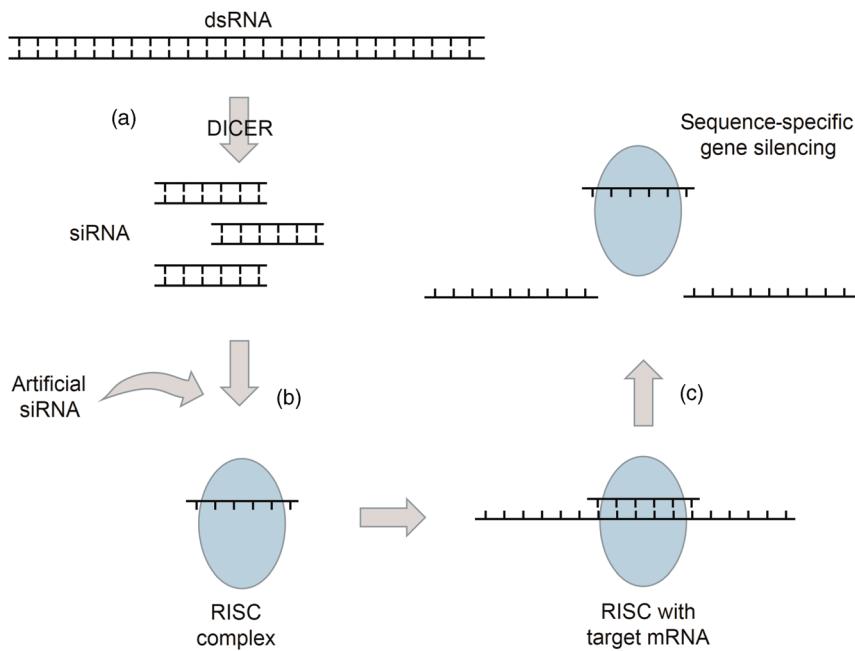


Figure 14.9 Mechanism of RNA interference (RNAi). (a) Double-stranded RNA (dsRNA) is cleaved by the endoribonuclease DICER into small fragments of 21–23 nucleotides, which are subsequently phosphorylated at the 5'-end. To become functional, these small interfering RNAs (siRNAs) form an RNA-induced silencing complex (RISC) with cellular proteins (b). If the RISC complex encounters an mRNA that is complementary to the siRNA, this mRNA is cleaved in the middle of the complementary sequence, leading to gene silencing (c). Exogenously added siRNAs are also functional in triggering RNAi.

discussed [36,37,39]. They can also be used to study regulatory interactions among proteins by down-regulating individual components and then measuring the effects on the global gene expression profile. Following this approach, it was possible to study the structure of a signaling pathway involved in the immune response of *Drosophila melanogaster* [40]. Finally, RNA interference could also be very useful for the model building of metabolic or gene networks. After a network has been formulated by a set of equations, it can be tested by silencing genes individually or in combination and then measuring the resulting new expression levels. This type of network perturbation can be used to iteratively improve the agreement of the model predictions with the experimental data.

Closely related to siRNAs are also the so-called microRNAs (miRNAs). They are of similar size and also use the DICER pathway for maturation [41,42]. The difference is that genes coding for miRNAs are a natural part of eukaryotic genomes and represent a further mechanism used by the cell to regulate and fine-tune protein synthesis.

14.14 ChIP-on-Chip and ChIP-PET

We have already discussed several techniques that produce various types of high-throughput data, such as

protein–protein interactions (yeast two-hybrid) or gene expression levels (DNA chips, RNA-Seq). During recent years, a large number of whole eukaryotic genomes have been sequenced, which makes other types of information increasingly interesting. One example is the recognition site that exists for DNA binding proteins. Of particular interest are the transcription factors that are of crucial importance for gene regulation (but the same strategy can also be applied to all other types of DNA binding proteins). TF binding sites are notoriously difficult to determine using computational approaches since they often represent degenerate motifs of 5–10 nucleotides that appear far too often in genomic DNA as to be specific. It seems that additional factors and specific conditions that exist *in vivo* provide the required additional specificity.

Chromatin immunoprecipitation (ChIP) is an experimental technique that can be used to identify TF binding sites under *in vivo* conditions. In a first step, the TF of interest is cross-linked with the DNA it binds to using formaldehyde fixation. Then the cells are lysed and the DNA is broken down mechanically (sonication) or enzymatically (nucleases), leading to double-stranded pieces of DNA around 1 kb in length. The next step is the purification of the TF–DNA complexes using immunoprecipitation. For this, either an antibody specific for the studied TF is necessary or an antibody against a tag (His

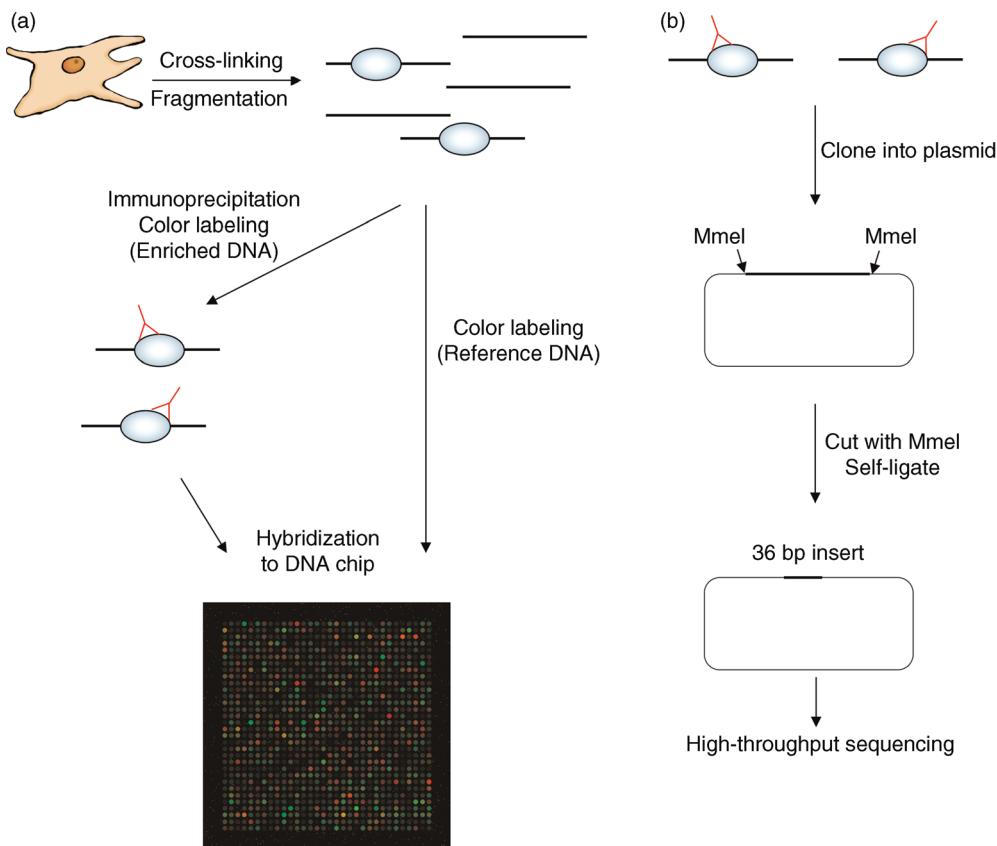


Figure 14.10 (a) ChIP-on-Chip is a high-throughput technique that provides information about DNA binding sites of proteins under *in vivo* conditions. After formaldehyde cross-linking, DNA protein complexes are immunoprecipitated followed by fluorescent labeling of the DNA and hybridization to a DNA microarray. (b) ChIP-PET also searches for DNA binding sites, but uses a high-throughput sequencing method to identify precipitated DNA sequences.

tag or c-myc) that is fused to the TF is necessary. At this point, the DNA fragments are released from the TF-DNA complexes by heat treatment and are finally amplified by PCR (see Section 14.6).

The problem with classical ChIP is that prior knowledge of the target DNA sequence is required to design the PCR primers. The combination of ChIP with array techniques overcomes this limitation, resulting in the ChIP-on-Chip method (Figure 14.10a) [43]. The initial steps of both methods are identical, but for ChIP-on-Chip all precipitated DNA fragments are labeled with one type of fluorescent dye (e.g., red) and a reference DNA sample (DNA nonenriched in TF binding sites) is labeled with a different dye (e.g., green). A mixture of enriched and control DNA is then hybridized to a chip containing DNA from the studied species. Positions that contain DNA from TF binding sites will display a different color than unspecific sites, because the hybridization sample contains a larger fraction of DNA fragments coming from the enriched sample, labeled with the red dye. The resolution, coverage, and density of the DNA chip

are important factors that control the quality and quantity of the identified binding sites. The method has been used successfully to determine the genome-wide locations of transcription factors in yeast [44] and *Drosophila* [45]. It has also been applied to recombination [46], replication [47], and studies of chromatin structure [48].

An important limitation of ChIP-on-Chip is that the parts of the genome that contain the DNA binding sites have to be represented on the microarray. For DNA binding proteins, these are normally intergenic regions, which make up a large fraction of the DNA of higher eukaryotes. ChIP-PET (paired end ditag) avoids these problems by combining chromatin immunoprecipitation with a high-throughput sequencing approach (Figure 14.10b) [49,50]. The precipitated DNA sequences are cloned into a vector with flanking *MmeI* restriction sites. The corresponding restriction enzyme cuts 18 bp away from its recognition site, so that restriction followed by religation results in a 36 bp ditag insert, consisting of the 5'- and 3'-ends of the isolated DNA fragment. High-throughput sequencing of a large number of ditags (to

improve the signal-to-noise ratio) is finally used to identify different genomic binding sites.

This technique, in principle, allows identifying arbitrary sequences that are not present on any DNA chip. Although currently hampered by high costs for sequencing, this problem might be overcome by the use of next-generation sequencing techniques (Section 14.7).

14.15 Green Fluorescent Protein

For the understanding of complex biological processes, a quantitative description of the participating components is necessary. Traditionally, quantification methods start with a tissue sample or large number of cells, which are then broken up to measure the protein of interest. This approach has a low time resolution since it is difficult and labor intensive to generate data points at short time intervals and often a relatively large sample size is required.

The discovery of GFP (green fluorescent protein) and its use to quantify arbitrary proteins, resolved in time and space, has revolutionized the field. GFP is a small protein (27 kDa) originally found in the jellyfish *Aequorea victoria* [51] that exhibits a strong green fluorescence. Two main approaches have been developed to use GFP as a reporter protein. The first is to simply clone the GFP protein behind a promoter of interest and introduce the construct into a cell. A fluorescence then indicates that the promoter is active, implying that endogenous proteins, encoded by the same promoter are also now synthesized. The second approach is to place the DNA for GFP directly at the beginning or end of the coding sequence of a protein under investigation, creating a fusion protein. Such an approach normally does not interfere with the functioning of the studied protein and allows not only analyzing when a protein is synthesized but also where in the cell it is located. Over the years, several modifications of the original GFP protein have been created that greatly enhance its usability. Variants have been developed with optimized folding capabilities, sensitivity to pH, shorter half-life (for better time resolution), and different emission wavelengths (yielding, for example, blue, cyan, and yellow variants) [52–55].

Because of the excellent time resolution, data from GFP measurements are also very interesting for systems biology. Work by Ronen *et al.* [56] showed that kinetic parameters of *Escherichia coli* promoters can be determined in parallel by using GFP reporter constructs to measure promoter activity with high accuracy and temporal resolution. For this purpose, the promoter region of interest was cloned in front of a GFP gene and the whole

construct was placed on a plasmid together with a selection marker and a low copy origin of replication (see Section 14.2). The authors used this approach to study eight operons of the bacterial SOS repair system by measuring fluorescence and optical density (OD) in intervals of 3 min. From the resulting 99 data points per operon, kinetic parameters of the promoters could be derived using a simple mathematical approach (see Section 9.3.5).

The following table shows the obtained values for the eight studied promoters. For six of the eight cases, the mean error for the predicted promoter activity is below 22%, which is a very good quantitative prediction. The genes *uvrY* and *polB*, however, showed errors of 30–45%, indicating that these genes are possibly influenced by additional factors. This study shows that kinetic data can be obtained using an approach that can, in principle, be scaled up to the whole genome.

Gene	k	β	Error	Function
<i>uvrA</i>	0.09 ± 0.04	2800 ± 300	0.14	Nucleotide excision repair
<i>lexA</i>	0.15 ± 0.08	2200 ± 100	0.10	Transcriptional repressor
<i>recA</i>	0.16 ± 0.07	3300 ± 200	0.12	LexA autocleavage, replication fork blocking
<i>umuD</i>	0.19 ± 0.1	330 ± 30	0.21	Mutagenesis repair
<i>polB</i>	0.35 ± 0.15	70 ± 10	0.31	trans-Lesion DNA synthesis, replication fork recovery
<i>ruvA</i>	0.37 ± 0.1	30 ± 2	0.22	Double-strand break repair
<i>uvrD</i>	0.65 ± 0.3	170 ± 20	0.20	Nucleotide excision repair, recombination repair
<i>uvrY</i>	0.51 ± 0.25	300 ± 200	0.45	Unknown function

The use of GFP and its variants for measuring kinetic and other data has several advantages over traditional experimental techniques.

- It opens the possibility to obtain kinetic data in a high-throughput approach.
- The measurements have a high time resolution (one data point every few minutes).
- The kinetic parameters are measured under *in vivo* conditions.
- Using high-throughput flow cytometry and microscopy, it is possible to perform single-cell measurements (see Section 14.16).
- The reproducibility of the measurements is very good (around 10% error).

These attractive features have led to a number of very interesting studies in recent years. One limitation of the described algorithm is that it cannot be applied to systems where more than one transcription factor controls the promoter activity. In this case it is necessary to have a quantitative understanding of how the input signals of different transcription factors are combined into the output signal (promoter activity). Section 9.3.1 describes in detail how this has been achieved for the promoter of the *lac* operon of *E. coli*.

In another study, 52 promoters of *E. coli* amino acid biosynthesis pathways were studied by placing the regulatory regions in front of a GFP gene [57]. Cells were shifted from a medium without any amino acids to a medium that contained a single amino acid and the GFP expression was measured every 8 min for 8 h. The results showed that the promoters of enzymes located early in unbranched pathways have the shortest response time and strongest activity. These design principles agree nicely with the results of a mathematical model that was optimized to achieve a flux goal with minimal enzyme production. The same group extended this GFP-based approach to a genomic scale by generating reporter strains for all intergenic regions in *E. coli* that are larger than 40 base pairs [58]. The resulting library of 2000 constructs was used in a diauxic shift experiment, where cells first feed on glucose and then switch to lactose once the glucose levels are depleted. This led to the discovery of 80 previously unknown promoters.

In a further high-throughput experiment, GFP constructs were used to provide genome-wide information about protein localization in *Saccharomyces cerevisiae* [59]. For each annotated ORF, a pair of oligonucleotides were synthesized with homologies to the desired chromosomal insertion site. After placing the GFP sequence between these short sequences, the whole construct was inserted at the C terminus of each ORF using homologous recombination. This resulted in 4156 fusion proteins (75% of the yeast proteome) with a C-terminal GFP tag. The information regarding the cellular localization of these proteins is publicly available in the “Yeast GFP Fusion Localization Database” (yeastgfp.yeastgenome.org). Together with the spatial information, the website also provides information about individual protein numbers per cell – another important resource for modeling.

14.16 Single-Cell Experiments

Most experimental techniques described so far require a tissue sample containing a large number of cells and the finally obtained rate or concentration represents an

average value for the original cell population. However, several situations exist where this value is misleading and obscures the underlying structure at the single-cell level. This can for instance occur when the cellular process depends on a small number of molecules and is thus subject to stochastic noise [60–62].

In this case, single-cell studies are necessary to reveal individual variability. A high-throughput method that combines flow cytometry with a library of GFP-tagged yeast strains is able to quantify more than 2500 different proteins in individual yeast cells [63]. Using this approach, the fluorescence of approximately 350 000 cells can be measured per minute. Analysis of these data shows that there is considerably intercell variability of different proteins, which is inversely correlated with protein abundance. The data are in good agreement with the hypothesis that the variation arises from the stochastic production and degradation of mRNA molecules that exist in very small numbers (1–2) per cell. But variability is also controlled by biological function and cellular location. Furthermore, cell cycle regulated proteins showed a bimodal distribution, which is very instructive because it reveals that the population of cells was not homogeneous, but consisted of two subpopulations (in different cell cycle phases). This could not have been seen if the measurements were taken on a population of cells.

A similar phenomenon has been observed in the field of aging research. It has been known for a long time that cells accumulate mutations of the mitochondrial DNA (mtDNA) with age. But this phenomenon was regarded as not relevant since the fraction of defective mtDNA is only around 1% [64,65]. However, it turned out that the underlying assumption that mitochondrial damage is distributed homogeneously within a tissue is wrong. Single-cell studies revealed that muscle tissue displays a mosaic pattern of mitochondrial damage. While in old individuals most cells harbor little or no damaged mitochondria, there are a few cells that contain such a large proportion of mitochondrial mutants that the affected cells show physiological deficits (see Section 12.4) [66–68].

Problems of hidden heterogeneity occur not only at the cellular level but also in populations of individuals. In many species, age-specific mortality levels off at advanced ages and one explanation is that the population consists of subgroups exhibiting different aging rates. To test this idea, Wu *et al.* [69] coupled GFP to the endogenous heat shock protein HSP-16.2 in the nematode *C. elegans*. After application of a heat shock, worms were sorted according to fluorescence (HSP-16.2 activity) and it was found that the lifespan of worms correlated with the amount of individual HSP-16.2 activity. This is a strong indication that population heterogeneity contributes to the flattening of mortality rates at advanced ages.

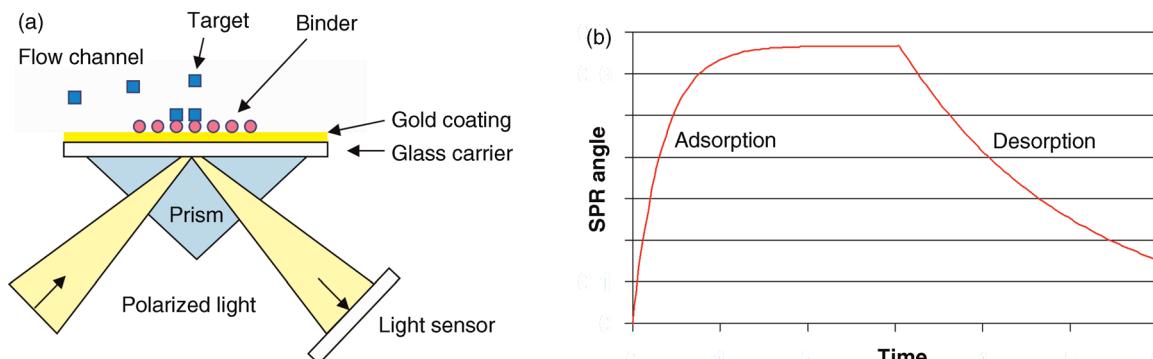


Figure 14.11 (a) Schematic diagram of the surface plasmon resonance (SPR) technique. If target and binder form a complex, the refractive index near the gold surface increases, causing a change in the amount of reflected light. (b) Typical result of an SPR experiment. Initially the flow channel contains target molecules, leading to the formation of target–binder complexes. Later, a washing solution is applied, leading to the desorption of target molecules.

14.17 Surface Plasmon Resonance

One major aim of systems biology is to model large and complex biological systems using mathematical equations. A necessary resource for this kind of quantitative modeling are kinetic data. The surface plasmon resonance (SPR) technique provides these types of data in the form of binding constants (k_{on} , k_{off}) for biochemical reactions [70,71].

The underlying principles of the method are in detail rather intricate, but a simplified overview is given in Figure 14.11a. Polarized light is directed onto a gold-coated glass carrier. When a light beam traveling in a medium of high refractive index meets the interface to a medium of lower refractive index below a certain critical angle, the beam experiences total internal reflection and is directed back into the medium with higher refractive index. During this process, the light beam generates an electrical field (called an evanescent field wave) in the medium of low refractive index. Under the right angle, it also excites surface plasmons (oscillating electrons) in the thin gold layer, which enhance the evanescent wave by extracting energy from the light beam. The angle required for inducing surface plasmons depends strongly on the refractive index of the sample medium close to the surface since the strength of the evanescent wave decays rapidly with its distance from this surface. Thus, the more material is bound to the surface, the higher the refractive index and the stronger the change of the SPR angle.

The sensitivity of SPR depends on the studied biomolecule, and ranges from 10 fmol for single-stranded DNA 18-mers [72] to 1 fmol for the specific binding of antibodies to peptide arrays [73]. By enzymatically amplifying the SPR signal, it was even possible to detect 5 amol of a single-stranded DNA 18-mer [74]. A typical result of

an SPR experiment consists of an adsorption phase during which the carrier is exposed to a solution containing target molecules and a desorption phase during which the surface is exposed to washing fluid (Figure 14.11b).

How are k_{on} and k_{off} values calculated from SPR curves? On the surface, the binding of target and binder ($T + B \xrightleftharpoons[k_{\text{off}}]{k_{\text{on}}} TB$) takes places, leading to the following differential equation for the time-dependent accumulation of TB complexes. It is assumed that target molecules exist in excess, so that their concentration remains unchanged at T_0 , while the concentration of binder molecules is given by the total concentration minus those in the complex TB . Equation (14.1) can be solved analytically and is given by Eq. (14.2) (assuming that the initial concentration of TB is zero). Once the washing fluid is applied, desorption starts and the TB complexes decay according to Eq. (14.3).

$$\frac{dT B}{dt} = k_{\text{on}} \cdot (B_0 - TB) \cdot T_0 - k_{\text{off}} \cdot TB, \quad (14.1)$$

$$TB(t) = \frac{k_{\text{on}} \cdot T_0 \cdot B}{k_{\text{on}} \cdot T_0 + k_{\text{off}}} \cdot (1 - e^{-(k_{\text{on}} \cdot T_0 + k_{\text{off}})t}), \quad (14.2)$$

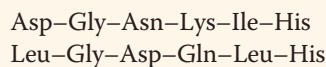
$$TB(t) = TB_0 \cdot e^{-k_{\text{off}} \cdot t}. \quad (14.3)$$

By fitting (14.3) to the desorption curve, it is immediately possible to determine k_{off} . The calculation of k_{on} , however, is more difficult. Because k_{on} always appears together with T in Eq. (14.2), it is not possible to obtain this value from a single measurement as that shown in Figure 14.11b. Instead, a series of such experiments have to be performed with different target concentrations. By fitting Eq. (14.2) to each of these adsorption curves, we obtain estimates of $(k_{\text{on}} \cdot T_0 + k_{\text{off}})$ for different values of T_0 . The linear slope of these data represents k_{on} and the intercept corresponds to k_{off} .

Exercises

- 1) The restriction enzyme Bam HI recognizes a sequence of 6 bp. How many restriction sites do you expect to find in the bacteriophage λ (48502 bp)? How many would you find in *E. coli* (4.6 Mb) if the recognition sequence is 8 bp long?
- 2) Bacteria can reach concentrations of up to 10^7 per milliliter in culture media. How long does it take to reach this concentration if 100 ml medium is inoculated with a single bacterium and the generation time is 20 min?
- 3) For DNA and protein gels, the pore size can be adjusted. When would you use a small pore size and when a large one?
- 4) What is the isoelectric point of a protein?
- 5) What is the purpose of the secondary antibody in Western blotting?
- 6) What is a His tag?
- 7) Is it possible that a 100 kDa protein has a smaller sedimentation coefficient, S , than a 70 kDa protein?
- 8) What are the advantages of HPLC over conventional chromatography?
- 9) Why are protein chips more difficult to generate and use than the DNA chips?
- 10) What accuracy in ppm is necessary for a mass spectrometer to be able to separate the following two peptides according to mass? Consider that

each peptide bond releases one molecule of water. We further assume that the peptides carry no net charge and that the atomic masses are given by the mass of the most frequent isotope rounded to the next dalton (i.e., C = 12 Da).



- 11) If a transgenic animal is heterozygous for a transgene, what is the proportion of offspring that are homozygous for the transgene if the animal is crossed (a) with another heterozygous animal or (b) with wild-type animals?
- 12) What are the advantages and disadvantages of the RNAi technique in contrast to knockout animals?
- 13) The analysis of binding curves of surface plasmon resonance experiments is based on the assumption that one binder molecule reacts with one target molecule $(T + B \xrightleftharpoons[k_{\text{off}}]{k_{\text{on}}} TB)$. Try to develop the differential equations that would result if one binder reacts with two targets $(2T + B \xrightleftharpoons[k_{\text{off}}]{k_{\text{on}}} T_2B)$, as could be expected if the binder were antibodies. Try to solve the differential equations for the washing period.

References

- 1 Burke, D.T., Carle, G.F., and Olson, M.V. (1987) Cloning of large segments of exogenous DNA into yeast by means of artificial chromosome vectors. *Science*, 236, 806–812.
- 2 Shizuya, H., Birren, B., Kim, U.J., Mancino, V., Slepak, T., Tachiiri, Y. et al. (1992) Cloning and stable maintenance of 300-kilobase-pair fragments of human DNA in *Escherichia coli* using an F-factor-based vector. *Proc. Natl. Acad. Sci. USA*, 89, 8794–8797.
- 3 Shapiro, A.L., Vinuela, E., and Maizel, J.V.J. (1967) Molecular weight estimation of polypeptide chains by electrophoresis in SDS-polyacrylamide gels. *Biochem. Biophys. Res. Commun.*, 28 (5), 815–820.
- 4 O'Farrell, P.H. (1975) High resolution two-dimensional electrophoresis of proteins. *J. Biol. Chem.*, 250 (10), 4007–4021.
- 5 Tonge, R., Shaw, J., Middleton, B., Rowlinson, R., Rayner, S., Young, J. et al. (2001) Validation and development of fluorescence two-dimensional differential gel electrophoresis proteomics technology. *Proteomics*, 1 (3), 377–396.
- 6 Southern, E.M. (1975) Detection of specific sequences among DNA fragments separated by gel electrophoresis. *J. Mol. Biol.*, 98 (3), 503–517.
- 7 Saiki, R.K., Scharf, S., Faloona, F., Mullis, K.B., Horn, G.T., Erlich, H.A. et al. (1985) Enzymatic amplification of beta-globin genomic sequences and restriction site analysis for diagnosis of sickle cell anemia. *Science*, 230 (4732), 1350–1354.
- 8 Maxam, A.M. and Gilbert, W. (1977) A new method for sequencing DNA. *Proc. Natl. Acad. Sci. USA*, 74 (2), 560–564.
- 9 Sanger, F., Nicklen, S., and Coulson, A.R. (1977) DNA sequencing with chain-terminating inhibitors. *Proc. Natl. Acad. Sci. USA*, 74 (12), 5463–5467.
- 10 Venter, J.C., Adams, M.D., Myers, E.W., Li, P.W., Mural, R.J., Sutton, G.G. et al. (2001) The sequence of the human genome. *Science*, 291 (5507), 1304–1351.
- 11 Metzker, M.L. (2010) Sequencing technologies: the next generation. *Nat. Rev. Genet.*, 11 (1), 31–46.
- 12 van Dijk, E.L., Auger, H., Jaszczyzyn, Y., and Thermes, C. (2014) Ten years of next-generation sequencing technology. *Trends Genet.*, 30 (9), 418–426.
- 13 DeRisi, J.L., Iyer, V.R., and Brown, P.O. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278 (5338), 680–686.
- 14 Lee, C.-K., Klopp, R.G., Weindruch, R., and Prolla, T.A. (1999) Gene expression profile of aging and its retardation by caloric restriction. *Science*, 285, 1390–1393.

- 15** Causton, H.C., Ren, B., Koh, S.S., Harbison, C.T., Kanin, E., Jennings, E.G. *et al.* (2001) Remodeling of yeast genome expression in response to environmental changes. *Mol. Biol. Cell*, 12 (2), 323–337.
- 16** Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95 (25), 14863–14868.
- 17** Cahill, D.J. and Nordhoff, E. (2003) Protein arrays and their role in proteomics. *Adv. Biochem. Eng. Biotechnol.*, 83, 177–187.
- 18** Abel, L., Kutschki, S., Turewicz, M., Eisenacher, M., Stoutjesdijk, J., Meyer, H.E. *et al.* (2014) Autoimmune profiling with protein microarrays in clinical applications. *Biochim. Biophys. Acta*, 1844 (5), 977–987.
- 19** Wierling, C.K., Steinath, M., Elge, T., Schulze-Kremer, S., Aanstad, P., Clark, M. *et al.* (2002) Simulation of DNA array hybridization experiments and evaluation of critical parameters during subsequent image and data analysis. *BMC Bioinformatics*, 3 (1), 29.
- 20** Wang, Z., Gerstein, M., and Snyder, M. (2009) RNA-Seq: a revolutionary tool for transcriptomics. *Nat. Rev. Genet.*, 10 (1), 57–63.
- 21** Corney, D.C. (2013) RNA-Seq using next generation sequencing. *Mat. Methods*, 3, 203.
- 22** Mortazavi, A., Williams, B.A., McCue, K., Schaeffer, L., and Wold, B. (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Methods*, 5 (7), 621–628.
- 23** Uetz, P., Giot, L., Cagney, G., Mansfield, T.A., Judson, R.S., Knight, J.R. *et al.* (2000) A comprehensive analysis of protein–protein interactions in *Saccharomyces cerevisiae*. *Nature*, 403 (6770), 623–627.
- 24** Ito, T., Chiba, T., Ozawa, R., Yoshida, M., Hattori, M., and Sakaki, Y. (2001) A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. USA*, 98 (8), 4569–4574.
- 25** Karas, M. and Hillenkamp, F. (1988) Laser desorption ionization of proteins with molecular masses exceeding 10,000 daltons. *Anal. Chem.*, 60 (20), 2299–2301.
- 26** Fenn, J.B., Mann, M., Meng, C.K., Wong, S.F., and Whitehouse, C.M. (1989) Electrospray ionization for mass spectrometry of large biomolecules. *Science*, 246 (4926), 64–71.
- 27** Domon, B. and Aebersold, R. (2006) Mass spectrometry and protein analysis. *Science*, 312 (5771), 212–217.
- 28** Gordon, J.W. and Ruddle, F.H. (1981) Integration and stable germ line transmission of genes injected into mouse pronuclei. *Science*, 214 (4526), 1244–1246.
- 29** Gossler, A., Doetschman, T., Korn, R., Serfling, E., and Kemler, R. (1986) Transgenesis by means of blastocyst-derived embryonic stem cell lines. *Proc. Natl. Acad. Sci. USA*, 83 (23), 9065–9069.
- 30** Urnov, F.D., Rebar, E.J., Holmes, M.C., Zhang, H.S., and Gregory, P.D. (2010) Genome editing with engineered zinc finger nucleases. *Nat. Rev. Genet.*, 11 (9), 636–646.
- 31** Gaj, T., Gersbach, C.A., and Barbas, C.F., 3rd (2013) ZFN, TALEN, and CRISPR/Cas-based methods for genome engineering. *Trends Biotechnol.*, 31 (7), 397–405.
- 32** Boch, J., Scholze, H., Schornack, S., Landgraf, A., Hahn, S., Kay, S. *et al.* (2009) Breaking the code of DNA binding specificity of TAL-type III effectors. *Supramol. Sci.*, 326 (5959), 1509–1512.
- 33** Jinek, M., Chylinski, K., Fonfara, I., Hauer, M., Doudna, J.A., and Charpentier, E. (2012) A programmable dual-RNA-guided DNA endonuclease in adaptive bacterial immunity. *Supramol. Sci.*, 337 (6096), 816–821.
- 34** Wang, H., Yang, H., Shivalila, C.S., Dawlaty, M.M., Cheng, A.W., Zhang, F. *et al.* (2013) One-step generation of mice carrying mutations in multiple genes by CRISPR/Cas-mediated genome engineering. *Cell*, 153 (4), 910–918.
- 35** Fire, A., Xu, S., Montgomery, M.K., Kostas, S.A., Driver, S.E., and Mello, C.C. (1998) Potent and specific genetic interference by double-stranded RNA in *Caenorhabditis elegans*. *Nature*, 391 (6669), 806–811.
- 36** Dykxhoorn, D.M., Novina, C.D., and Sharp, P.A. (2003) Killing the messenger: short RNAs that silence gene expression. *Nat. Rev. Mol. Cell Biol.*, 4 (6), 457–467.
- 37** Ui-Tei, K. (2013) Optimal choice of functional and off-target effect-reduced siRNAs for RNAi therapeutics. *Front. Genet.*, 4, 107.
- 38** Holen, T., Amarzguioui, M., Wiiger, M.T., Babaie, E., and Prydz, H. (2002) Positional effects of short interfering RNAs targeting the human coagulation trigger tissue factor. *Nucleic Acids Res.*, 30 (8), 1757–1766.
- 39** Stevenson, M. (2003) Dissecting HIV-1 through RNA interference. *Nat. Rev. Immunol.*, 3 (11), 851–858.
- 40** Boutros, M., Agaisse, H., and Perrimon, N. (2002) Sequential activation of signaling pathways during innate immune responses in *Drosophila*. *Dev. Cell*, 3 (5), 711–722.
- 41** Dogini, D.B., Pascoal, V.D., Avansini, S.H., Vieira, A.S., Pereira, T.C., and Lopes-Cendes, I. (2014) The new world of RNAs. *Genet. Mol. Biol.*, 37 (1 Suppl.), 285–293.
- 42** Fukunaga, R. and Zamore, P.D. (2012) Loquacious, a dicer partner protein, functions in both the MicroRNA and siRNA pathways. *Enzymes*, 32, 37–68.
- 43** Buck, M.J. and Lieb, J.D. (2004) ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics*, 83 (3), 349–360.
- 44** Lee, T.I., Rinaldi, N.J., Robert, F., Odom, D.T., Bar-Joseph, Z., Gerber, G.K. *et al.* (2002) Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, 298 (5594), 799–804.
- 45** Zeitlinger, J., Zinzen, R.P., Stark, A., Kellis, M., Zhang, H., Young, R.A. *et al.* (2007) Whole-genome ChIP-chip analysis of Dorsal, Twist, and Snail suggests integration of diverse patterning processes in the *Drosophila* embryo. *Genes Dev.*, 21 (4), 385–390.
- 46** Gerton, J.L., DeRisi, J., Shroff, R., Lichten, M., Brown, P.O., and Petes, T.D. (2000) Global mapping of meiotic recombination hotspots and coldspots in the yeast *Saccharomyces cerevisiae*. *Proc. Natl. Acad. Sci. USA*, 97 (21), 11383–11390.
- 47** Wysocki, J.J., Aparicio, J.G., Chen, T., Barnett, J.D., Jennings, E.G., Young, R.A. *et al.* (2001) Genome-wide distribution of ORC and MCM proteins in *S. cerevisiae*: high-resolution mapping of replication origins. *Science*, 294 (5550), 2357–2360.
- 48** Robyr, D., Suka, Y., Xenarios, I., Kurdistani, S.K., Wang, A., Suka, N. *et al.* (2002) Microarray deacetylation maps determine genome-wide functions for yeast histone deacetylases. *Cell*, 109 (4), 437–446.
- 49** Wei, C.L., Wu, Q., Vega, V.B., Chiu, K.P., Ng, P., Zhang, T. *et al.* (2006) A global map of p53 transcription-factor binding sites in the human genome. *Cell*, 124 (1), 207–219.
- 50** Hudson, M.E. and Snyder, M. (2006) High-throughput methods of regulatory element discovery. *Biotechniques*, 41 (6), 673, 5, 7 passim.
- 51** Tsien, R.Y. (1998) The green fluorescent protein. *Annu. Rev. Biochem.*, 67, 509–544.
- 52** Pedelacq, J.D., Cabantous, S., Tran, T., Terwilliger, T.C., and Waldo, G.S. (2006) Engineering and characterization of a super-folder green fluorescent protein. *Nat. Biotechnol.*, 24 (1), 79–88.
- 53** Miesenbock, G., De Angelis, D.A., and Rothman, J.E. (1998) Visualizing secretion and synaptic transmission with pH-sensitive green fluorescent proteins. *Nature*, 394 (6689), 192–195.
- 54** Blokpoel, M.C., O'Toole, R., Smeulders, M.J., and Williams, H.D. (2003) Development and application of unstable GFP variants to

- kinetic studies of mycobacterial gene expression. *J. Microbiol. Methods*, 54 (2), 203–211.
- 55 Hadjantonakis, A.K. and Nagy, A. (2001) The color of mice: in the light of GFP-variant reporters. *Histochem. Cell Biol.*, 115 (1), 49–58.
- 56 Ronen, M., Rosenberg, R., Shraiman, B.I., and Alon, U. (2002) Assigning numbers to the arrows: parameterizing a gene regulation network by using accurate expression kinetics. *Proc. Natl. Acad. Sci. USA*, 99 (16), 10555–10560.
- 57 Zaslaver, A., Mayo, A.E., Rosenberg, R., Bashkin, P., Sherro, H., Tsalyuk, M. et al. (2004) Just-in-time transcription program in metabolic pathways. *Nat. Genet.*, 36 (5), 486–491.
- 58 Zaslaver, A., Bren, A., Ronen, M., Itzkovitz, S., Kikoin, I., Shavit, S. et al. (2006) A comprehensive library of fluorescent transcriptional reporters for *Escherichia coli*. *Nat. Methods*, 3 (8), 623–628.
- 59 Huh, W.K., Falvo, J.V., Gerke, L.C., Carroll, A.S., Howson, R.W., Weissman, J.S. et al. (2003) Global analysis of protein localization in budding yeast. *Nature*, 425 (6959), 686–691.
- 60 Ferrell, J.E., Jr., and Machleder, E.M. (1998) The biochemical basis of an all-or-none cell fate switch in *Xenopus oocytes*. *Science*, 280 (5365), 895–898.
- 61 Biggar, S.R. and Crabtree, G.R. (2001) Cell signaling can direct either binary or graded transcriptional responses. *EMBO J.*, 20 (12), 3167–3176.
- 62 Lahav, G., Rosenfeld, N., Sigal, A., Geva-Zatorsky, N., Levine, A.J., Elowitz, M.B. et al. (2004) Dynamics of the p53-Mdm2 feedback loop in individual cells. *Nat. Genet.*, 36 (2), 147–150.
- 63 Newman, J.R., Ghaemmaghami, S., Ihmels, J., Breslow, D.K., Noble, M., DeRisi, J.L. et al. (2006) Single-cell proteomic analysis of *S. cerevisiae* reveals the architecture of biological noise. *Nature*, 441 (7095), 840–846.
- 64 Cortopassi, G.A., Shibata, D., Soong, N.W., and Arnheim, N. (1992) A pattern of accumulation of a somatic deletion of mitochondrial DNA in aging human tissues. *Proc. Natl. Acad. Sci. USA*, 89, 7370–7374.
- 65 Randerath, K., Randerath, E., and Filburn, C. (1996) Genomic and mitochondrial DNA alterations in aging, in *Handbook of the Biology of Aging*, 4th edn (eds L.E. Schneider and J.W. Rowe), Academic Press, London, pp. 198–214.
- 66 Herbst, A., Pak, J.W., McKenzie, D., Bua, E., Bassiouni, M., and Aiken, J.M. (2007) Accumulation of mitochondrial DNA deletion mutations in aged muscle fibers: evidence for a causal role in muscle fiber loss. *J. Gerontol. A Biol. Sci. Med. Sci.*, 62 (3), 235–245.
- 67 Gokey, N.G., Cao, Z., Pak, J.W., Lee, D., McKiernan, S.H., McKenzie, D. et al. (2004) Molecular analyses of mtDNA deletion mutations in microdissected skeletal muscle fibers from aged rhesus monkeys. *Aging Cell*, 3 (5), 319–326.
- 68 Cao, Z., Wanagat, J., McKiernan, S.H., and Aiken, J.M. (2001) Mitochondrial DNA deletion mutations are concomitant with ragged red regions of individual, aged muscle fibers: analysis by laser-capture microdissection. *Nucleic Acids Res.*, 29 (21), 4502–4508.
- 69 Wu, D., Rea, S.L., Yashin, A.I., and Johnson, T.E. (2006) Visualizing hidden heterogeneity in isogenic populations of *C. elegans*. *Exp. Gerontol.*, 41 (3), 261–270.
- 70 Lee, H.J., Yan, Y., Marriott, G., and Corn, R.M. (2005) Quantitative functional analysis of protein complexes on surfaces. *J. Physiol.*, 563 (Part 1), 61–71.
- 71 Wegner, G.J., Wark, A.W., Lee, H.J., Codner, E., Saeki, T., Fang, S. et al. (2004) Real-time surface plasmon resonance imaging measurements for the multiplexed determination of protein adsorption/desorption kinetics and surface enzymatic reactions on peptide microarrays. *Anal. Chem.*, 76 (19), 5677–5684.
- 72 Lee, H.J., Goodrich, T.T., and Corn, R.M. (2001) SPR imaging measurements of 1-D and 2-D DNA microarrays created from microfluidic channels on gold thin films. *Anal. Chem.*, 73 (22), 5525–5531.
- 73 Wegner, G.J., Lee, H.J., and Corn, R.M. (2002) Characterization and optimization of peptide arrays for the study of epitope–antibody interactions using surface plasmon resonance imaging. *Anal. Chem.*, 74 (20), 5161–5168.
- 74 Goodrich, T.T., Lee, H.J., and Corn, R.M. (2004) Enzymatically amplified surface plasmon resonance imaging method using RNase H and RNA microarrays for the ultrasensitive detection of nucleic acids. *Anal. Chem.*, 76 (21), 6173–6178.

Mathematical and Physical Concepts

15

Summary

Mathematical and physical concepts build the basis for data analysis and model building and description. In this chapter, we introduce the most important fundamentals from linear algebra, dynamic systems theory, and statistics that are continually used in the book.

15.1 Linear Algebra

Summary

In the modeling of biochemical systems, many relations do not hold only for one quantity, but for several ones. For example, all metabolites of a pathway have concentrations, which may be concisely represented in a vector of concentrations. These metabolites are involved in a subset of the reactions occurring in this pathway; the respective stoichiometric coefficients may be presented in a matrix. Using techniques of linear algebra helps to understand properties of biological systems. In this chapter, we will shortly recall the classical problem of how to solve a system linear equation, since the solution algorithm represents a basic strategy. Afterward, we will introduce our notions for vectors, matrices, rank, null space, eigenvalues, and eigenvectors.

15.1.1 Linear Equations

A linear equation in n variables x_1, x_2, \dots, x_n is an equation of the form

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b, \quad (15.1)$$

where a_1, a_2, \dots, a_n, b are real numbers. For example, $2x_1 + 5x_2 = 10$ describes a line passing through the points $(x_1, x_2) = (5, 0)$ and $(x_1, x_2) = (0, 2)$. A *system* of m linear equations in n variables x_1, x_2, \dots, x_n is a system of linear

15.1 Linear Algebra

- Linear Equations
- Matrices

15.2 Dynamic Systems

- Describing Dynamics with Ordinary Differential Equations
- Linearization of Autonomous Systems
- Solution of Linear ODE Systems
- Stability of Steady States
- Global Stability of Steady States
- Limit Cycles

15.3 Statistics

- Basic Concepts of Probability Theory
- Descriptive Statistics
- Testing Statistical Hypotheses
- Linear Models
- Principal Component Analysis

15.4 Stochastic Processes

- Chance in Physical Theories
- Mathematical Random Processes
- Brownian Motion as a Random Process
- Markov Processes
- Markov Chains
- Jump Processes in Continuous Time
- Continuous Random Processes
- Moment-Generating Functions

15.5 Control of Linear Dynamical Systems

- Linear Dynamical Systems
- System Response and Linear Filters
- Random Fluctuations and Spectral Density
- The Gramian Matrices
- Model Reduction
- Optimal Control

15.6 Biological Thermodynamics

- Microstate and Statistical Ensemble
- Boltzmann Distribution and Free Energy
- Entropy
- Equilibrium Constant and Energies
- Chemical Reaction Systems
- Nonequilibrium Reactions
- The Role of Thermodynamics in Systems Biology

15.7 Multivariate Statistics

- Planning and Designing Experiments for Case-Control Studies
- Tests for Differential Expression
- Multiple Testing
- ROC Curve Analysis
- Clustering Algorithms
- Cluster Validation
- Overrepresentation and Enrichment Analyses
- Classification Methods

Exercises

References

equations

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2, \\ &\vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m. \end{aligned} \quad (15.2)$$

If $b_1 = b_2 = \cdots = b_n = 0$, the system is *homogenous*. We wish to determine whether the system (15.2) has a solution; that is, whether there exist numbers x_1, x_2, \dots, x_n that satisfy each of the equations simultaneously. We say that the system is *consistent* if it has a solution. Otherwise, the system is called *inconsistent*.

In order to find the solution, we employ the matrix formalism (Section 15.1.2). The matrix \mathbf{A}_c is the *coefficient matrix* of the system and has the dimension $m \times n$, while the matrix \mathbf{A}_a of dimension $m \times (n+1)$ is called the *augmented matrix* of the system:

$$\mathbf{A}_c = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix},$$

$$\mathbf{A}_a = \left(\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{array} \right). \quad (15.3)$$

The solution of a single linear equation with one unknown is easy. A system of linear equations can be solved using the *Gaussian elimination algorithm*. The following terms are needed:

A matrix is in *row-echelon form* if

- 1) all zero rows (if any) are at the bottom of the matrix and
- 2) if two successive rows are nonzero, the second row starts with more zeros than the first (moving from left to right).

Example 15.1

Matrix \mathbf{B}_r is in the row-echelon form and matrix \mathbf{B}_n in the non row-echelon form:

$$\mathbf{B}_r = \begin{pmatrix} 3 & 0 & 0 & 1 \\ 0 & 2 & 2 & 3 \\ 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \mathbf{B}_n = \begin{pmatrix} 3 & 0 & 0 & 1 \\ 0 & 2 & 2 & 3 \\ 0 & 0 & 0 & 4 \\ 0 & 1 & 2 & 0 \end{pmatrix}. \quad (15.4)$$

A matrix is in *reduced row-echelon form* if

- 1) it is in row-echelon form,
- 2) the leading (leftmost nonzero) entry in each nonzero row is 1, and
- 3) all other elements of the column in which the leading entry 1 occurs are zeros.

Example 15.2

\mathbf{A}_1 and \mathbf{A}_2 are matrices in the reduced row-echelon form, \mathbf{A}_3 is not:

$$\mathbf{A}_1 = \begin{pmatrix} 1 & 4 & 0 & 7 \\ 0 & 0 & 1 & 2 \end{pmatrix} \quad \mathbf{A}_2 = \begin{pmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 4 \\ 0 & 0 & 1 & -5 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$$\mathbf{A}_3 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix}. \quad (15.5)$$

The zero matrix of any size is always in the reduced row-echelon form.

The following operations are the ones used on systems of linear equations and do not change the solutions.

There are three types of *elementary row operations* that can be performed on matrices:

- 1) Interchanging two rows: $\mathbf{R}_i \leftrightarrow \mathbf{R}_j$
- 2) Multiplying a row by a real number: $\mathbf{R}_i \rightarrow \alpha \cdot \mathbf{R}_i$
- 3) Adding a multiple of one row to another row: $\mathbf{R}_j \rightarrow \mathbf{R}_j + \alpha \cdot \mathbf{R}_i$.

A matrix \mathbf{A} is row equivalent to matrix \mathbf{B} if \mathbf{B} is obtained from \mathbf{A} by a sequence of elementary row operations.

Example 15.3

Elementary row operations

$$\mathbf{A} = \begin{pmatrix} 2 & 4 \\ 7 & 5 \\ 1 & 2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 \\ 7 & 5 \\ 2 & 4 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 \\ 0 & -9 \\ 2 & 4 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 2 \\ 0 & -9 \\ 1 & 2 \end{pmatrix} = \mathbf{B}. \quad (15.6)$$

$R_1 \leftrightarrow R_3$ $R_2 \rightarrow R_2 - 7R_1$ $R_3 \rightarrow 1/2 \cdot R_3$

Thus, \mathbf{A} and \mathbf{B} are row equivalent.

If \mathbf{A} and \mathbf{B} are row equivalent matrices of two systems of linear equations, then the two systems have the same solution sets – a solution of the one system is a solution of the other. In other words, each row of \mathbf{A} is a linear combination of the rows of \mathbf{B} .

15.1.1.1 The Gaussian Elimination Algorithm

The Gaussian elimination algorithm is a method for solving linear equation systems by transforming the systems augmented matrix \mathbf{A} into its row-equivalent reduced row-echelon form \mathbf{B} by elementary row operations. \mathbf{B} is simpler than \mathbf{A} , and it allows one to read off the consistency or inconsistency of the corresponding equation system and even the complete solution of the equation system.

- 1) Sort the rows such that the upper rows always have less or equal zero entries before the first nonzero entry (counting from the left) than the lower rows. Perform the following row operations. If the mentioned matrix element is zero continue with its next nonzero right neighbor.
- 2) Divide the first row by a_{11} (or in case, by its next nonzero right neighbor a_{1C_1}) and subtract then $a_{i1} \cdot \mathbf{R}_1$ (or $a_{iC_1} \cdot \mathbf{R}_1$) from all other rows i . Now all elements of the first (C_1 -th) column apart from the first are zero.
- 3) Divide the second row by the new value of a_{22} (or a_{2C_2}); subtract $a_{i2} \cdot \mathbf{R}_2$ (or $a_{iC_2} \cdot \mathbf{R}_2$) from all other rows i . Now all elements of the second (C_2 -th) column apart from the second are zero.
- 4) Repeat this for all lower rows until the lowest row or all lower rows contain only zeros.

The reduced row-echelon form to a given matrix is unique.

15.1.1.2 Systematic Solution of Linear Systems

Suppose a system of m linear equations in n unknowns x_1, x_2, \dots, x_n has the augmented matrix \mathbf{A} and \mathbf{A} is row-equivalent to the matrix \mathbf{B} , which is in reduced row-echelon form. \mathbf{A} and \mathbf{B} have the dimension $m \times (n+1)$. Suppose that \mathbf{B} has r nonzero rows and that the leading entry 1 in row i occurs in column number C_i for $1 \leq i \leq r$. Then

$$1 \leq C_1 < C_2 < \dots < C_r \leq n+1. \quad (15.8)$$

The system is inconsistent, if $C_r = n+1$. The last nonzero row of \mathbf{B} has the form $(0, 0, \dots, 0, 1)$. The corresponding equation is

$$0x_1 + 0x_2 + \dots + 0x_n = 1. \quad (15.9)$$

This equation has no solution. Consequently, the original system has no solution.

The system of equations corresponding to the nonzero rows of \mathbf{B} is consistent, if $C_r \leq n$. It holds $r \leq n$.

If $r = n$, then $C_1 = 1, C_2 = 2, \dots, C_n = n$ and the corresponding matrix is

$$\mathbf{B} = \left(\begin{array}{cccc|c} 1 & 0 & \dots & 0 & d_1 \\ 0 & 1 & \dots & 0 & d_2 \\ \vdots & & & \vdots & \vdots \\ 0 & 0 & \dots & 1 & d_n \\ 0 & 0 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 \end{array} \right). \quad (15.10)$$

Example 15.4

$$\begin{pmatrix} 2 & 2 & 2 \\ 1 & 0 & -1 \\ 3 & 2 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & -1 \\ 3 & 2 & 1 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & -1 & -2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & -1 & -2 \end{pmatrix} \rightarrow \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \end{pmatrix} \quad (15.7)$$

$R_1 \rightarrow R_1/2$ $R_2 \rightarrow R_2 - R_1$ $R_3 \rightarrow R_3 - 3R_1$ $R_2 \rightarrow R_2 / -1$ $R_1 \rightarrow R_1 - R_2$
 $R_3 \rightarrow R_3 + R_2$

There is a unique solution $x_1 = d_1, x_2 = d_2, \dots, x_n = d_n$, which can be directly read off from \mathbf{B} .

If $r < n$, the system is underdetermined. There will be more than one solution (in fact, infinitely many solutions). To obtain all solutions take x_{C_1}, \dots, x_{C_r} as *dependent* variables and use the r equations corresponding to the nonzero rows of \mathbf{B} to express these variables in terms of the remaining *independent* variables $x_{C_{r+1}}, \dots, x_{C_n}$, which can assume arbitrary values:

$$\begin{aligned} x_{C_1} &= b_{1n+1} - b_{1C_{r+1}}x_{C_{r+1}} - \dots - b_{1C_n}x_{C_n} \\ &\vdots \\ x_{C_r} &= b_{rn+1} - b_{rC_{r+1}}x_{C_{r+1}} - \dots - b_{rC_n}x_{C_n} \end{aligned} \quad (15.11)$$

In particular, taking $x_{C_{r+1}} = 0, \dots, x_{C_{n-1}} = 0$ and $x_{C_n} = 0$ or $x_{C_n} = 1$ produces at least two solutions.

Example 15.5

Solving the system

$$\begin{aligned} x_1 + x_2 + x_3 &= 0 \\ x_1 - x_2 - x_3 &= 1, \end{aligned} \quad (15.12)$$

with the following augmented and the reduced row-echelon form matrices

$$\mathbf{A} = \left(\begin{array}{ccc|c} 1 & 1 & 1 & 0 \\ 1 & -1 & -1 & 1 \end{array} \right) \quad \mathbf{B} = \left(\begin{array}{ccc|c} 1 & 0 & 0 & 1/2 \\ 0 & 1 & 1 & -1/2 \end{array} \right), \quad (15.13)$$

leads with the choice $x_3 = 1$ to the solution $x_2 = -3/2$ and $x_1 = 1/2$.

A system of linear equations (15.2) with $b_1 = 0, \dots, b_m = 0$ (i.e., a homogeneous system) is always consistent as $x_1 = 0, \dots, x_n = 0$ is always a solution, which is called the *trivial* solution. Any other solution is called a *nontrivial* solution. It holds that a homogeneous system of m linear equations in n unknowns always has a non-trivial solution if $m < n$.

15.1.2 Matrices

15.1.2.1 Basic Notions

Let us consider the space of real numbers \mathbb{R} . A *scalar* is a quantity whose value can be expressed by a real number, that is, by an element of \mathbb{R} . It has a magnitude, but no direction. A *vector* is an element of the space \mathbb{R}^n . It contains numbers for each coordinate of this space, for

$$\text{example, } x = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}.$$

A *matrix* is a rectangular array of $m \times n$ elements of \mathbb{R} in m rows and n columns, like

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} = [a_{ik}]. \quad (15.14)$$

Here and below holds $i = 1, \dots, m$ and $k = 1, \dots, n$.

For our purpose, a vector can be considered as a matrix comprising only one column ($m \times 1$).

In a *zero* matrix $\mathbf{0}$, all elements are zero ($a_{ik} = 0$ for all i, k).

The matrix is *square* matrix, if $m = n$ holds.

A square matrix is a *diagonal* matrix, if $a_{ik} = 0$ for all $i \neq k$.

A diagonal matrix is called *identity* matrix \mathbf{I}_n if it holds

$$a_{ik} = 1, \quad \text{for } i = k, \text{ so } \mathbf{I}_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{pmatrix}.$$

15.1.2.2 Linear Dependency

The vectors x_1, \dots, x_m of type $n \times 1$ are said to be *linearly dependent* if there exist scalars $\alpha_1, \dots, \alpha_m$, not all zero, such that $\alpha_1 x_1 + \dots + \alpha_m x_m = 0$. In other words, one of the vectors can be expressed as a sum over certain scalar multiples of the remaining vectors, or one vector is a linear combination of the remaining vectors. If $\alpha_1 x_1 + \dots + \alpha_m x_m = 0$ has only the trivial solution $\alpha_1 = \dots = \alpha_m = 0$, the vectors are linearly independent. A set of m vectors of type $n \times 1$ is linearly dependent if $m > n$. Equivalently, a linear independent set of m vectors must have $m \leq n$.

15.1.2.3 Basic Matrix Operations

The *transpose* \mathbf{A}^T of a matrix \mathbf{A} is obtained by interchanging rows and columns:

$$\mathbf{A}^T = [a_{ik}]^T = [a_{ki}]. \quad (15.15)$$

The sum of two matrices \mathbf{A} and \mathbf{B} of the same size $m \times n$ is

$$\mathbf{A} + \mathbf{B} = [a_{ik}] + [b_{ik}] = [a_{ik} + b_{ik}]. \quad (15.16)$$

The matrix product of matrix \mathbf{A} with sizes $m \times n$ and matrix \mathbf{B} with size $n \times p$ is

$$\mathbf{A} \cdot \mathbf{B} = \left[\sum_{j=1}^n a_{ij} \cdot b_{jk} \right]. \quad (15.17)$$

A scalar multiple of a matrix \mathbf{A} is

$$\alpha \cdot \mathbf{A} = \alpha \cdot [a_{ik}] = [\alpha \cdot a_{ik}]. \quad (15.18)$$

Subtraction of matrices is composed of scalar multiplication with -1 and summation:

$$\mathbf{A} - \mathbf{B} = \mathbf{A} + (-1) \cdot \mathbf{B}. \quad (15.19)$$

Division of two matrices is not possible. However, for a square matrix \mathbf{A} of size $n \times n$, one may in some cases find the *inverse* matrix \mathbf{A}^{-1} fulfilling

$$\mathbf{A} \cdot \mathbf{A}^{-1} = \mathbf{A}^{-1} \cdot \mathbf{A} = \mathbf{I}_n. \quad (15.20)$$

If the respective inverse matrix \mathbf{A}^{-1} exists, then \mathbf{A} is called *nonsingular (regular)* and *invertible*. If the inverse matrix \mathbf{A}^{-1} does not exist, then \mathbf{A} is called *singular*. The inverse matrix of an invertible matrix is unique. For invertible matrices, it holds

$$(\mathbf{A}^{-1})^{-1} = \mathbf{A}, \quad (15.21)$$

$$(\mathbf{A} \cdot \mathbf{B})^{-1} = \mathbf{B}^{-1} \cdot \mathbf{A}^{-1}. \quad (15.22)$$

Matrix inversion: for the inverse of a 1×1 matrix, it holds that $(a_{11})^{-1} = (a_{11}^{-1})$. The inverse of a 2×2 matrix is calculated as

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} = \frac{1}{ad - bc} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}. \quad (15.23)$$

In general, the inverse of a $n \times n$ matrix is given as

$$\mathbf{A}^{-1} = \frac{1}{\text{Det}\mathbf{A}} \begin{pmatrix} A_{11} & A_{21} & \cdots & A_{n1} \\ A_{12} & A_{22} & \cdots & A_{n2} \\ \vdots & \vdots & & \vdots \\ A_{1n} & A_{2n} & \cdots & A_{nn} \end{pmatrix}, \quad (15.24)$$

where A_{ik} are the adjoints of \mathbf{A} . For $\text{Det}\mathbf{A}$, see below:

If a square matrix \mathbf{A} is invertible, its rows (or columns) are linearly independent. In this case, the linear equation system $\mathbf{A} \cdot \mathbf{x} = \mathbf{0}$ with $\mathbf{x} = (x_1, \dots, x_m)^T$ has only the trivial solution $\mathbf{x} = \mathbf{0}$. If \mathbf{A} is singular, that is, rows (or columns) are linearly dependent, and the linear equation system $\mathbf{A} \cdot \mathbf{x} = \mathbf{0}$ has a nontrivial solution.

The *determinant* of \mathbf{A} ($\text{Det}\mathbf{A}$) is a real or complex number that can be assigned to every square matrix. For the 1×1 matrix (a_{11}) holds $\text{Det}\mathbf{A} = a_{11}$. For a 2×2 matrix, it is calculated as

$$\text{Det} \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}. \quad (15.25)$$

The value of a determinant of higher order can be obtained by an iterative procedure, that is, by expanding the determinant with respect to one row or column: sum up every element of this row (or column) multiplied by the value of its adjoint. The *adjoint* A_{ik} of element a_{ik} is obtained by deleting the i -th row and the k -th column of the determinant (forming the (i,k) minor of \mathbf{A}), calculating the value of the (i,k) minor and multiplying by $(-1)^{i+k}$. For example, a determinant of third order is

$$\begin{aligned} \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} &= a_{11}a_{11} + a_{12}a_{12} + a_{13}a_{13} \\ &= a_{11} \cdot (-1)^2 \cdot \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} + a_{12} \cdot (-1)^3 \cdot \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} \\ &\quad + a_{13} \cdot (-1)^4 \cdot \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \\ &= a_{11} \cdot (a_{22}a_{33} - a_{23}a_{32}) - a_{12} \cdot (a_{21}a_{33} - a_{23}a_{31}) \\ &\quad + a_{13} \cdot (a_{21}a_{32} - a_{22}a_{31}). \end{aligned} \quad (15.26)$$

The value of a determinant is zero, if (a) it contains a zero row or a zero column or if (b) one row (or column) is a linear combination of the other rows (or columns). In this case, the respective matrix is singular.

15.1.2.4 Dimension and Rank

Subspace of a vector space: Let us further consider the vector space \mathbf{V}^n of all n -dimensional column vectors ($n \times 1$). A subset \mathbf{S} of \mathbf{V}^n is called a *subspace* of \mathbf{V}^n if (a) the zero vector belongs to \mathbf{S} , (b) with two vectors belonging to \mathbf{S} also their sum belongs to \mathbf{S} , and (c) with one vector belonging to \mathbf{S} also its scalar multiples belong to \mathbf{S} . Vectors $\mathbf{x}_1, \dots, \mathbf{x}_m$ belonging to a subspace \mathbf{S} form a *basis* of a vector subspace \mathbf{S} , if they are linearly independent and if every vector in \mathbf{S} is a linear combination of $\mathbf{x}_1, \dots, \mathbf{x}_m$. A subspace where at least one vector is nonzero has a basis. In general, a subspace will have more than one basis. Every linear combination of basis vectors is itself a basis. The number of vectors making up a basis is called the dimension of \mathbf{S} ($\dim\mathbf{S}$). For the n -dimensional vector space, it holds $\dim\mathbf{S} \leq n$.

The *rank* of a matrix is an integer associated with a matrix \mathbf{A} of size $m \times n$: $\text{Rank}\mathbf{A}$ is equal to the number of linearly independent columns or rows in \mathbf{A} and equal to the number of nonzero rows in the reduced row-echelon form of the matrix \mathbf{A} . It holds that $\text{Rank}\mathbf{A} \leq m, n$.

Null space of a vector space: The solution of a homogeneous linear equation system, $\mathbf{A} \cdot \mathbf{x} = \mathbf{0}$, leads to the notion *null space* (or *kernel*) of matrix \mathbf{A} . Nontrivial solutions for the vector \mathbf{x} exist if $\text{Rank}\mathbf{A} < m$, that is, if there are linear dependencies between the columns of \mathbf{A} . A

Example 15.6

The matrix

$$\mathbf{A} = \begin{pmatrix} 2 & 1 & 1 \\ 4 & 2 & 2 \end{pmatrix} \quad \text{with} \quad \mathbf{R}_2 \rightarrow \mathbf{R}_2 - 2\mathbf{R}_1 \quad \begin{pmatrix} 2 & 1 & 1 \\ 0 & 0 & 0 \end{pmatrix},$$

has $m = 2$ rows, $n = 3$ columns, and $\text{Rank } \mathbf{A} = 1$.

kernel matrix \mathbf{K} with

$$\mathbf{A} \cdot \mathbf{K} = \mathbf{0}, \quad (15.27)$$

can express these dependencies. The $m - \text{Rank A}$ columns, \mathbf{k}_i , of \mathbf{K} are particular, linearly independent solutions of the homogeneous linear equation system and span the null-space of matrix \mathbf{A} . \mathbf{K} is not uniquely determined: all linear combinations of the vectors \mathbf{k}_i constitute again valid solution. In other terms, postmultiplying matrix \mathbf{K} by a nonsingular square matrix \mathbf{Q} of matching type gives another null-space matrix \mathbf{K}' .

15.1.2.5 Eigenvalues and Eigenvectors of a Square Matrix

Let \mathbf{A} be a $(n \times n)$ square matrix. If λ is a complex number and \mathbf{b} a nonzero complex vector satisfying

$$\mathbf{A} \cdot \mathbf{b} = \lambda \mathbf{b} \quad (15.28)$$

then \mathbf{b} is called an *eigenvector* of \mathbf{A} , while λ is called an *eigenvalue* of \mathbf{A} . Equation (15.28) can be rewritten as $(\mathbf{A} - \lambda \mathbf{I}_n) \cdot \mathbf{b} = 0$. This equation has nontrivial solutions, only if

$$\text{Det}(\mathbf{A} - \lambda \mathbf{I}_n) = 0. \quad (15.29)$$

In this case, there are at most n distinct eigenvalues of \mathbf{A} . Equation (15.29) is called the *characteristic equation* of \mathbf{A} and $\text{Det}(\mathbf{A} - \lambda \mathbf{I}_n)$ is the *characteristic polynomial* of \mathbf{A} . The eigenvalues are the roots of the characteristic polynomial.

For a (2×2) matrix $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ the characteristic polynomial is $\lambda^2 - \lambda \cdot \text{Trace A} + \text{Det A}$, where $\text{Trace A} = a_{11} + a_{22}$ is the sum of the diagonal elements of \mathbf{A} .

Example 15.7

For the matrix $\mathbf{A} = \begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix}$, the characteristic equation reads $\lambda^2 - \lambda \cdot 4 + 3 = (\lambda - 1) \cdot (\lambda - 3) = 0$ and the eigenvalues are $\lambda_1 = 1, \lambda_2 = 3$. The eigenvector equation reads $\begin{pmatrix} 2 - \lambda & 1 \\ 1 & 2 - \lambda \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$. Taking $\lambda_1 = 1$ results in the equation system $\begin{cases} b_1 + b_2 = 0 \\ b_1 + b_2 = 0 \end{cases}$. Thus, it holds $b_1 = -b_2$ with arbitrary values $b_1 \neq 0$. The eigenvectors corresponding to λ_1 are the vectors $\begin{pmatrix} b_1 \\ -b_1 \end{pmatrix}$. For $\lambda_2 = 3$, the corresponding eigenvector is $\begin{pmatrix} b_1 \\ b_1 \end{pmatrix}$.

15.2 Dynamic Systems

Summary

An important problem in the modeling of biological systems is to characterize the dependence of certain properties on time and space. One frequently applied strategy is the description of the change of state variables by differential equations. If only temporal changes are considered, ordinary differential equations (ODEs) are used. For changes in time and space, partial differential equations (PDEs) are appropriate. In this chapter, we will deal with the solution, analysis, and a numerical integration of ordinary differential equations and with basic concepts of dynamical systems theory such as state space, trajectory, steady states, and stability.

15.2.1 Describing Dynamics with Ordinary Differential Equations

The time behavior of biological systems in a deterministic approach can be described by a set of differential equations

$$\frac{dx_i}{dt} = \dot{x}_i = f_i(x_1, \dots, x_n, p_1, \dots, p_l, t) \quad i = 1, \dots, n, \quad (15.30)$$

where x_i are the variables, for example, concentrations, and p_j are the parameters, for example, enzyme concentrations or kinetic constants, and t is the time. We will use the notions dx/dt and \dot{x} interchangeably. In vector notation, Eq. (15.30) reads

$$\frac{d}{dt} \mathbf{x} = \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{p}, t), \quad (15.31)$$

with $\mathbf{x} = (x_1, \dots, x_n)^T$, $\mathbf{f} = (f_1, \dots, f_n)^T$, and $\mathbf{p} = (p_1, \dots, p_l)^T$. For biochemical reaction systems, the functions f_i are frequently given by the contribution of producing and degrading reactions as described for the balance equations in Section 1.2.

15.2.1.1 Notations

ODEs depend on one variable (e.g., time t). Otherwise, they are called PDEs. PDEs are not considered here.

An implicit ODE

$$F(t, x, x', \dots, x^{(n)}) = 0 \quad (15.32)$$

includes the variable t , the unknown function x , and its derivatives up to n th order. An explicit ODE of n th order has the form

$$x^{(n)} = f(t, x, x', \dots, x^{(n-1)}). \quad (15.33)$$

The highest derivative (here n) determines the order of the ODE.

Studying the time behavior of our system, we may be interested in finding solutions of the ODE, that is, finding an n times differentiable function y fulfilling Eq. (15.33). Such a solution may depend on parameters, so-called integration constants, and represents a set of curves. A solution of an ODE of n th order depending on n integration parameters is a *general* solution. Specifying the integration constants, for example, by specifying n initial conditions (for $n = 1$: $x(t = 0) = x^0$) leads to a special or *particular* solution.

We will not show here all possibilities of solving ODEs, instead we will focus on some cases relevant for this book.

If the right-hand sides of the ODEs are not explicitly dependent on time t ($\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{p})$), the system is called autonomous. Otherwise it is nonautonomous. This case will not be considered here.

The system state is a snapshot of the system at a given time that contains enough information to predict the behavior of the system for all future times. The state of the system is described by the set of variables. The set of all possible states is the state space. The number n of independent variables is equal to the dimension of the state space. For $n = 2$, the two-dimensional state space can be called phase plane.

A particular solution of the ODE system $\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{p}, t)$, determined from the general solution by specifying

parameter values \mathbf{p} and initial conditions $\mathbf{x}(t_0) = \mathbf{x}^0$, describes a path through the state space and is called trajectory.

Stationary states or steady states are points $\bar{\mathbf{x}}$ in the phase plane, where the condition $\dot{\mathbf{x}} = \mathbf{0}$ ($\dot{x}_1 = 0, \dots, \dot{x}_n = 0$) is met. At steady state, the system of n differential equations is represented by a system of n algebraic equations for n variables. The equation system $\dot{\mathbf{x}} = \mathbf{0}$ can have multiple solutions referring to multiple steady states. The change of number or stability of steady states upon changes of parameter values p is called a bifurcation.

Linear systems of ODEs have linear functions of the variables as right-hand sides, such as

$$\begin{aligned}\frac{dx_1}{dt} &= a_{11}x_1 + a_{12}x_2 + z_1 \\ \frac{dx_2}{dt} &= a_{21}x_1 + a_{22}x_2 + z_2\end{aligned}, \quad (15.34)$$

or in general $\dot{\mathbf{x}} = \mathbf{A} \cdot \mathbf{x} + \mathbf{z}$. The matrix $\mathbf{A} = \{a_{ik}\}$ is the system matrix containing the system coefficients $a_{ik} = a_{ik}(\mathbf{p})$ and the vector $\mathbf{z} = (z_1, \dots, z_n)^T$ contains inhomogeneities. The linear system is *homogeneous* if $\mathbf{z} = \mathbf{0}$ holds. Linear systems can be solved analytically. Although in real-world problems, the functions are usually nonlinear, linear systems are important as linear approximations in the investigation of steady states.

Example 15.8

The simple linear system

$$\frac{dx_1}{dt} = a_{12}x_2, \quad \frac{dx_2}{dt} = -x_1 \quad (15.35)$$

has the general solution

$$\begin{aligned}x_1 &= \frac{1}{2}e^{-i\sqrt{a_{12}}t} \left(1 + e^{2i\sqrt{a_{12}}t}\right)C_1 - \frac{1}{2}ie^{-i\sqrt{a_{12}}t} \left(-1 + e^{2i\sqrt{a_{12}}t}\right)\sqrt{a_{12}}C_2 \\ x_2 &= \frac{i}{2\sqrt{a_{12}}}e^{-i\sqrt{a_{12}}t} \left(1 + e^{2i\sqrt{a_{12}}t}\right)C_1 + \frac{1}{2}e^{-i\sqrt{a_{12}}t} \left(1 + e^{2i\sqrt{a_{12}}t}\right)C_2\end{aligned}$$

with the integration constants C_1 and C_2 . Choosing $a_{12} = 1$ simplifies the system to

$x_1 = C_1 \cos t + C_2 \sin t$ and $x_2 = C_2 \cos t - C_1 \sin t$. Specification of the initial conditions to $x_1(0) = 2, x_2(0) = 1$ gives the particular solution $x_1 = 2 \cos t + \sin t$ and $x_2 = \cos t - 2 \sin t$. The solution can be presented in the phase plane or directly as functions of time (Figure 15.1).

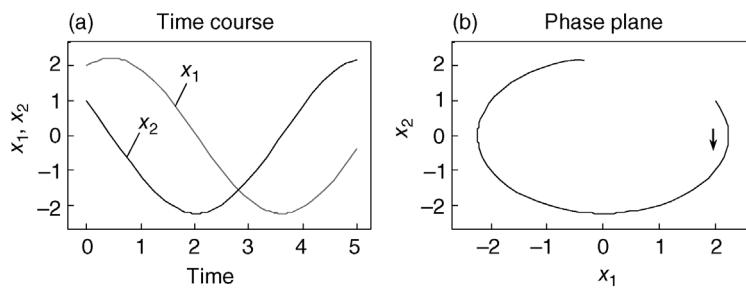


Figure 15.1 Solution of the ordinary differential equation system in Eq. (15.35). Panel (a) shows the plot of dynamics versus time, while panel (b) represents the plot of x_2 versus x_1 , where time is not explicitly shown.

15.2.2

Linearization of Autonomous Systems

In order to investigate the behavior of a system close to steady state, it may be useful to linearize it. Considering the deviation $\hat{\mathbf{x}}(t)$ from steady state with $\mathbf{x}(t) = \bar{\mathbf{x}} + \hat{\mathbf{x}}(t)$, it follows

$$\dot{\mathbf{x}} = f(\bar{\mathbf{x}} + \hat{\mathbf{x}}(t)) = \frac{d}{dt}(\bar{\mathbf{x}} + \hat{\mathbf{x}}(t)) = \frac{d}{dt}\hat{\mathbf{x}}(t). \quad (15.36)$$

Taylor expansion of the temporal change of the deviation, $\frac{d}{dt}\hat{\mathbf{x}}_i = f_i(\bar{x}_1 + \hat{x}_1, \dots, \bar{x}_n + \hat{x}_n)$, gives

$$\begin{aligned} \frac{d}{dt}\hat{\mathbf{x}}_i &= f_i(\bar{x}_1, \dots, \bar{x}_n) + \sum_{j=1}^n \frac{\partial f_i}{\partial x_j} \hat{x}_j \\ &+ \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n \frac{\partial^2 f_i}{\partial x_j \partial x_k} \hat{x}_j \hat{x}_k + \dots \end{aligned} \quad (15.37)$$

Since we consider steady state, it holds $f_i(\bar{x}_1, \dots, \bar{x}_n) = 0$. Neglecting terms of higher order, we have

$$\frac{d}{dt}\hat{\mathbf{x}}_i = \sum_{j=1}^n \frac{\partial f_i}{\partial x_j} \hat{x}_j = \sum_{j=1}^n a_{ij} \hat{x}_j. \quad (15.38)$$

The coefficients $a_{ij} = \partial f_i / \partial x_j$ are calculated at the steady state. They form the so-called *Jacobian* matrix:

$$\mathbf{J} = \{a_{ij}\} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}. \quad (15.39)$$

For linear systems, it holds $\mathbf{J} = \mathbf{A}$.

15.2.3

Solution of Linear ODE Systems

We are interested in two different types of problems: describing the temporal evolution of the system and finding its steady state. The problem of finding the steady state $\bar{\mathbf{x}}$ of a linear ODE system, $\dot{\mathbf{x}} = \mathbf{0}$, implies solution of $\mathbf{A}\bar{\mathbf{x}} + \mathbf{z} = \mathbf{0}$. The problem can be solved by inversion of the system matrix \mathbf{A} :

$$\bar{\mathbf{x}} = -\mathbf{A}^{-1}\mathbf{z}. \quad (15.40)$$

The time course solution of homogeneous linear ODEs is described in the following. The systems can be solved with an exponential function as ansatz. In the simplest case $n = 1$, we have

$$\frac{dx_1}{dt} = a_{11}x_1. \quad (15.41)$$

Introducing the ansatz $x_1(t) = b_1 e^{\lambda t}$ with constant b_1 into Eq. (15.41) yields $b_1 \lambda e^{\lambda t} = a_{11} b_1 e^{\lambda t}$, which is true, if $\lambda = a_{11}$. This leads to the general solution $x_1(t) = b_1 e^{a_{11}t}$. To find a particular solution, we must specify the initial condition $x_1(t=0) = x_1^{(0)} = b_1 e^{a_{11}t}|_{t=0} = b_1$. Thus, the solution is

$$x_1(t) = x_1^{(0)} e^{a_{11}t}. \quad (15.42)$$

For a linear homogeneous system of n differential equations, $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x}$, the ansatz is $\mathbf{x} = \mathbf{b}e^{\lambda t}$. This gives $\dot{\mathbf{x}} = \mathbf{b}\lambda e^{\lambda t} = \mathbf{A}\mathbf{b}e^{\lambda t}$. The scalar factor $e^{\lambda t}$ can be canceled, leading to $\mathbf{b}\lambda = \mathbf{A}\mathbf{b}$ or the characteristic equation

$$(\mathbf{A} - \lambda \mathbf{I}_n) \cdot \mathbf{b} = 0. \quad (15.43)$$

For homogeneous linear ODE systems, the *superposition principle* holds: if \mathbf{x}_1 and \mathbf{x}_2 are solutions of this ODE system, then also their linear combination is a solution. This leads to the general solution of the homogeneous linear ODE system:

$$\mathbf{x}(t) = \sum_{i=1}^n c_i \mathbf{b}^{(i)} e^{\lambda_i t}, \quad (15.44)$$

where $\mathbf{b}^{(i)}$ is the eigenvectors of the system matrix \mathbf{A} corresponding to the eigenvalues λ_i . A particular solution specifying the coefficients c_i can be found considering the initial conditions $\mathbf{x}(t=0) = \mathbf{x}^{(0)} = \sum c_i \mathbf{b}^{(i)}$. This constitutes an inhomogeneous linear equation system to be solved for c_i .

For the solution of inhomogeneous linear ODEs, the system $\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{z}$ can be transformed into a homogeneous system by transformation of coordinates $\hat{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$. Since $\frac{d}{dt}\bar{\mathbf{x}} = \mathbf{A}\bar{\mathbf{x}} + \mathbf{z} = \mathbf{0}$, it holds $\frac{d}{dt}\hat{\mathbf{x}} = \mathbf{A}\hat{\mathbf{x}}$. Therefore, we can use the solution algorithm for homogeneous systems for the transformed system.

15.2.4

Stability of Steady States

If a system is at steady state, it should stay there – until an external perturbation occurs. Depending on the system behavior after perturbation, steady states are either

- *stable* – the system returns to this state
- *unstable* – the system leaves this state
- *metastable* – the system behavior is indifferent

A steady state is *asymptotically stable*, if it is stable and solutions based on nearby initial conditions tend to this state for $t \rightarrow \infty$. *Local* stability describes the behavior after small perturbations, *global* stability after any perturbation.

To investigate, whether a steady state $\bar{\mathbf{x}}$ of the ODE system $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x})$ is asymptotically stable, we consider the

linearized system $d\hat{\mathbf{x}}/dt = \mathbf{A}.\hat{\mathbf{x}}$ with $\hat{\mathbf{x}}(t) = \mathbf{x}(t) - \bar{\mathbf{x}}$. The steady state $\bar{\mathbf{x}}$ is asymptotically stable, if the Jacobian \mathbf{A} has n eigenvalues with strictly negative real parts each. The steady state is unstable, if at least one eigenvalue has a positive real part. This will now be explained in more detail for one- and two-dimensional systems.

We start with one-dimensional systems, that is, $n = 1$, and assume without loss of generality $\bar{x}_1 = 0$. The system $\dot{x}_1 = f_1(x_1)$ yields the linearized system $\dot{x}_1 = \left. \frac{\partial f_1}{\partial x_1} \right|_{\bar{x}_1} x_1 = a_{11}x_1$. The Jacobian matrix $\mathbf{A} = \{a_{11}\}$ has only one eigenvalue $\lambda_1 = a_{11}$. The solution is $x_1(t) = x_1^{(0)}e^{\lambda_1 t}$. It is obvious that $e^{\lambda_1 t}$ increases for $\lambda_1 > 0$ and the system runs away from the steady state. For $\lambda_1 < 0$, the deviation from steady state decreases and $x_1(t) \rightarrow \bar{x}_1$ for $t \rightarrow \infty$. For $\lambda_1 = 0$, consideration of the linearized system allows no conclusion about stability of the original system because higher order terms in Eq. (15.37) play a role.

Consider the two-dimensional case, $n = 2$. For the general (linear or nonlinear) system

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2) \\ \dot{x}_2 &= f_2(x_1, x_2),\end{aligned}\quad (15.45)$$

we can compute the linearized system

$$\begin{aligned}\dot{x}_1 &= \left. \frac{\partial f_1}{\partial x_1} \right|_{\bar{x}} x_1 + \left. \frac{\partial f_1}{\partial x_2} \right|_{\bar{x}} x_2 \\ \dot{x}_2 &= \left. \frac{\partial f_2}{\partial x_1} \right|_{\bar{x}} x_1 + \left. \frac{\partial f_2}{\partial x_2} \right|_{\bar{x}} x_2 \quad \text{or} \\ \dot{\mathbf{x}} &= \left(\begin{array}{cc} \left. \frac{\partial f_1}{\partial x_1} \right|_{\bar{x}} & \left. \frac{\partial f_1}{\partial x_2} \right|_{\bar{x}} \\ \left. \frac{\partial f_2}{\partial x_1} \right|_{\bar{x}} & \left. \frac{\partial f_2}{\partial x_2} \right|_{\bar{x}} \end{array} \right) \dot{\mathbf{x}} \\ \mathbf{x} &= \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \\ \mathbf{x} &= \mathbf{A} \cdot \mathbf{x}.\end{aligned}\quad (15.46)$$

To find the eigenvalues of \mathbf{A} , we have to solve the characteristic polynomial

$$\lambda^2 - \underbrace{(a_{11} + a_{22})}_{\text{TrA}} \lambda + \underbrace{a_{11}a_{22} - a_{12}a_{21}}_{\text{DetA}} = 0,\quad (15.47)$$

with TrA the trace and DetA the determinant of \mathbf{A} , and get

$$\lambda_{1/2} = \frac{\text{TrA}}{2} \pm \sqrt{\frac{(\text{TrA})^2}{4} - \text{DetA}}.\quad (15.48)$$

The eigenvalues are either real for $(\text{TrA})^2/4 - \text{DetA} \geq 0$ or complex (otherwise). For complex eigenvalues, the solution contains oscillatory parts.

For stability, it is necessary that $\text{TrA} < 0$ and $\text{DetA} > 0$. Depending on the sign of the eigenvalues,

steady states of a two-dimensional system may have the following characteristics:

- 1) $\lambda_1 < 0, \lambda_2 < 0$, both real: stable node
- 2) $\lambda_1 > 0, \lambda_2 > 0$, both real: unstable node
- 3) $\lambda_1 > 0, \lambda_2 < 0$, both real: saddle point, unstable
- 4) $\text{Re}(\lambda_1) < 0, \text{Re}(\lambda_2) < 0$, both complex with negative real parts: stable focus
- 5) $\text{Re}(\lambda_1) > 0, \text{Re}(\lambda_2) > 0$, both complex with positive real parts: unstable focus
- 6) $\text{Re}(\lambda_1) = \text{Re}(\lambda_2) = 0$, both complex with zero real parts: center, unstable.

Graphical representation of stability depending on trace and determinant is given in Figure 15.2.

Up to now we considered only the linearized system. For the stability of the original system, the following holds. If the steady state of the linearized system is asymptotically stable, then the steady state of the complete system is also asymptotically stable. If the steady state of the linearized system is a saddle point, an unstable node or an unstable focus, then is the steady state of the complete system also unstable. This means that statements about the stability remain true, but the character of the steady state is not necessarily kept. For the center, no statement on the stability of the complete system is possible.

Routh–Hurwitz theorem [1]: For systems with $n > 2$ differential equations, we obtain the characteristic polynomial

$$c_n \lambda^n + c_{n-1} \lambda^{n-1} + \dots + c_1 \lambda + c_0 = 0.\quad (15.49)$$

This is a polynomial of degree n , which frequently cannot be solved analytically (at least for $n > 4$). We can use the Hurwitz criterion to test, whether the real parts of all eigenvalues are negative. We have to form the Hurwitz matrix \mathbf{H} , containing the coefficients of the characteristic polynomial:

$$\mathbf{H} = \begin{pmatrix} c_{n-1} & c_{n-3} & c_{n-5} & \cdots & 0 \\ c_n & c_{n-2} & c_{n-4} & \cdots & 0 \\ 0 & c_{n-1} & c_{n-3} & \cdots & 0 \\ 0 & c_n & c_{n-2} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & c_0 \end{pmatrix} = \{h_{ik}\} \quad \text{with} \\ h_{ik} = \begin{cases} c_{n+i-2k}, & \text{if } 0 \leq 2k - i \leq n \\ 0, & \text{else} \end{cases}.\quad (15.50)$$

It has been shown that all solutions of the characteristic polynomial have negative real parts, if all coefficients c_i of the polynomial as well as all principal leading minors of \mathbf{H} have positive values.

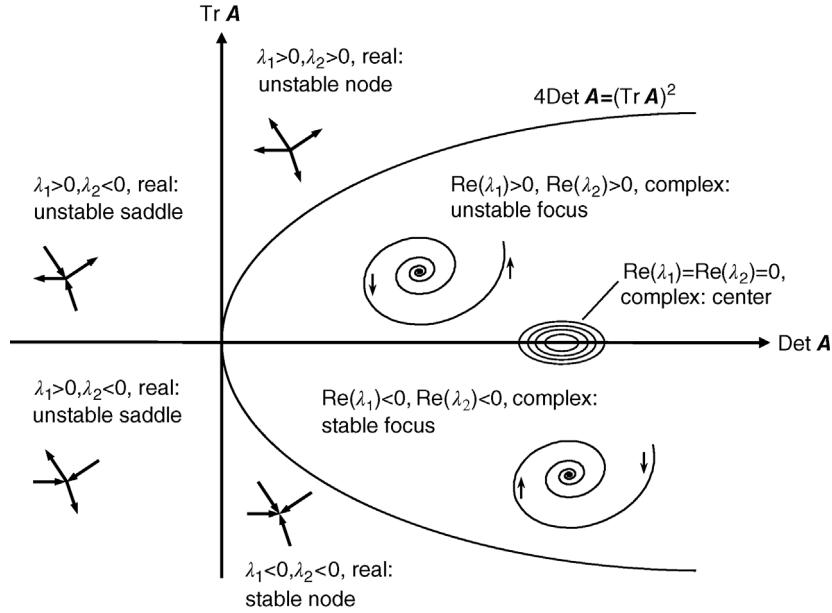


Figure 15.2 Plot of trace versus determinant for a differential equation system with two variables. Indicated are different regions of stability and the respective ranges of eigenvalues as well as the principle behavior of the system in phase plane.

15.2.5 Global Stability of Steady States

A state is globally stable, if the trajectories for all initial conditions approach it for $t \rightarrow \infty$. The stability of a steady state of an ODE system can be tested with a method proposed by Lyapunov:

Shift the steady state into the point of origin by transformation of coordinates $\hat{\mathbf{x}} = \mathbf{x} - \bar{\mathbf{x}}$.

Find a function $V_L(x_1, \dots, x_n)$, called Lyapunov function, with the following properties:

- 1) $V_L(x_1, \dots, x_n)$ has continuous derivatives with respect to all variables x_i .
- 2) $V_L(x_1, \dots, x_n)$ satisfies $V_L(x_1, \dots, x_n) = 0$ for $x_i = 0$ and is positive definite elsewhere, that is, $V_L(x_1, \dots, x_n) > 0$ for $x_i \neq 0$.

The time derivative of $V_L(\mathbf{x}(t))$ is given by

$$\frac{dV_L}{dt} = \sum_{i=1}^n \frac{\partial V_L}{\partial x_i} \frac{dx_i}{dt} = \sum_{i=1}^n \frac{\partial V_L}{\partial x_i} f_i(x_1, \dots, x_n). \quad (15.51)$$

A steady state $\bar{\mathbf{x}} = 0$ is stable, if the time derivative of $V_L(\mathbf{x}(t))$ in a certain region around this state has no positive values. The steady state is asymptotically stable, if the time derivative of $V_L(\mathbf{x}(t))$ in this region is negative definite, that is, $dV_L/dt = 0$ for $x_i = 0$ and $dV_L/dt < 0$ for $x_i \neq 0$.

Example 15.9

The system $\dot{x}_1 = -x_1, \dot{x}_2 = -x_2$ has the solution $x_1(t) = x_1^{(0)} e^{-t}, x_2(t) = x_2^{(0)} e^{-t}$ and the state $x_1 = x_2 = 0$ is asymptotically stable.

The global stability can also be shown using the positive definite function $V_L = x_1^2 + x_2^2$ as Lyapunov function. It holds $dV_L/dt = (\partial V_L/\partial x_1)\dot{x}_1 + (\partial V_L/\partial x_2)\dot{x}_2 = 2x_1(-x_1) + 2x_2(-x_2)$, which is negative definite.

15.2.6 Limit Cycles

Oscillatory behavior is a typical phenomenon in biology. The cause of the oscillation may be different, either imposed by external influences or encoded by internal structures and parameters. Internally caused stable oscillations can be found if we have a limit cycle in the phase space.

A *limit cycle* is an isolated closed trajectory. All trajectories in its vicinity are periodic solutions winding toward (stable limit cycle) or away from (unstable) the limit cycle for $t \rightarrow \infty$ (Figure 15.3).

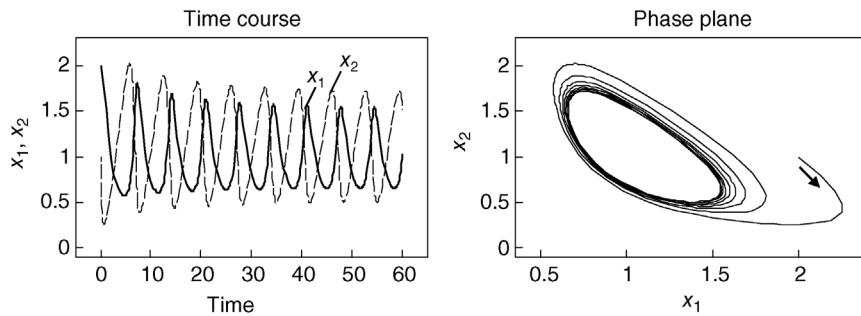


Figure 15.3 Time course and phase plan representation of a oscillatory system exhibiting a stable limit cycle.

Example 15.10

The nonlinear system $\dot{x}_1 = x_1^2 x_2 - x_1, \dot{x}_2 = p - x_1^2 x_2$ has a steady state at $\bar{x}_1 = p, \bar{x}_2 = 1/p$. If we choose for example, $p = 0.98$, this steady state is unstable since $\text{TrA} = 1 - p^2 > 0$.

For two-dimensional systems, there are two criteria to check whether a limit cycle exists. Consider the system of differential equations

$$\begin{aligned}\dot{x}_1 &= f_1(x_1, x_2) \\ \dot{x}_2 &= f_2(x_1, x_2)\end{aligned}\quad (15.52)$$

The *negative criterion of Bendixson* states: if the expression $\text{TrA} = \partial f_1 / \partial x_1 + \partial f_2 / \partial x_2$ does not change its sign in a certain region of the phase plane, then there is no closed trajectory in this area. Hence, a necessary condition for the existence of a limit cycle is the change of the sign of TrA .

Example 15.11

In the Example holds $\text{TrA} = (2x_1 x_2 - 1) + (-x_1^2)$. Therefore, $\text{TrA} = 0$ is fulfilled at $x_2 = (x_1^2 + 1)/(2x_1)$ and TrA may assume positive or negative values for varying x_1, x_2 , and the necessary condition for the existence of a limit cycle is met.

The criterion of Poincaré–Bendixson states: if a trajectory in the two-dimensional phase plane remains within a finite region without approaching a singular point (a steady state), then this trajectory is either a limit cycle or it approaches a limit cycle. This criterion provides a sufficient condition for the existence of a limit cycle. Nevertheless, the limit cycle trajectory can be computed analytically only in very rare cases.

15.3 Statistics

Summary

In this section, we give an introduction to basic concepts of probability theory and statistics. In practice, experimental measurements are afflicted with some uncertainty (concentrations, RNA level, etc.) and statistical concepts give us a framework to quantify this uncertainty. Concepts of *probability theory* provide the necessary mathematical models for computing the significance of the experimental outcome. The focus of *elementary statistics* is to describe the underlying probabilistic parameters by functions on the experimental sample, the sample statistics, and provide tools for visualization of the data. *Statistical test theory* provides a framework for judging the significance of statements (hypotheses) with respect to the data. *Linear Models* are one of the most prominent tools to analyze complex experimental procedures. *Principal component analysis* is a fundamental dimension reduction and visualization method from multivariate statistics in the course of analyzing and simplifying complex data. *Bayesian network analysis* is a widely used method from graphy theory for modeling genetic networks.

15.3.1

Basic Concepts of Probability Theory

15.3.1.1 Probability Spaces

The quantification and characterization of uncertainty is formally described by the concept of a probability space for a random experiment. A *random experiment* is an experiment that consists of a set of possible outcomes with a quantification of the possibility of each outcome. For example, if a coin is tossed one cannot deterministically predict the outcome of “Head” or “Number” but rather assign a probability that either of the outcomes will occur. Intuitively, one will assign a probability of 0.5 if the coin was fair (both outcomes are equally likely).

Random experiments are described by a set of probability axioms.

A *probability space* is a triplet (Ω, A, P) , where Ω is a nonempty set, A is a σ -algebra of subsets of Ω , and P is a probability measure on A . A σ -*algebra* is a family of subsets of Ω that (i) contains Ω itself, (ii) contains for every element $B \in A$ the complementary element $B^c \in A$, and (iii) contains for every series of elements $B_1, B_2, \dots \in A$ their union, that is, $\cup_{i=1}^{\infty} B_i \in A$. The pair (Ω, A) is called a *measurable space*. An element of A is called an *event*. If Ω is discrete, that is, it has at most countable many elements then a natural choice of A would be the *power set* of Ω , $\mathcal{P}(\Omega)$, that is, the set of all subsets of Ω .

A *probability measure* $P : A \rightarrow [0, 1]$ is a real-valued function that has the properties

$$P(B) \geq 0 \quad \text{for all } B \in A \quad \text{and} \quad P(\Omega) = 1,$$

and

$$P\left(\cup_{i=1}^{\infty} B_i\right) = \sum_{i=1}^{\infty} P(B_i) \quad \text{for all series of disjoint sets } B_1, B_2, \dots \in A \text{ (}\sigma\text{-additivity).}$$

Example 15.12 Urn Models

Many practical problems can be described with *urn models*. Consider an urn containing N balls out of which K are red and $N-K$ are black. The random experiment consists of n draws from that urn. If the ball is replaced in the urn after each draw we call the experiment *drawing with replacement* otherwise *drawing without replacement*. Here, Ω is the set of all n -dimensional binary sequences, $\Omega = \{(x_1, \dots, x_n); x_i \in \{0, 1\}\}$ where a "1" means that a red ball was drawn and a "0" means that a black ball was drawn. Since Ω is discrete, a suitable σ -algebra is the power set of Ω . Of practical interest is the calculation of the probability of having exactly k red balls among the n balls drawn. This is given by $P(k) = \binom{n}{k} p^k (1-p)^{n-k}$, with $p = \frac{k}{N}$ if we draw with replacement and $P(k) = \frac{\binom{K}{k} \binom{N-K}{n-k}}{\binom{N}{n}}$ if we draw without replacement. Here, for all numbers $a, b \geq 0$ it is defined

$$\binom{a}{b} = \frac{a!}{(a-b)!b!}. \quad (15.53)$$

We can further define the concept of *conditional dependency*. Let (Ω, A, P) be a probability space. Let $B_1, B_2 \in A$ be two events. In general, there will be some dependency between the two events that influence the probability that both events will occur simultaneously.

For example, consider B_1 being the event that a randomly picked person has lung cancer and let B_2 be the event that the person is a smoker. If both events were independent on each other, then the probability of the joint event, $B_1 \cap B_2$, would be the product of the marginal probabilities, that is, $P(B_1 \cap B_2) = P(B_1)P(B_2)$. That would mean the probability that a randomly picked person has lung cancer is independent on the fact that he is a smoker. Otherwise, the probability of B_1 would be higher conditioned on B_2 .

We can generalize this idea to define another probability measure with respect to any previously given event C with positive probability. For any event $B \in A$, define $P(B|C) = \frac{P(B \cap C)}{P(C)}$, the *conditional probability* of B given C . The measure $P(\cdot|C)$ is a probability measure on the measurable space (Ω, A) (Figure 15.4).

If we have a decomposition of Ω into disjoint subsets $\{B_1, B_2, \dots\}$ with $P(B_i) > 0$ then the probability of any event C can be retrieved by the sum of probabilities with respect to the decomposition, that is,

$$P(C) = \sum_i P(C|B_i)P(B_i). \quad (15.54)$$

Conversely, if $P(C) > 0$ the probability for each B_i conditioned on C can be calculated by *Bayes' rule*, that is,

$$P(B_i|C) = \frac{P(C|B_i)P(B_i)}{\sum_j P(C|B_j)P(B_j)}. \quad (15.55)$$

In the Bayesian set up, the probabilities $P(B_i)$ are called a priori probabilities. These describe the probability of the events with no additional information. If we now consider an event C with positive probability one can ask about the a posteriori probabilities $P(B_i|C)$ of the events

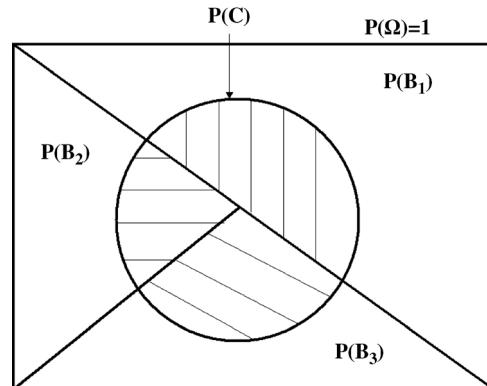


Figure 15.4 Illustration of conditional probability. Three events B_1, B_2, B_3 build a partition of the probability space Ω with a priori probabilities $P(B_1) = 0.5, P(B_2) = P(B_3) = 0.25$. Any event C defines a conditional probability measure with respect to C . Here, the a posteriori probabilities given the event C are $P(B_1) = 0.5, P(B_2) = 0.17$, and $P(B_3) = 0.33$, respectively.

in the light of event C . In practice, formula (15.55) is very important, since many problems do not allow a direct calculation of the probability of an event but rather the probability of the event conditioned on another event or series of other events.

Example 15.13 Power of Diagnostics

Consider a specific disease affecting 0.5% of the population. A diagnostic test with a false positive rate of 5% and a true positive rate of 90% is conducted with a randomly picked person. The test result is positive. What is the probability that this person has the disease? Let B_1 be the event that a person has the disease (B_1^c is the complementary event). Let B_2 be the event that the test is positive. Thus, we are asking for the conditional probability that the person has the disease given that the test is positive, that is, $P(B_1|B_2)$. From Eq. (15.55) we get

$$\begin{aligned} P(B_1|B_2) &= \frac{P(B_2|B_1)P(B_1)}{P(B_2|B_1)P(B_1) + P(B_2|B_1^c)P(B_1^c)} \\ &= \frac{0.9 \cdot 0.005}{0.9 \cdot 0.005 + 0.05 \cdot 0.995} = 0.083. \end{aligned}$$

That means that only 8% of the persons with a positive test result will actually have the disease!

The above effect is due to the fact that the disease is rare and thus that a randomly picked person will have a low chance a priori to have the disease. The diagnostic test, however, will produce a high number of false positives on this sample. The diagnostic power of the test can be improved by decreasing the error rate. For example, decreasing the false positive rate to 1% would give a predictive success of 31.142% (0.5% would give 47.493%).

15.3.1.2 Random Variables, Densities, and Distribution Functions

Let (Ω, A) and (Ω', A') be two measurable spaces, then a function $f : \Omega \rightarrow \Omega'$ is called *measurable*, if $f^{-1}(B') \in A$ for all $B' \in A'$. A measurable function defined on a probability space is called a *random variable*. Random variables are used to describe outcomes of random experiments. A particular result of a random experiment will occur with a given probability.

Of practical interest are real- or vector-valued random variables, that is, $\Omega' = \mathfrak{R}$ or $\Omega' = \mathfrak{R}^n$. In this case a σ -algebra can be defined straightforwardly: Let \mathfrak{I} be the family of all n -dimensional semiopen intervals $Q = (a_1, b_1] \times \dots \times (a_n, b_n]$ then there exists a minimal σ -algebra that contains \mathfrak{I} , the *Borel- σ -algebra*. This σ -algebra contains all sets that one can typically imagine such as all

open, closed, semiopen intervals, and arbitrary mixtures of these. Indeed, it is not straightforward to define sets in \mathfrak{R}^n that are not contained in the Borel- σ -algebra! A random variable is commonly denoted as $x : \Omega \rightarrow \mathfrak{R}$ in order to point to the outcomes (or realizations) of x . The probability measure P defined on Ω induces a probability measure, P_x , on the Borel- σ -algebra on \mathfrak{R} through the equality $P_x(B') = P(x \in B') = P(x^{-1}(B'))$.

If \mathbf{x} is a random vector, then the distribution of \mathbf{x} is uniquely defined by assigning a probability to each n -dimensional vector \mathbf{z} by $F(\mathbf{z}) = P(\mathbf{x} \leq \mathbf{z}) = P(x_1 \leq z_1, \dots, x_n \leq z_n)$. F is called the *cumulative distribution function* of \mathbf{x} . If F admits the n th order mixed partial derivative, then the *density function* of \mathbf{x} is defined as $f(\mathbf{z}) = f(z_1, \dots, z_n) = \frac{\partial^n}{\partial z_1 \dots \partial z_n} F(z_1, \dots, z_n)$ and the relation holds

$$F(\mathbf{z}) = F(z_1, \dots, z_n) = \int_{-\infty}^{z_1} \dots \int_{-\infty}^{z_n} f(t_1, \dots, t_n) dt_1 \dots dt_n. \quad (15.56)$$

If \mathbf{x} is a discrete random vector, that is, if \mathbf{x} can adopt only countable many outcomes, then the density function can be denoted as $f(\mathbf{z}) = P(\mathbf{x} = \mathbf{z}) = P(x_1 = z_1, \dots, x_n = z_n)$. In the discrete case, f is often called the probability mass function of \mathbf{x} .

Example 15.14

In the one-dimensional case, the distribution function of a random variable is defined by $F(t) = P(x \leq t) = P_x((-\infty, t])$. If x is a continuous random variable, the density function f is defined as $P_x((-\infty, t]) = \int_{-\infty}^t f(z)dz$, if x is a discrete random variable, then we have $P_x((-\infty, t]) = \sum_{x \leq t} f(x)$.

Important characteristics of a probability distribution are the mean outcome that one would expect if all possible outcomes together with their respective probabilities were considered (expectation) and the mean squared deviation of the outcomes from the mean outcome (variance). The *expectation* of a random variable, x , is defined as

$$E(x) = \int_{-\infty}^{\infty} tf(t)dt = \mu, \quad (15.57)$$

and the *variance* as

$$\text{Var}(x) = \int_{-\infty}^{\infty} (t - \mu)^2 f(t)dt. \quad (15.58)$$

The variance is equal to $\text{Var}(x) = E(x^2) - E(x)^2$. The covariance of two random variables x and y is defined as

$$\text{Cov}(x, y) = E((x - E(x))(y - E(y))). \quad (15.59)$$

Note that $\text{Var}(x) = \text{Cov}(x, x)$. If x_1, \dots, x_n are random variables with means $E(x_i) = \mu_i$, variances $\text{Var}(x_i) = c_{ii}$ and covariances $\text{Cov}(x_i, x_j) = c_{ij} = c_{ji}$, then the random vector $\mathbf{x} = (x_1, \dots, x_n)^T$ has expectation $E(\mathbf{x}) = \boldsymbol{\mu}$, where $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)^T$, and covariance matrix

$$\text{Cov}(\mathbf{x}) = E((\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^T) = \begin{pmatrix} c_{11} & \cdots & c_{1n} \\ \cdots & \cdots & \cdots \\ c_{n1} & \cdots & c_{nn} \end{pmatrix}. \quad (15.60)$$

A random vector is called *nonsingular* (*singular*) when its covariance matrix is nonsingular (singular).

If \mathbf{x} is an n -dimensional random vector, \mathbf{A} is a $p \times n$ matrix, and \mathbf{b} is a p -dimensional vector, we get the following transformation rules:

$$\begin{aligned} E(\mathbf{Ax} + \mathbf{b}) &= \mathbf{AE}(\mathbf{x}) + \mathbf{b} \quad \text{and} \quad \text{Cov}(\mathbf{Ax} + \mathbf{b}) \\ &= \mathbf{ACov}(\mathbf{x})\mathbf{A}^T. \end{aligned} \quad (15.61)$$

Equation (16.61) gives the expectation of a random vector under an affine transformation. Under general transformations the expectation cannot be calculated straightforwardly from the expectation of \mathbf{x} . However, one can give a lower bound for the expectation of the transformation in some cases that is useful in practice. Let \mathbf{x} be a random vector and let g be a real-valued convex function, that is, a function for which $g(\sum_{i=1}^n \lambda_k \mathbf{x}_k) \leq \sum_{i=1}^n \lambda_k g(\mathbf{x}_k)$, where $0 \leq \lambda_k \leq 1$ and $\sum_{i=1}^n \lambda_k = 1$ (if the inequality is reversed we call g a concave function). Then the inequality holds (*Jensen's inequality*)

$$g(E(\mathbf{x})) \leq E(g(\mathbf{x})). \quad (15.62)$$

Example 15.15

The variance of a random variable is always nonnegative, because $g(x) = x^2$ is a convex function and thus it follows from Equation (15.62): $E(x)^2 \leq E(x^2)$.

Example 15.16

The normal distribution is the most important distribution in probability theory. Numerous methods in test theory and multivariate statistics rely on calculus with the normal distribution (compare also Sections 15.3.3 and 15.3.4). x has a one-dimensional normal (or Gaussian) distribution with

parameters μ, σ^2 if the density of x is equal to

$$f(t) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp^{-\frac{(t-\mu)^2}{2\sigma^2}}. \quad (15.63)$$

This is also denoted as $x \sim N(\mu, \sigma^2)$. The special case $\mu = 0, \sigma^2 = 1$ is called the standard normal distribution. The expectation and the variance of the standard normal distribution can be calculated as $E(x) = 0$ and $\text{Var}(x) = 1$. If z is standard normally distributed, then the random variable $x = \sigma z + \mu$ is distributed with parameters $x \sim N(\mu, \sigma^2)$. From Eq. (15.61) it follows that $E(x) = \mu$ and $\text{Var}(x) = \sigma^2$. The normal distribution can be easily generalized. Let \mathbf{x} be an n -dimensional random vector that follows a normal distribution, then the density of \mathbf{x} is

$$f(\mathbf{z}) = \frac{1}{(2\pi)^{n/2} (\det(\Sigma))^{1/2}} \exp\left(-\frac{1}{2}(\mathbf{z} - \boldsymbol{\mu})^T \Sigma (\mathbf{z} - \boldsymbol{\mu})\right), \quad (15.64)$$

where $\boldsymbol{\mu}$ is the mean vector and Σ is covariance matrix composed of the components x_1, \dots, x_n of \mathbf{x} .

Example 15.17

The exponential distribution is important in modeling decay rates and in the characterization of stochastic processes. A random variable is exponentially distributed with parameter $\lambda > 0$ if the density of x is equal to

$$f(t) = \lambda e^{-\lambda t}, \quad (15.65)$$

where $t \geq 0$. The expectation and variance of x are equal to $E(x) = \frac{1}{\lambda}$ and similar $\text{Var}(x) = \frac{1}{\lambda^2}$.

Example 15.18

An example for a discrete distribution is the Binomial distribution. The Binomial distribution is used to describe urn models with replacement (cf., Example), where we ask specifically after the number of successes in n independent repetitions of a random experiment with binary outcomes (a Bernoulli experiment). If x_i is the random variable that describes the outcome of the i th experiment, then the random variable $x = \sum_{i=1}^n x_i$ has a binomial distribution with probability mass function

$$f(x = k) = \binom{n}{k} p^k (1-p)^{n-k}. \quad (15.66)$$

The expectation and the variance are $E(x) = np$ and $\text{Var}(x) = np(1-p)$.

Example 15.19 Affine Transformations of a Probability Density

Let \mathbf{x} be a random vector with density function f , let \mathbf{h} be a vector-valued affine function $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$, that is, $\mathbf{h}(\mathbf{x}) = \mathbf{Ax} + \boldsymbol{\mu}$ for an $n \times n$ matrix \mathbf{A} and an n -dimensional vector $\boldsymbol{\mu}$, then the density function of the random vector $\mathbf{y} = \mathbf{h}(\mathbf{x})$ is equal to $g(\mathbf{y}) = f(\mathbf{A}^{-1}(\mathbf{y} - \boldsymbol{\mu}))|\det(\mathbf{A}^{-1})|$. In particular, in the one-dimensional case we have the transformation $h(x) = ax + b$ and the corresponding probability density $g(y) = f\left(\frac{y-b}{a}\right)\frac{1}{|a|}$.

Example 15.20 The Density Function of a Log-Normal Distribution

A random variable y is log-normally distributed, if the random variable $\ln(y) = x$ is Gaussian distributed with parameters $x \sim N(\mu, \sigma^2)$. The density of y can be calculated according to the transformation rule, we have $y = h(x) = e^x$ and $h^{-1}(y) = \ln(y)$ and we get the density function of y as $g(y) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(\ln(y)-\mu)^2}{2\sigma^2}}\frac{1}{y}$.

15.3.1.3 Transforming Probability Densities

Let x be a random variable with probability density f then for each measurable function h , $y = h(x)$ $Y = h(x)$ is a random variable too. The *transformation rule* states that if h is a function with strictly positive (negative) derivative and inverse function h^{-1} then y has the density $g(y) = \frac{f(h^{-1}(y))}{|h'(h^{-1}(y))|}$. More generally, let \mathbf{x} be an n -dimensional random vector, let \mathbf{h} be a vector-valued function $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ that is differentiable, that is, \mathbf{h} admits the partial derivatives. Let \mathbf{h}^{-1} be the inverse function with $\det(\mathbf{J}_{\mathbf{h}^{-1}}(\mathbf{y})) \neq 0$ for all $\mathbf{y} \in \mathbb{R}^n$, where $\mathbf{J}_{\mathbf{h}^{-1}}$ is the Jacobi matrix of \mathbf{h}^{-1} . Then, the density of the random vector $\mathbf{y} = \mathbf{h}(\mathbf{x})$ is given by

$$g(\mathbf{y}) = f(\mathbf{h}^{-1}(\mathbf{y}))|\det(\mathbf{J}_{\mathbf{h}^{-1}}(\mathbf{y}))|. \quad (15.67)$$

15.3.1.4 Product Experiments and Independence

Consider n different probability spaces (Ω_i, A_i, P_i) . In many applications the actual interesting probability space would be the *product space* $(\otimes\Omega_i, \otimes A_i, \prod P_i)$. Here, the product set and the product σ -algebra denote the *Cartesian products*. The product measure is defined as the product of the individual probability measures. Implicitly, we have used this definition before, for example, an experiment described by the Binomial distribution is the product experiment of individual Bernoulli experiments.

Let $\mathbf{x} = (x_1, \dots, x_n)^T$ and $\mathbf{y} = (y_1, \dots, y_m)^T$ be two n - and m -dimensional random vectors respectively, then the *joint probability density* of the vector $(\mathbf{x}^T, \mathbf{y}^T)^T$ is defined as $f_{xy}(x_1, \dots, x_n, y_1, \dots, y_m)$ and the *marginal density* f_x of x can be written as

$$\begin{aligned} f_x(\mathbf{x}) &= f_x(x_1, \dots, x_n) \\ &= \int_{-\infty}^{+\infty} \cdots \int_{-\infty}^{+\infty} f_{xy}(x_1, \dots, x_n, y_1, \dots, y_m) dy_1 \cdots dy_m. \end{aligned} \quad (15.68)$$

Two random vectors x and y are *independent* on each other, when the joint probability function is the product of the marginal probability functions, that is,

$$f_{xy}(\mathbf{x}, \mathbf{y}) = f_x(\mathbf{x})f_y(\mathbf{y}). \quad (15.69)$$

Let x_1, x_2 be two independent real-valued random variables with probability densities f_1, f_2 . Many practical problems require the distribution of the sum of the random variables, $y = g(x_1, x_2) = x_1 + x_2$. For each realization c of y we have $g^{-1}(c) = \{(a, b); a + b = c\}$ and we get $P(y \leq c) = \iint_{\{(a,b);a+b \leq c\}} f_1(a)f_2(b) da db = \int_{-\infty}^c \left(\int_{-\infty}^{+\infty} f_1(u-b) f_2(b) db \right) du = \int_{-\infty}^c (f_1 * f_2)(u) du$. The function $f_1 * f_2$ is called the *convolution* of f_1, f_2 .

Example 15.21 Convolution Rule of the Normal Distribution

Let x_1, \dots, x_n be independent random variables that are Gaussian distributed with $x_i \sim N(\mu_i, \sigma_i^2)$. Then, $y = x_1 + \dots + x_n$ is Gaussian distributed $y \sim N(\mu_1 + \dots + \mu_n, \sigma_1^2 + \dots + \sigma_n^2)$.

15.3.1.5 Limit Theorems

In this subsection, we list some fundamental theorems of probability theory that describe the convergence properties of series of random variables. The first theorem states that the empirical distribution function is converging against the true underlying (but unknown) distribution function. The second tells us that the mean and the variance are estimators for the first distribution moments and the third states that distributions for series of random variables converge asymptotically against a Gaussian distribution if they are transformed conveniently.

All convergence properties are defined *almost everywhere*. This technical term of measure theory is introduced to indicate that a result for a probability space is valid everywhere except on subsets of probability zero.

Theorem of Glivenko–Cantelli. Let x_1, \dots, x_n be random variables that are independently and identically distributed with distribution function F . Then the empirical distribution function $F_n(t)$ converges (almost everywhere) to the true distribution function, that is,

$$\sup_{t \in \mathbb{R}} |F_n(t) - F(t)| \rightarrow_{n \rightarrow +\infty} 0 \text{ (almost everywhere).} \quad (15.70)$$

Strong Law of the Large Numbers. Let x_1, x_2, \dots be a series of uncorrelated real-valued random variables with $\text{Var}(x_i) \leq M < +\infty$ for all i , then the series of random variables

$$z_n = \frac{1}{n} \sum_{i=1}^n (x_i - E(x_i)), \quad (15.71)$$

converges to zero (almost everywhere).

Central Limit Theorem. Let Φ be the distribution function of the standard Gaussian distribution. Let x_1, x_2, \dots be a series of independently identically distributed random variables with finite and nonzero variance, that is, $0 < \text{Var}(x_i) < +\infty$. Define the series of random

variables $z_n = \frac{\sum_{i=1}^n x_i - nE(x_1)}{\sqrt{\text{Var}(x_1)\sqrt{n}}}$, then z_n converges to Φ (almost everywhere), that is,

$$\sup_{t \in \mathbb{R}} |z_n(t) - \Phi(t)| \rightarrow_{n \rightarrow +\infty} 0. \quad (15.72)$$

15.3.2

Descriptive Statistics

The basic object of descriptive statistics is the *sample*. A sample is a subset of data measured from an underlying population, for example repeated measurements of expression levels from the same gene. A numerical function of a sample is called a *statistic*. Commonly, a statistic is used to compress the information inherent in the sample and to describe certain properties of the

sample. Interesting features that characterize the sample are

- statistics for sample location,
- statistics for sample variance, and
- statistics for sample distribution.

In the following sections the main concepts are introduced.

15.3.2.1 Statistics for Sample Location

Measures of location describe the center or gravity of the sample. The most commonly used measures of location are the mean and the median. Let x_1, \dots, x_n be a sample of n values, then the *mean* of the sample is defined as

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i, \quad (15.73)$$

and the *median* is defined as the value that is greater than or equal to 50% of the sample elements. For the proper definition of the median, it is necessary to introduce the definition of a *percentile*: Consider the ordered sample $x^{(1)} \leq \dots \leq x^{(n)}$ derived from x_1, \dots, x_n by sorting the sample in increasing order, then the p th percentile is the smallest value that is larger or equal than $p\%$ of the measurements. It is clear that the 0th-percentile and the 100th-percentile are the minimum and the maximum of the sample. The median is the 50th-percentile. If the sample size is odd then the median is defined as $x^{(n+1)/2}$, if the sample size is even the median is not unique. It can be any value between $x^{n/2}$ and $x^{(n/2)+1}$, for example, the average of both values $(x^{n/2} + x^{(n/2)+1})/2$. An important characteristic of the median is its robustness against outlier values. In contrast, the mean value is biased to a large extent by outlier values. In order to robustify the mean value, we define the *α -trimmed mean*. Here, simply the $\alpha\%$ lowest and highest values are deleted from the data set and the mean is calculated of the remaining sample values. Common values of α are between 10 and 20%.

Example 15.22

Consider the measurements of gene expression for a particular gene in a certain tissue in 12 individuals. These individuals represent a common group (disease type) and we want to derive the mean expression level.

Sample Value	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}
2434	2289	5599	2518	1123	1768	2304	2509	14820	2489	1349	1494	

We get the following values for the mean, $\bar{x} = 3391.33$ and for the median, $x_{\text{med}} = 0.5(2304 + 2434) = 2369$. If we look more deeply into the sample, we would rather prefer the median as sample location since most values scatter around the median. The overestimation of the sample location by the mean results from the high values of the outlier value x_9 (and probably x_3). The 10%-trimmed mean of the sample is $\bar{x}_{10} = 2475.3$, which is comparable with the median.

Another measure of location that is preferably used if the sample values represent proportions rather than absolute values is the *geometric mean*. The geometric mean is defined as

$$\bar{x}_g = \sqrt[n]{\prod_{i=1}^n x_i}. \quad (15.74)$$

Note that it always holds that $\bar{x}_g \leq \bar{x}$.

15.3.2.2 Statistics for Sample Variability

Once we have determined the center of the sample, another important bit of information is how the sample values scatter around that center. A very simple way of measuring sample variability is the *range*, the difference between the maximum and the minimum values, $x_{\max} - x_{\min}$. The most common statistic for sample variability is the *standard deviation*,

$$s_n = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2}, \quad (15.75)$$

where s_n^2 is called the *variance* of the sample. The standard deviation measures the individual difference of each sample value and the sample mean. Similar to the mean it is influenced by outlier values. Standard deviations cannot directly be compared since they are dependent on the scale of the values. For example, if two series of distance values were measured in meter and millimeter, the latter one would have a higher standard deviation. In order to compare sample variability from different samples, one

rather compares the relative standard deviations. This measure is called *coefficient of variation*,

$$cv_n = \frac{s_n}{\bar{x}}. \quad (15.76)$$

The interpretation of the coefficient of variation is variability relative to location. A more robust measure of variability is the *interquartile range*, IQR, that is, the difference of the upper and lower quartile of the sample: $IQR_n = x^{([0.75n])} - x^{([0.25n])}$. Here $[an]$ denotes the smallest integer that is greater than or equal to an . Analog to the median another measure is the *median absolute deviation* from the median, MAD,

$$MAD = \text{median}(|x_1 - x_{med}|, \dots, |x_n - x_{med}|). \quad (15.77)$$

Both measures of location, \bar{x} and \bar{x}_{med} , are related to their corresponding measure of variability and can be derived as solutions of a minimization procedure. We have

$$\begin{aligned} \bar{x} &\in \arg \min \left\{ a; \sum_{i=1}^n (x_i - a)^2 \right\} \quad \text{and} \\ x_{med} &\in \arg \min \left\{ a; \sum_{i=1}^n |x_i - a| \right\}. \end{aligned}$$

The sample characteristics are commonly condensed and visualized by a *box plot* (Figure 15.5).

15.3.2.3 Density Estimation

In order to describe the distribution of the sample values across the sample range, one commonly defines the

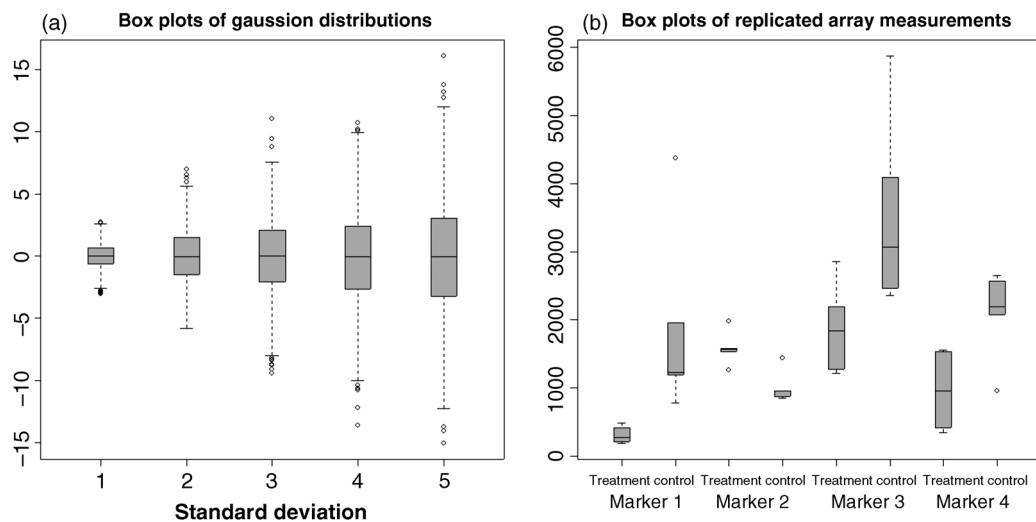


Figure 15.5 Visualization of sample characteristics by box plots. (a) Five different samples from Gaussian distributions with mean $\mu = 0$ and standard deviations $\sigma = 1, 2, 3, 4$, and 5 , respectively were randomly generated. The box displays the interquartile range, the line is the median of the samples. The whiskers denote an area that identifies outliers (circles). Graphics was generated with R-statistical software package. (b) Four marker genes determined in a study on Down's syndrome. Samples are based on six (treatment) and five (control) individuals, respectively. Three markers are downregulated (markers 1, 3, and 4), and one marker gene is upregulated.

histogram. Let I_1, \dots, I_M be disjoint intervals, $I_m = (a_{m-1}, a_m]$, and let $n_m = \{x_i; x_i \in I_m\}$ the number of sample values that fall in the respective interval, then the *weighted histogram* statistic

$$f_n(t) = \begin{cases} \frac{n_m}{n} \frac{1}{a_m - a_{m-1}}, & t \in I_m, \\ 0, & \text{else} \end{cases} \quad (15.78)$$

can be used to estimate the density of the distribution. A fundamental statistic of the sample is the *empirical distribution function*. This is defined as

$$F_n(t) = \frac{1}{n} \sum_{i=1}^n 1_{(-\infty, t]}(x_i), \quad (15.79)$$

where $1_{(-\infty, t]}(x_i) = \begin{cases} 1, & \text{if } x_i \leq t, \\ 0, & \text{else} \end{cases}$ denotes the indicator function. This function is a real-valued step function with values in the interval $[0, 1]$ that has a step at each point x_i . Above we showed that the two statistics above converge to the probability density and the distribution function with respect to an underlying probability law. Figure 15.6 shows as an example, the density, cumulative

distribution function, and the empirical distribution function of a Gaussian distribution.

15.3.2.4 Correlation of Samples

So far we have discussed statistics for samples measured on one variable. Let us now consider a sample measured on two variables, that is, z_1, \dots, z_n , where $z_i = (x_i, y_i)$ is a two-dimensional observation. A fundamental question is whether the two individual samples x_1, \dots, x_n and y_1, \dots, y_n correlate with each other, that is, have a similar trend. A measure of correlation is *Pearson's correlation coefficient*. This is defined as

$$\text{PC} = \frac{\sum_{i=1}^n (x_i - \bar{x}_.) (y_i - \bar{y}_.)}{\sqrt{\sum_{i=1}^n (x_i - \bar{x}_.)^2 \sum_{i=1}^n (y_i - \bar{y}_.)^2}}. \quad (15.80)$$

The Pearson correlation measures the linear relationship of both samples. It is close to one if both samples have strong linear correlation, it is negative if the samples are anticorrelated and it scatters around zero if there is no linear trend observable. Outliers can influence the Pearson correlation to a large extent. Therefore, robust statistics for sample correlation have been defined. We call

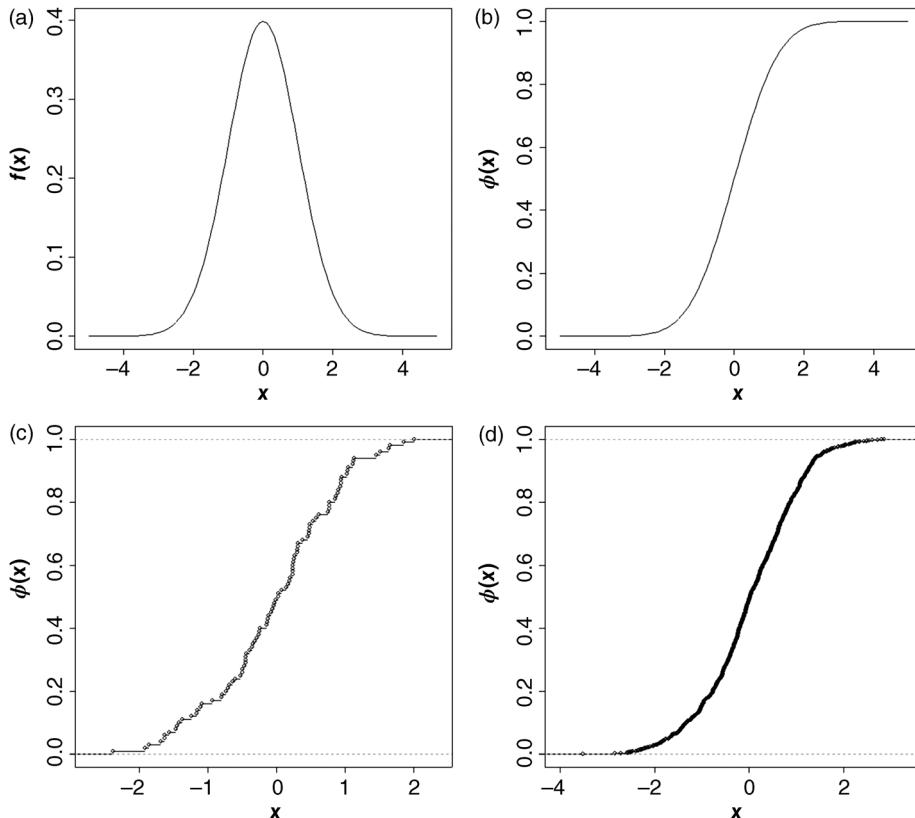


Figure 15.6 Density function (a) and cumulative distribution function (b) of a standard normal distribution with parameters $\mu = 0$ and $\sigma = 1$. Empirical distribution function of a random sample of 100 (c) and 1000 (d) values drawn from a standard normal distribution.

$r_i^x = \#\{x_j; x_j \leq x_i\}$ the rank of x_i within the sample x_1, \dots, x_n . It denotes the number of sample values smaller or equal to the i th value. Note, that the minimum, the maximum, and the median of the sample have ranks, 1, n and $\frac{n}{2}$, respectively and that the ranks and the ordered sample have the correspondence that $x_i = x^{(r_i^x)}$. A more robust measure of correlation than Pearson's correlation coefficient is *Spearman's rank correlation*

$$SC = \frac{\sum_{i=1}^n (r_i^x - \bar{r}^x)(r_i^y - \bar{r}^y)}{\sqrt{\sum_{i=1}^n (r_i^x - \bar{r}^x)^2 \sum_{i=1}^n (r_i^y - \bar{r}^y)^2}}. \quad (15.81)$$

Here, \bar{r}^x denotes the mean rank. SC is derived from PC by replacing the actual sample values by their ranks within the respective sample. Another advantage of this measure is the fact that SC can measure other relationships than linear ones. For example, if the second sample is derived from the first by any monotonic function (square root, logarithm) then the correlation is still high (Figure 15.7). Measures of correlation are extensively used in many algorithms of multivariate statistical

analysis such as pairwise similarity measures for gene expression profiles.

15.3.3

Testing Statistical Hypotheses

Many practical applications imply statements like "it is very likely that two samples are different" or "this fold change of gene expression is significant." Consider the following problems:

- 1) We observe the expression of a gene in replicated measurements of cells with a chemical treatment and control cells. Can we quantify whether a certain observed fold change in gene expression is caused by the treatment?
- 2) We observe the expression of a gene in different individuals of disease and a control group. Is the variability in the two groups equal?
- 3) We measure gene expression of many genes. Does the signal distribution of these genes resemble a specific distribution?

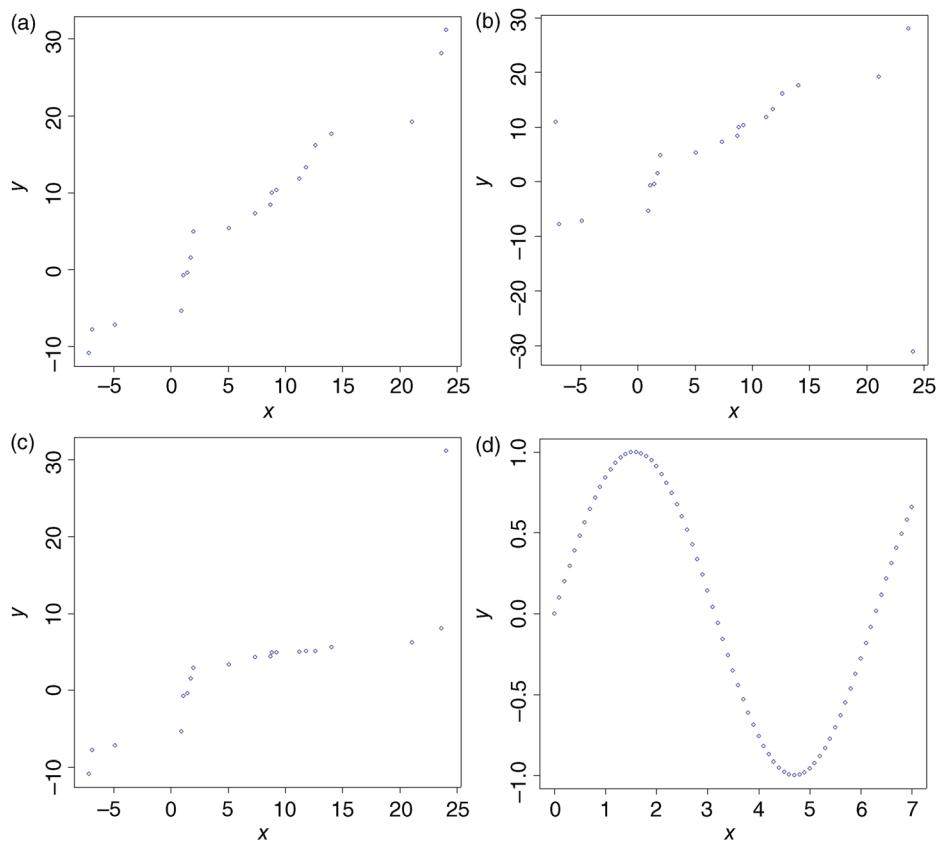


Figure 15.7 Correlation plots and performance of correlation measures. (a) Linear correlation of two random variables ($PC = 0.98$, $SP = 1.00$); (b) presence of two outliers ($PC = 0.27$, $SP = 0.60$); (c) nonlinear, monotonic correlation ($PC = 0.83$, $SP = 1.00$); (d) nonlinear, nonmonotonic correlation ($PC = -0.54$, $SP = -0.54$).

Statistical test theory provides a unique framework to tackle these questions and to determine the significance of these differences.

15.3.3.1 Statistical Framework

Replicated measurements of the same object in a treatment and a control condition typically yield two series of values, x_1, \dots, x_n and y_1, \dots, y_m . The biological problem of judging differences from replicated measurements can be formulated as statistical hypotheses, the null hypothesis, H_0 , and the alternative, H_1 .

An important class of tests is the two-sample location test. Here, the null hypothesis states that the quantities represented by two samples have identical mean values, whereas the alternative states that there is a difference.

$$H_0 : \mu_x = \mu_y \text{ versus } H_1 : \mu_x \neq \mu_y,$$

where μ_x, μ_y are the mean values of the quantities represented by the respective samples.

A very simple argument would be to calculate the averages of the two series and compare the difference. However, this would not allow judging whether a difference different from zero stems from an actual difference between the quantities or just from random scatter of the samples. If we make some additional assumptions, we can describe the problem using an appropriate probability distribution. We regard the two series as realizations of random variables x_1, \dots, x_n and y_1, \dots, y_m . Statistical tests typically have two constraints: (i) It is assumed that repetitions are independent and (ii) that the random variables are identically distributed within each sample. Test decisions are based upon a reasonable test statistic, a real-valued function T , on both samples. For specific functions and using the distributional assumptions it has been shown that they follow a quantifiable probability law given the null hypothesis H_0 .

Suppose that we observe a value of the test statistic $T(x_1, \dots, x_n, y_1, \dots, y_m) = \hat{t}$. If T can be described with a probability law, we can judge the significance of the observation by $\text{prob}(T \text{ more extreme than } \hat{t})$. This probability is called a *P-value*. Thus, if one gives a *P-value* of 0.05 to a certain observation this means that under the distributional assumptions the probability of observing an outcome more extreme than the observed one is 0.05 given the null hypothesis. Observations with a small *P-value* typically give incidence that the null hypothesis should be rejected. This makes it possible to quantify statistically if the result is significant by using a probability distribution. In practice, confidence levels of 0.01, 0.05, and 0.1 are used as upper bounds for significant results. In such a test set up, two types of error occur: error of the first kind and of the second kind:

	H_0 is true	H_1 is true
Test does not reject H_0 (test negative)	No error (TN)	Error of the second kind (FN)
Test rejects H_0 (test positive)	Error of the first kind (FP)	No error (TP)

The *error of the first kind* is the *false positive rate* of the test. Usually, this error can be controlled by the analysis by assuming a significance level α and judging only those results as significant where the probability is lower than α . The *error of the second kind* is the false negative rate of the test. The *power* of a test (given a significance level α) is defined as the probability of rejecting H_0 across the parameter space that is under consideration. It should be low in the subset of the parameter space that belongs to H_0 and high in the subset that belongs to H_1 . The quantities $\frac{TP}{TP+FN}$ and $\frac{TN}{FP+TN}$ are called *sensitivity* and *specificity*, respectively. An optimal test procedure would give a result of 1 to both quantities.

15.3.3.2 Two Sample Location Tests

Assume that the elements of both samples are independently Gaussian distributed, $N(\mu_x, \sigma^2)$ and $N(\mu_y, \sigma^2)$, respectively, with equal variances. Thus, we interpret the sample values x_i as outcomes of independent random variables that are Gaussian distributed with the respective parameters (y_i likewise). We want to test the hypothesis whether the sample means are equal, that is,

$$H_0 : \mu_x = \mu_y \text{ versus } H_1 : \mu_x \neq \mu_y,$$

Under the above assumptions the test statistic

$$T(x_1, \dots, x_n, y_1, \dots, y_m) = \frac{\bar{x} - \bar{y}}{\sqrt{\frac{(n-1)s_x^2 + (m-1)s_y^2}{n+m-2} \sqrt{\frac{1}{n} + \frac{1}{m}}}}, \quad (15.82)$$

can be quantified by a probability distribution. T is distributed according to a *t*-distribution with $m+n-2$ degrees of freedom. The test based on this assumption is called *unpaired Student's t-test*.

For a given value of the *T*-statistic, \hat{t} , we can now judge the probability of having an even more extreme outcome by calculating the probability $P(|T| > |\hat{t}|) = 2P(T > |\hat{t}|) = 2 \int_{\hat{t}}^{\infty} f_{T,p}(z) dz$, where

$$f_{T,p}(z) = \frac{\Gamma(\frac{p+1}{2})}{\Gamma(\frac{p}{2}) \Gamma(\frac{1}{2}) \sqrt{p}} \left(1 + \frac{z^2}{p}\right)^{-(p+1)/2}, \quad (15.83)$$

is the probability distribution of the respective *t*-distribution with p degrees of freedom. Here, $\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt$ is the *gamma function*.

For most practical applications, the assumptions of Student's *t*-test are too strong since the data are often not

Gaussian distributed with equal variances. Furthermore, since the statistic is based on the mean values, the test is not robust against outliers. Thus, if the underlying distribution is not known and in order to define a more robust alternative, we introduce the *Wilcoxon test*. Here, instead of evaluating the signal values, only the ranks of the signals are taken under consideration. Consider the combined series $x_1, \dots, x_n, y_1, \dots, y_m$. Under the null hypothesis this series represents $m + n$ independent identically distributed random variables. The test statistic of the Wilcoxon test is

$$T = \sum_{i=1}^n R_i^{x,y}, \quad (15.84)$$

where $R_i^{x,y}$ is the rank of x_i in the combined series. The minimum and maximum values of T are $(n(n+1))/2$ and $((m+n)(m+n+1))/2 - ((n(n+1))/2)$, respectively. The expected value under the null hypothesis is $E_{H_0}(T) = (mn/2)$ and the variance is $Var_{H_0}(T) = (mn(m+n+1)/12)$. Thus, under the null hypothesis, values for T will scatter around the expectation and unusually low or high values will indicate that the null hypothesis should be rejected. For small sample sizes P -values of the Wilcoxon test can be calculated exactly, for larger sample sizes we have the following approximation

$$P\left(\frac{T - (mn/2)}{\sqrt{mn(m+n+1)/12}} \leq z\right) \rightarrow \Phi(z) \quad \text{for } n, m \rightarrow \infty. \quad (15.85)$$

The P -values of the Wilcoxon test statistic can be approximated by the standard normal distribution. This

approximation has been shown to be accurate for $n + m > 25$.

In practice, some of the series values might be equal, for example because of the resolution of the measurements. Then, the Wilcoxon test statistic can be calculated using *ties*. Ties can be calculated by the average rank of all values that are equal. Ties have an effect on the variance of the statistic, which is often underestimated and should be corrected in the normal approximation. The correction is calculated by replacing the original variance by

$$Var_{H_0,corr}(T) = Var_{H_0}(T) - \frac{mn}{12(m+n)(m+n-1)} \sum_{i=1}^r (b_i^3 - b_i). \quad (15.86)$$

Here, r is the number of different values in the combined series of values and b_i is the frequency.

15.3.4 Linear Models

The general linear model has the form $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, with the assumptions $E(\boldsymbol{\epsilon}) = \mathbf{0}$ and $Cov(\boldsymbol{\epsilon}) = \sigma^2\mathbf{I}$. Here \mathbf{y} is an n -dimensional vector of observations, $\boldsymbol{\beta}$ is a p -dimensional vector of unknown parameters, \mathbf{X} is an $n \times p$ dimensional matrix of known constants (the design matrix), and $\boldsymbol{\epsilon}$ is a vector of random errors. Since the errors are random, \mathbf{y} is a random vector as well. Thus, the observations are separated in a deterministic part and a random part. The ratio behind linear models is that the deterministic part of the experimental observations (dependent variable) is a linear function of the design

Example 15.23

In practice, experimental measurements often contain outliers and Student's *t*-test is very prone to these and might generate wrong results. In this example, expression of a specific gene was measured in cortex brain tissue in two different mouse strains, the control (normal) mouse strain and the Ts65Dn mouse strain, which is a mouse model for the Down syndrome [2]. Repeated array hybridization experiments with different animals yielded the following series of measurements for control mice:

2434, 2289, 5599, 2518, 1123, 1768, 2304, 2509, 14820, 2489, 1349, 1494

and Ts65Dn mice

3107, 3365, 4704, 3667, 2414, 4268, 3600, 3084, 3997, 3673, 2281, 3166.

Due to two outlier values in the control series (5599 and 14820) the trisomic versus control ratio is close to 1, 1.02, and the P -value of Student's *t*-test is not significant, $p = 9.63 \times 10^{-1}$. For the Wilcoxon statistic, however, we get $T = \sum_{i=1}^n R_i^{x,y} = 14 + 16 + 22 + 18 + 8 + 21 + 17 + 13 + 20 + 19 + 5 + 15 = 188$, $E_{H_0}(T) = 72$, $Var_{H_0}(T) = 300$ and for the Z-score we have $z = (116/\sqrt{300}) \sim 6.70$, which indicates that the result is significant. The exact P -value of the Wilcoxon test is $p = 2.84 \times 10^{-2}$.

matrix and the unknown parameter vector. Note that linearity is required in the parameters not in the design matrix. For example, problems such as $x_{ij} = x_i^j$ for $i = 1, \dots, n$ and $j = 0, \dots, p - 1$ are also linear models. Here, for each coordinate i , we have the equation $y_i = \beta_0 + \sum_{j=1}^{p-1} \beta_j x_i^j + \varepsilon_i$ and the model is called *polynomial regression model*.

The goal of linear models is testing of complex statistical hypotheses and parameter estimation. In the following sections, we introduce two classes of linear models, Analysis of Variance and Regression.

15.3.4.1 ANOVA

In Section 15.3.3.2, we introduced a particular test problem – the two-sample location test. Purpose of this test is to judge whether two samples are drawn from the same population or not by the comparison of the centers of these samples. The null hypothesis was $H_0 : \mu_1 = \mu_2$ and the alternative hypothesis was $H_1 : \mu_1 \neq \mu_2$, where μ_i is the mean of the i th sample. A generalization of the null hypothesis is targeted in this section. Assume n different samples where each samples measures the same factor of interest. Within each sample, i , the factor is measured n_i times. This results in a table of the following form:

$$\begin{array}{cccc} x_{11} & x_{21} & \cdots & x_{n1} \\ \cdots & \cdots & \cdots & \cdots \\ x_{1n_1} & x_{1n_2} & \cdots & x_{1n_N} \end{array}$$

Here, the columns correspond to the different individual samples and the rows correspond to the individual repetitions within each sample (the number of rows within each sample can vary!). The interesting question now is, whether there is any difference in the sample means, or, alternatively, whether the samples represent the same population. We thus test the null hypothesis $H_0 : \mu_1 = \mu_2 = \dots = \mu_n$ against the alternative $H_1 : \mu_i \neq \mu_j$ for at least one pair, $i \neq j$. This question is targeted by the so-called *one-way ANOVA*. As in the case of Student's *t*-test, additional assumptions on the data samples are necessary:

- 1) The n samples are drawn independently from each other representing populations with mean values $\mu_1, \mu_2, \dots, \mu_n$.
- 2) All population variances have the same variance σ^2 (*homoscedasticity*).
- 3) All populations are Gaussian distributed, $N(\mu_i, \sigma^2)$.

Although, the one-way ANOVA is based on the analysis of variance, it is essentially a test for location. This is exemplified in Figure 15.8. The idea of ANOVA is the comparison of between- and within-group variability. If the variance between the groups is not different from the variance within the groups, we cannot reject the null hypotheses (Figure 15.8(a)), if the variances differ we would reject the null hypothesis and conclude that the means are different (Figure 15.8(b)).

The calculation of the one-way ANOVA is based on the partition of the sample variance $\sum_{i=1}^n \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_{..})^2$ into

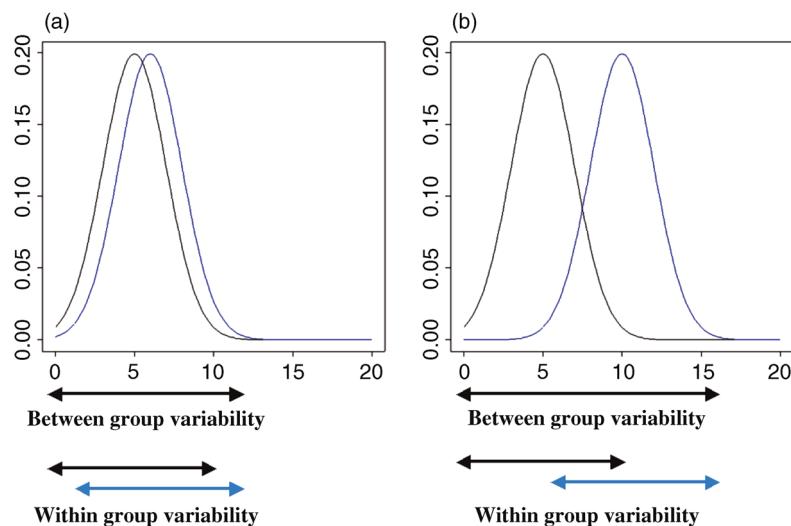


Figure 15.8 ANOVA test for differential expression. (a) Two normal distributions with means $\mu_1 = 5, \mu_2 = 6$ and equal variances. The variability between the groups is comparable with the variability within the groups. (b) Two normal distributions with means $\mu_1 = 5, \mu_2 = 10$ and equal variances. The variability between the groups is higher than the variability within the groups.

two parts that account for the between- and within-group variability, that is,

$$\begin{aligned} SS_{\text{total}} &= \sum_{i=1}^n \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_{..})^2 \\ &= \sum_{i=1}^n \sum_{j=1}^{n_i} (\bar{x}_i - \bar{x}_{..})^2 + \sum_{i=1}^n \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2 \\ &= SS_{\text{between}} + SS_{\text{within}}. \end{aligned} \quad (15.87)$$

We choose the test statistic $T = (SS_{\text{between}}/SS_{\text{within}})((M-n)/(n-1))$, where $M = \sum_{i=1}^n n_i$. It can be shown that

under the null hypothesis M is distributed according to an *F distribution* with degrees of freedom $\nu_1 = n - 1$ and $\nu_2 = M - n$, respectively. The multiplicative constant accounts for the degrees of freedom of the two terms. Thus, we can quantify experimental outcomes according to this distribution. If the μ_i are not equal, SS_{between} will be high compared to SS_{within} and conversely, if all μ_i are equal, then the two factors will be similar and T will be small.

15.3.4.2 Multiple Linear Regression

Let \mathbf{y} be an n -dimensional observation vector (dependent variables) and let $\mathbf{x}_1, \dots, \mathbf{x}_p$ be the independent n -dimensional variables. We assume that the number of observations is greater than the number of variables, that is, $n > p$. The standard model here is

$$\mathbf{y} = \mathbf{X}\beta + \boldsymbol{\varepsilon}, \quad (15.88)$$

where $\mathbf{E}(\boldsymbol{\varepsilon}) = 0$ and $\Sigma_{\boldsymbol{\varepsilon}} = \sigma^2 \mathbf{I}_n$. In our model, we are interested in an optimal estimator for the unknown parameter vector β . The *least-squares method* defines this optimization as a vector $\hat{\beta}$ that minimizes the Euclidean norm of the residuals, that is,

$$\hat{\beta} \in \arg \min \{\beta; \|\mathbf{y} - \mathbf{X}\beta\|^2\}.$$

Using partial derivatives we can transform this problem into a linear equation system by

$$\mathbf{X}^T \mathbf{X} \beta = \mathbf{X}^T \mathbf{y}, \quad (15.89)$$

and, if the inverse exists, get the solution

$$\hat{\beta} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}. \quad (15.90)$$

The solution is called the *least-squares estimator* for β . The least-squares estimator is unbiased, that is, $\mathbf{E}(\hat{\beta}) = \beta$ and the covariance matrix $\Sigma_{\hat{\beta}}$ of $\hat{\beta}$ is equal to $\Sigma_{\hat{\beta}} = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1}$. Through the estimator for β we have an immediate estimator for the error vector $\boldsymbol{\varepsilon}$ using the residuals

$$\hat{\boldsymbol{\varepsilon}} = \mathbf{y} - \mathbf{X}\hat{\beta} = \mathbf{y} - \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} = \mathbf{y} - \mathbf{P}\mathbf{y}. \quad (15.91)$$

Geometrically, \mathbf{P} is the projection of \mathbf{y} in the p -dimensional subspace of \mathfrak{R}^n that is spanned by the columns vectors of \mathbf{X} . An unbiased estimator for the unknown standard deviation of the residuals is given by

$$s^2 = \frac{\|\mathbf{y} - \mathbf{X}\hat{\beta}\|^2}{n-p}, \quad (15.92)$$

thus $E(s^2) = \sigma^2$.

Example 15.24 Simple Linear Regression

An important application is the simple linear regression of two samples x_1, \dots, x_n and y_1, \dots, y_n . Here Eq. (16.88) reduces to

$$\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} = \begin{pmatrix} 1 & x_1 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{pmatrix}$$
 and the parameters of interest are β_1, β_2 the intercept and the slope of the regression

line. Minimizing the Euclidean norm of the residuals computes that line that minimizes the vertical distances of all points to the regression line. Solving according to Eq. (16.90) gives

$$\begin{aligned} \mathbf{X}^T \mathbf{X} &= \begin{pmatrix} n & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n x_i^2 \end{pmatrix} \quad \text{and} \quad \mathbf{X}^T \mathbf{y} = \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{pmatrix} \quad \text{and thus we have} \\ \hat{\beta} &= \frac{1}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2} \begin{pmatrix} \sum_{i=1}^n x_i^2 & -\sum_{i=1}^n x_i \\ -\sum_{i=1}^n x_i & n \end{pmatrix} \begin{pmatrix} \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i y_i \end{pmatrix} = \frac{1}{n \sum_{i=1}^n x_i^2 - \left(\sum_{i=1}^n x_i \right)^2} \begin{pmatrix} \sum_{i=1}^n x_i^2 \sum_{i=1}^n y_i - \sum_{i=1}^n x_i \sum_{i=1}^n x_i y_i \\ n \sum_{i=1}^n x_i y_i - \sum_{i=1}^n x_i \sum_{i=1}^n y_i \end{pmatrix}. \end{aligned}$$

Thus, the slope of the regression line is the correlation of the samples divided by the variance of the dependent variables. The slope of the regression line is called *empirical regression coefficient*.

15.3.5

Principal Component Analysis

Principal component analysis (PCA) is a statistical method to reduce dimensionality and to visualize high-dimensional data in two- or three dimensions. Consider an $n \times p$ expression matrix \mathbf{X} , where rows correspond to genes and columns correspond to experiments. Thus, each gene is viewed as a data vector in the p -dimensional space. In general not all dimensions will contribute equally to the variation across the genes so that we can hope to reduce the overall dimension to the most relevant ones. The idea of PCA is to transform the coordinate system to a system whose axes display the maximal directions of variation of the data sample [3].

Figure 15.9a shows an example for $p=2$. Here, essentially one dimension contains the variation of the sample and thus the dimensionality can be reduced to one after transforming the coordinate system appropriately.

Consider now more generally n p -dimensional data vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ with component-wise mean vector $\bar{\mathbf{x}}$. PCA is computed with a decomposition of the $p \times p$ -dimensional *empirical covariance matrix* of the sample (compare formula (15.57))

$$\mathbf{S} = \frac{1}{n-1} \sum_{i=1}^n (\mathbf{x}_i - \bar{\mathbf{x}})(\mathbf{x}_i - \bar{\mathbf{x}})^T. \quad (15.93)$$

Since the matrix \mathbf{S} is symmetric and positive semidefinite there exist p nonnegative *eigenvalues* $\lambda_1 \geq \dots \geq \lambda_p \geq 0$

(which we may assume to numerate in decreasing order, cf. Section 3.1.2). Let $\mathbf{r}_1, \dots, \mathbf{r}_p$ be the corresponding *eigenvectors* such that

$$\begin{aligned} \mathbf{r}_j^T \mathbf{r}_k &= \begin{cases} 1, & \text{if } j = k \\ 0, & \text{if } j \neq k \end{cases} \\ \mathbf{S} \mathbf{r}_j &= \lambda_j \mathbf{r}_j. \end{aligned} \quad (15.94)$$

If we denote with \mathbf{R} the $p \times p$ -dimensional matrix whose columns are composed of the p eigenvectors of \mathbf{S} and if

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_p \end{pmatrix} \quad \text{denotes the } p \times p\text{-dimensional}$$

diagonal matrix whose diagonal elements are the p eigenvalues we get the decomposition

$$\mathbf{S} = \mathbf{R} \Lambda \mathbf{R}^T. \quad (15.95)$$

Geometrically, the eigenvectors of \mathbf{S} are the main axes of dispersion of the data set $\{\mathbf{x}_1, \dots, \mathbf{x}_n\}$. The dispersion is maximal high in the first principal component, second highest with the second principal component, and so on. The dispersion in each principal component, i , equals $\sqrt{\lambda_i}$. Suppose now that an eigenvalue λ_k is close to zero. This means that there is not much variance along that principal component at all and that the k th-coordinate of the vectors \mathbf{x}_i are close to zero in the transformed coordinate system. Thus, this dimension does not contribute much to the overall dispersion of the data and can be neglected without essential loss of information. Similarly, this holds for $j=k+1, \dots, p$ since we assumed the

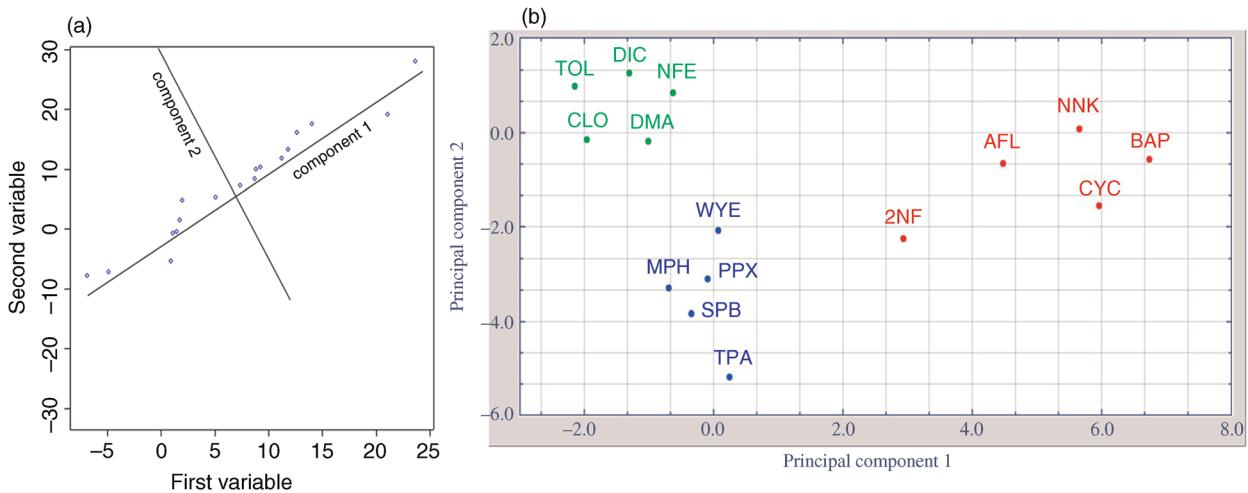


Figure 15.9 Principal component analysis. (a) A two-dimensional example of dimension reduction. The scatterplot shows highly correlated data samples that show significant variation with respect to the coordinate axes. Applying PCA will replace the original axes by principal components centered at the mean data vector whose directions determine the data variation. In the new coordinate system component 1 explains most of the data variation. (b) Practical example of PCA-based visualization of gene expression in chemically treated liver-like cells (genotoxic carcinogens in red, nongenotoxic carcinogens in blue, and noncarcinogens in green). Gene expression was measured with DNA arrays and a subset of profiles was preselected that separates the different chemicals. PCA allows the display using only two main directions and explaining 70% of the variance. Analysis was generated with *J-Express Pro* (Molmine, Bergen Norway). (Data was taken from Yildirimman et al., 2012.)

eigenvalues to be sorted in decreasing order. Thus, we have replaced the original p dimensions into $k < p$ dimensions that explain the relevant variance of the data sample.

An important question is how many principal components are needed to explain a sufficient amount of the data variance. Denote each vector by its coordinates $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^T$ and let \bar{x}_j be the j th coordinate of the mean vector $\bar{\mathbf{x}}$. A suitable measure for the total variance of the sample is the sum of the variances of the p coordinates given by

$$\sum_{j=1}^p \frac{1}{n-1} \sum_{i=1}^n (x_{ij} - \bar{x}_j)^2 = \text{Trace}(\mathbf{S}) = \sum_{j=1}^p \lambda_j. \quad (15.96)$$

Thus, for each $k < p$ the relative amount of variance explained by the first k principal components is given by

$$\frac{\sum_{j=1}^k \lambda_j}{\sum_{j=1}^p \lambda_j}. \quad (15.97)$$

In gene expression analysis, for instance, PCA is widely used to reduce the gene expression matrix or its transpose matrix, the condition expression matrix, to two or three reductions. Formula (15.97) is widely used in practice to characterize the computed dimension reduction. If the amount of variance explained is high then such a reduction makes sense otherwise the data set is too complex to be visualized with PCA.

Example 15.25 PCA Examples

Figure 15.9b shows a display of PCA using gene expression data for measuring response of liver-like cells after treatment with 15 chemicals that were classified into three classes (genotoxic carcinogens, nongenotoxic carcinogens, and noncarcinogens). PCA shows that the three classes of chemicals could be discriminated by their gene expression response. The selected first two principal components explain approximately 70% of the data variance indicating that the underlying dimension reduction is reasonable.

Markov processes, the probabilities for a system's future behavior are fully determined by the momentary state. Some random processes in continuous time and state space can be formally written as Langevin equations, which appear like ordinary differential equations with a noise term. Depending on the type of process – discrete or continuous – the state distribution, as a function of time, follows a Master equation or a Fokker–Planck equation.

15.4.1

Chance in Physical Theories

What is chance? If we call rolling a dice a “random experiment,” do we assume some essential randomness at work, like a physical force, or do we simply imply that predicting the outcome precisely would be too difficult? There are different views on this topic. Generally, the notion of randomness applies to all processes for which no causal explanation can be given, for whatever reason.

An *aleatoric* perspective on chance presupposes some fundamentally unpredictable processes in nature, for example, processes described by quantum mechanics, which then cause random behavior in the observable macroscopic world. In this view, probabilities can be inferred from frequencies in repeated experiments. While not being directly measurable, they are still a feature of objective reality – at least of what a positivist would call objective reality: the world of possible measurement results.

In the *epistemic perspective*, by contrast, probabilities describe our subjective degree of uncertainty about an existent, yet unknown reality. In this view, which is predominant in Bayesian statistics, we acknowledge that reality cannot be known completely and precisely, but only in terms of possibilities. Therefore, a model is not an exclusive, true description of the world, but has the status of a hypothesis among other possible ones. What we can do is delimit possibilities (e.g., rule out states of the world considered impossible) and weight them by probabilities. In this perspective, probabilities quantify subjective uncertainties and reflect what information is available based on data and prior knowledge.

The two philosophical notions of chance are expressed by the same mathematical formalism: probability theory, the theory of random variables and random processes. To describe a random experiment, we consider all of its possible outcomes and group them into sets called “random events.” These events are further quantified by weights called probabilities, which must satisfy the axioms of probability theory. In random processes, probabilities are not applied to single properties or variables, but to entire temporal histories of a system. Starting with statistical

15.4 Stochastic Processes

Summary

The microscopic motion of molecules and their fluctuations in numbers due to chemical reactions can be described by random processes. In a random process, a system moves randomly in some state space, where time and system states may be discrete or continuous. In

thermodynamics, chance has become a central element in many physical theories, and can be justified by a combination of arguments:

- 1) *Underlying microscopic dynamics.* If systems have many microscopic degrees of freedom, a deterministic simulation will be practically impossible. Even if microscopic states could be measured in theory, models describing them would not be tractable.
- 2) *Perturbations by the environment.* Systems are influenced by their environment, both on the macroscopic and microscopic level, in unknown ways. Thus, even if physical laws were fully deterministic, every system would still be coupled to a virtually infinite number of unknown degrees of freedom in its environment.
- 3) *Quantum-mechanical effects.* According to the positivist Copenhagen interpretation of quantum mechanics, any measurement process is a random experiment. In most cases, the precise values of quantum mechanical observables cannot be known in advance; their probabilities are given by the quantum mechanical wave function, which follows the Schrödinger equation, a deterministic wave equation. In a realist theory of quantum mechanics, called de Broglie–Bohm theory [4], quantum mechanical indeterminacy is explained by the lack of knowledge about the true states of particle and measurement apparatus. During measurement, particle and measurement apparatus become entangled by a joint Schrödinger wave function, which makes the measurement process appear like a random experiment. Both theories lead to the same results. In particular, they both predict that physical interactions between a system and its environment can lead to decorrelation, making large-scale quantum-mechanical systems behave, effectively, like classical systems.
- 4) *Deterministic chaos.* In some physical systems, the dynamics is such that minimal perturbations (in the initial conditions or in the course of the dynamics) suffice to drastically perturb later dynamic behavior. Even very small microscopic perturbations (caused by 1–3) suffice to make the system dynamics appear stochastic. Numerical errors in simulations would have similar effects. Precise predictions thus become practically impossible even for in principle deterministic systems.

When setting up biochemical models, we need to be aware that the way in which molecules move and react is affected by chance. Therefore, and since molecules, pathways, and cells are coupled to fluctuating environments, some parameters in our models may not be fixed, but described by random values or mathematical random processes.

15.4.2 Mathematical Random Processes

A random process (also called “stochastic process”) describes a system that moves unpredictably between the states in some state space [5,6]. The states may, for instance, represent the coordinates of a diffusing particle or the molecule numbers in a biochemical network. System states $x \in \mathcal{X}$ and time variable $t \in \mathcal{T}$ can be discrete (e.g., described by positive integer numbers) or continuous (i.e., described by real numbers). Formally, a random process X is a set of \mathcal{X} -valued random variables $X(t)$ indexed by a time variable $t \in \mathcal{T}$; the random variables for different time points follow a joint probability distribution. In practice, a random process can also be seen as a collection of possible histories of a system, called *realizations*, together with their probabilities. If time is

Example 15.26 Discrete Random Walk

A simple random process, the discrete random walk, is shown in Figure 15.10. The random walker moves on a one-dimensional grid; in each time step, he can stay at his position or make a random step to the left or right. Each step is chosen randomly with the probabilities

Step	Transition	Probability	
Right	$x \rightarrow x + 1$	q_+	(15.98)
Left	$x \rightarrow x - 1$	q_-	
Stay	$x \rightarrow x$	$1 - q_+ - q_-$	

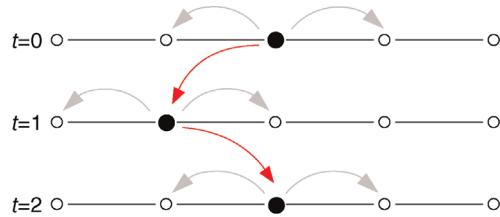


Figure 15.10 Random walk in discrete time and space. In each time step t , the walker can stay at its place or make a step left or right (possible steps shown in gray, actual steps in red).

and is independent of all previous steps. As an initial condition, we can specify a distribution $p(x, 0)$ for positions at time $t = 0$. If the walker starts precisely at position $x = 0$, this distribution reads $p(1, 0) = 1, p(x, 0) = 0$ for all $x \neq 0$. The initial distribution, together with the probabilities for individual steps, defines a joint distribution for all possible paths. To obtain an intuitive picture of the process, we may consider a large ensemble of random walkers, each taking different random decisions and thereby realizing different paths.

discrete ($t = 1, 2, \dots$), each realization consists of a series of states $\{x(t)\}$, characterized by a certain probability to occur in the process. For processes in continuous time $t \in R$, the realizations are functions $x(t)$, characterized by probability densities. *Stochastic simulations* generate realizations of a process, with frequencies reflecting their probabilities (for discrete processes) or probability densities (for continuous processes). As a form of Monte Carlo sampling, stochastic simulations can be used to compute statistical properties of a random process or to visualize its dynamics.

15.4.2.1 Reduced and Conditional Distributions

A random process defines a joint probability for the system states at all time points. If we pick n of the time points, the states $X(t_1), \dots, X(t_n)$ will form an n -dimensional random vector with joint probability $p(x_\omega, t_n; \dots; x_\alpha, t_1)$, where possible states are labeled by Greek subscripts. The system state at a single time point is described by a distribution $p(x, t)$, which can change in time. The *transition probability* $p(x_\alpha, t_2 | x_\beta, t_1)$ is the conditional probability to find the system in state x_α at t_2 if it was in state x_β at t_1 . With the formula for conditional probabilities

$$p(x_\alpha, t_2 | x_\beta, t_1) = \frac{p(x_\alpha, t_2; x_\beta, t_1)}{p(x_\beta, t_1)}, \quad (15.99)$$

the transition probability follows directly from the joint distribution at two time points t_1 and t_2 . Moreover, the transition probabilities obey a sum formula

$$p(x_\alpha, t_3 | x_\beta, t_1) = \sum_{x_\xi} p(x_\alpha, t_3 | x_\xi, t_2) p(x_\xi, t_2 | x_\beta, t_1), \quad (15.100)$$

called *Chapman–Kolmogorov equation*. The equation refers to a system that, on its way from x_β (at time t_1) to x_α (at time t_3), passes some intermediate state x_ξ (at time t_2); to obtain the total transition probability, we consider all possibilities for x_ξ , enumerate the corresponding paths, and sum over their probabilities. A random process is called *stationary* if the ensemble of realizations remains unchanged under a shift of the time variable $t \rightarrow t + \Delta t$. Each single realization, however, may still be time dependent. In stationary processes, the state distribution $p(x, t)$ is constant in time and the transition probabilities $p(x_\alpha, t_2 | x_\beta, t_1)$ depend only on time differences $t_2 - t_1$.

15.4.3

Brownian Motion as a Random Process

Macroscopically, substance diffusion is described by a time-dependent concentration profile following the diffusion equation (7.37). Initially concentrated in one point,

the concentration profile becomes dispersed and assumes the shape of a Gaussian distribution. Microscopically, diffusion is explained by Brownian motion, that is, a particle's movement caused by collisions with molecules of the surrounding liquid or gas. Brownian motion could be simulated by deterministic multiparticle simulations, but it is typically modeled as a random process. Since the movement takes place in continuous space and time, one often assumes a random process that produces continuous, but infinitely fine jittering movements.

Before we define this process formally, let us try to understand the path of a single diffusing particle. From the macroscopic concentration profiles, we know that a particle's position after a time interval Δt will be Gaussian-distributed. This end position is the starting point of a new diffusive movement in the following time interval, so we obtain a series of "snapshots" of the process, with Gaussian jumps in between. This process can be easily simulated: starting from an initial value $x(0) = 0$, we perform, for every time interval Δt , a jump $x(t + \Delta t) = x(t) + \sqrt{\Delta t} \eta$ with a standard Gaussian random number η (i.e., $\sqrt{\Delta t} \eta$ having the variance Δt). How will the variance of an ensemble of Brownian particles grow in time? Since the particles have no "memory," relative movements at different times are independent and their variances can simply be added ($\text{var}(A + B) = \text{var}(A) + \text{var}(B)$, as for any independent random variables A and B). Thus, the variance grows linearly, following exactly to the same law that we assumed to describe the random movement within each time step.

To get from this jump process to fine-grained, continuous particle paths, we further subdivide the time intervals. Instead of an interval of length $\Delta t = 1$ s and jumps of variance 1, we consider 100 intervals of length $\Delta t = 1/100$ s, each with a variance of 1/100. With this subdivision, we obtain the same distribution of end positions (Gaussian, with variance 1), but the path in between has a fine-grained shape. If we go on subdividing, then in the limit of infinitely fine subdivisions, we obtain a continuous random process called Wiener process.

Random models of Brownian motion exemplify the fact that mathematical models can display realistic behavior even if they employ unrealistic assumptions about details. On the one hand, we have Brownian motion as a physical phenomenon: whether it is truly random or a result of a very complicated, but deterministic, molecular dynamics, is unclear. On the other hand, we have several mathematical random processes with different combinations of discrete or continuous space or time, supposed to describe this motion. As examples, Figure 15.12 shows realizations of the discrete random walk (a), a random walk in discrete space and continuous time (b), and the Wiener process (c). In fact, the Wiener process (c) can be seen as

Example 15.27 The Wiener Process

The Wiener process $W(t)$ describes a continuous variable $w \in R$ in continuous time $t \in \mathbb{R}_+$ and is defined by the following properties:

- 1) $W(t)$ starts in the state $W(0) = 0$.
- 2) All realizations of $W(t)$ (except for a set of probability 0) are continuous functions of time t .
- 3) The increments $W(t_2) - W(t_1)$, for any time points $0 \leq t_1 \leq t_2$, follow a Gaussian distribution with mean 0 and variance $t_2 - t_1$.
- 4) Increments are independent of past increments, that is, for $t_1 \leq t_2 \leq t_3 \leq t_4$, the increments $W(t_2) - W(t_1)$ and $W(t_4) - W(t_3)$ are independent random variables.

Figure 15.11 illustrates the time behavior of the Wiener process: all realizations start at position $x(0) = 0$; at later time points, their distribution $p(x, t)$ becomes wider. Positions at different times are correlated: whenever the walker has reached a position x above average, it will tend to stay above average at later times (Figure 15.11d). The probability density for particle positions associated with the Wiener process follows from a Fokker–Planck equation (as described further below). With the initial position known precisely, we obtain for each spatial dimension a Gaussian-shaped probability distribution with a constant mean value and a variance proportional to time t , that is, a width $\sim \sqrt{t}$.

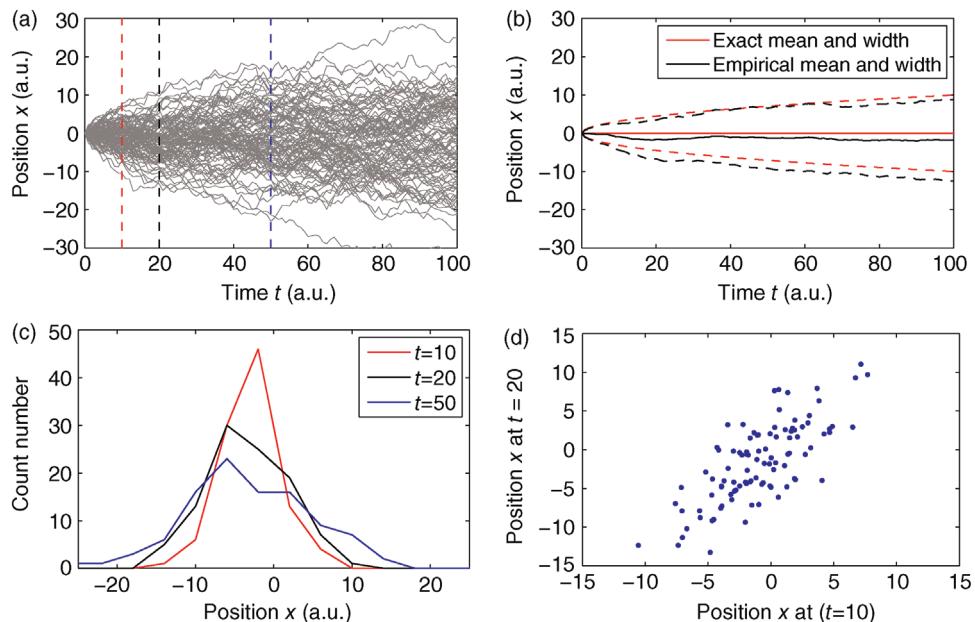


Figure 15.11 Statistical properties of the Wiener process. (a) 100 realizations of the Wiener process W . (b) Time-dependent mean values and standard deviations estimated from the realizations (black solid and dashed lines). Exact ensemble averages $\langle W(t) \rangle = 0$ and standard deviation $\sqrt{\langle W(t)^2 \rangle} = \sqrt{t}$ for the process are shown by red lines. (c) Empirical distribution of states at time points $t = 10$, $t = 20$, and $t = 50$ (dashed lines in (a)). The distribution for an infinitely large ensemble would be Gaussian-shaped. (d) A scatter plot for the states at $t = 10$ and $t = 20$ shows the temporal correlations.

a limiting case of the discrete random walk (a): starting from the discrete random walk, we can associate discrete space and time points with a grid in continuous space and time (interval sizes Δt and Δx). If we make the interval smaller and smaller, while keeping $\Delta t/\Delta x^2$ constant, the discrete random walk will become more fine-grained and, in the limit, approximate the Wiener process.

None of the processes in Figure 15.12 captures the molecular interactions between particles realistically; nevertheless, they all describe the long-term particle trajectories very well and in practically the same manner. They can do so because the diffusive motion, observed on larger space and time scales, is practically independent of the specific elementary steps assumed – as long as steps are random, of similar size, and independent of past

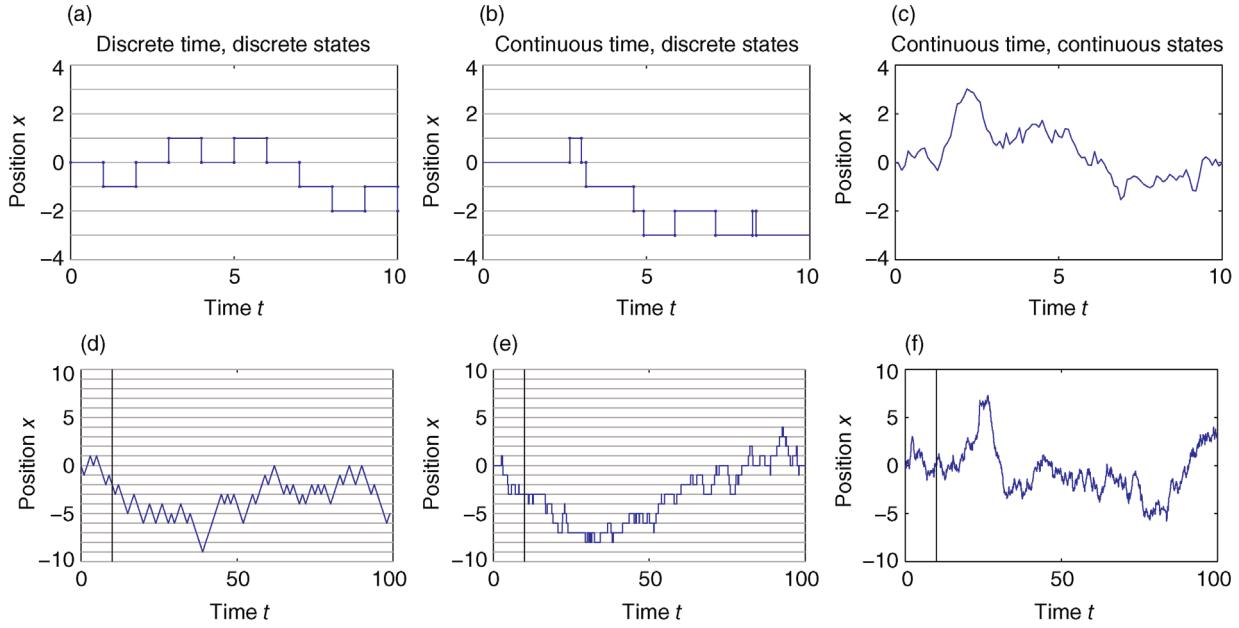


Figure 15.12 Three random processes describing particle motion. (a) Random walk with discrete space and time intervals $\Delta x = 1$ and $\Delta t = 1$ (see Figure 15.10). At each tick of the clock, the walker makes a random step up or down with probabilities $q_{\pm} = 1/2$. (b) In a random walk in continuous time, steps can occur at any moment and with a rate (i.e., probability per time) $w_{\pm} = 1/2$. (c) The Wiener process describes random motion in continuous space and time (approximation with small finite time intervals shown). The bottom panels (d), (e), and (f) show how the realizations continue on a longer time scale. Vertical lines indicate the detail shown in the top row pictures.

movements. According to the central limit theorem, these assumptions suffice to yield Gaussian long-term movements. This means that the relevance of the models does not lie in the microscopic mechanisms they assume, but in the long-term, stochastic behavior that emerges from them. Finally, this also explains why Brownian motion in water or air may look similar, even if the molecules interact very differently in both media. By contrast, convection would lead to qualitatively different movements because it violates the independence assumption.

15.4.4 Markov Processes

In dynamical models, the future behavior of a system may either follow from the current system state, or it may additionally depend on system states in the past. In the first case, the model can be called “memoryless” because it keeps no additional memory of the past except for the system state itself. An example is differential equation models of the form $dx/dt = f(x)$, in which the future behavior unfolds from the current state. Memory-carrying models, in contrast, can explicitly refer to past states in their equations. An example are delay differential equations (for an example, see Section 12.4.3), for example, $dx(t)/dt = f(x(t - \tau))$, in which the time derivatives at time t explicitly depend on system states at past time

points $t - \tau$. Delay differential equations can be used to collapse subsequent processes like transcription and translation into a single step, while accounting for the time lag between them. The time delay makes the system’s dynamics explicitly dependent on the system’s past and replaces variables like the mRNA levels, which exist in reality, but are omitted in this model. In this sense, the system state as represented by the model is an incomplete description of the real system.

A similar distinction between memoryless and memory-carrying models can be made for random processes. Of course, we will not assume that future behavior is fully predetermined by the current state. We can assume, however, that the *transition probabilities* between the present and future states – and therefore, the probabilities for future states – depend on the current state only, and not additionally on past states. Processes with this property, for example, the random walk models in Figure 15.12, are called *Markov processes*. Mathematically, Markov processes satisfy the condition

$$p(x_\alpha, t_{n+1} | x_\beta, t_n; \dots x_\omega, t_1) = p(x_\alpha, t_{n+1} | x_\beta, t_n) \quad (15.101)$$

for all time points $t_1 < \dots < t_n < t_{n+1}$. In other words: if state x_ω at time t_n is given, the states x_α at later times $t_{n+1} > t_n$ will be conditionally independent of all states earlier than t_n . Markov processes and memoryless deterministic models share the “completeness” assumption

that only system variables – and no variables from outside the system – affect the dynamics. Accordingly, a Markov process is fully specified by its initial probability distribution $p(x, t_0)$ and the transition probabilities $p(x_\alpha, t_2 | x_\beta, t_1)$ for all time points $t_1, t_2 > t_0$. Depending on the type of time and space variables – discrete or continuous – we distinguish between Markov chains, Markov jump processes, and continuous Markov processes.

15.4.5 Markov Chains

Markov processes in discrete time $t = 1, 2, \dots$ such as the discrete random walk (15.98), are called *Markov chains*. The states x can be discrete or continuous, but for simplicity we shall consider discrete states only. A Markov chain is determined by its initial probabilities $p(x, 0)$ and the transition probabilities $q_{\alpha\beta}(t) = p(x_\alpha, t + 1 | x_\beta, t)$. Here we assume that the $q_{\alpha\beta}$ are constant in time.

Realizations $\{x(t)\}$ of a Markov chain can be obtained by stochastic simulation: we first draw an initial state $x(0)$ from the distribution $p(x_\alpha, 0)$. Then we jump from state to state, choosing each state $x(t+1)$ randomly and with probabilities $p(x_\alpha, t + 1 | x_\beta(t), t)$ based on the previous state $x_\beta(t)$. A similar iteration procedure yields the time-dependent distribution $p(x, t)$: starting from the initial distribution $p(x, 0)$, we iteratively compute all later distributions with the formula

$$\begin{aligned} p(x_\alpha, t + 1) &= \sum_{\beta} q_{\alpha\beta} p(x_\beta, t) \\ &= q_{\alpha\alpha} p(x_\alpha, t) + \sum_{\beta \neq \alpha} q_{\alpha\beta} p(x_\beta, t). \end{aligned} \quad (15.102)$$

The two terms describe how a system can reach state x_α at time $t + 1$: at time t , the system was either in the same state x_α (conditional probability to stay in state α : $q_{\alpha\alpha} = p(x_\alpha, t + 1 | x_\alpha, t)$) or in a different state β (conditional probability for a jump from x_β to x_α : $q_{\alpha\beta} = p(x_\alpha, t + 1 | x_\beta, t)$). The transition probabilities $q_{\alpha\beta}$ have to be normalized to $\sum_{\alpha} q_{\alpha\beta} = 1$ or otherwise the new probabilities would not sum up to 1. Equation (15.102) can also be written in vector notation: with a vector $\mathbf{p}(t)$ containing the probabilities $p(x_\alpha, t)$ at time t , the iteration formula reads

$$\mathbf{p}(t + 1) = \mathbf{Q}\mathbf{p}(t). \quad (15.103)$$

To compute the distribution $\mathbf{p}(t)$ at time t (with a constant transition matrix \mathbf{Q}), we apply the formula recursively to the initial distribution $p(x, 0)$ and obtain $\mathbf{p}(t) = \mathbf{Q}^t \mathbf{p}(0)$. The *invariant distribution* \mathbf{p}_∞ is given by an eigenvector of \mathbf{Q} with the eigenvalue 1, satisfying $\mathbf{p}_\infty = \mathbf{Q}\mathbf{p}_\infty$. Once reached, the invariant distribution $\mathbf{p}(0) = \mathbf{p}_\infty$ remains constant in time. In this stationary

ensemble, the individual realizations of the process still describe random movements, but the probability flows between different states are balanced and the state distribution remains constant.

Example 15.28 Probability Distribution in the Discrete Random Walk

The distribution $p(x, t)$ of particle positions in the discrete random walk can be computed by enumerating all possible paths between position 0 (at time 0) and position x (at time t). If steps are symmetric and obligatory, $q_{\pm} = 1/2$, all paths have identical probabilities $(1/2)^t$. At time point t , the walker has made t steps, among which there are Y steps to the right. The random number Y follows a binomial distribution $p(y, t) = \binom{t}{y} (1/2)^t$ and determines the current position $X = 2Y - t$. The current position is restricted to even positions x at even time points and odd positions x at odd time points. Large distances $|x|$ are improbable and most possible paths remain close to the starting point. Transition probabilities can be computed in a similar way: the probability to move from x_β (at time t_1) to x_α (at time t_2) depends only on the differences $t_2 - t_1$ and $x_\alpha - x_\beta$ (because the probabilities for relative steps do not depend on absolute time or position). The transition probabilities are given by $p(x_\alpha, t_2 | x_\beta, t_1) = p(x_\alpha - x_\beta, t_2 - t_1)$.

15.4.6 Jump Processes in Continuous Time

In some stochastic models, for example, population models for animals with yearly reproduction cycles, discrete times – in this case, time steps of years – is a natural assumption. Chemical reaction systems, however, have no natural time steps and are typically described by *Markov jump processes* in continuous time. Like in the process in Figure 15.12b and e, states x are discrete, but state transition can occur at any time $t \in \mathbb{R}$.

15.4.6.1 Deriving the Master Equation

To describe discrete state transitions in continuous time, we consider small time intervals $[t, t + \Delta t]$ and assess the probability for transitions within one interval. For very small intervals $\Delta t \rightarrow 0$, the probability for a transition is proportional to Δt and multiple transitions are so unlikely that they can be neglected. The probability for a transition $x_\beta \rightarrow x_\alpha$ can be approximated by $w_{\alpha\beta}(t)\Delta t$ with *transition rates* $w_{\alpha\beta}(t)$, which are assumed to be constant or to be computable from the system state. Diagonal elements $w_{\alpha\alpha}(t)$ in the transition rate matrix describe the decreasing probability of state α due to transitions away

from this state. Due to the transitions, the state probabilities $p(x, t)$ will change in time. This is described by a differential equation system called *Master equation*. To derive it, we consider two time points $t_1 < t_2$ and first note that $p(x_\alpha, t_2; x_\beta, t_1) = p(x_\alpha, t_2 | x_\beta, t_1)p(x_\beta, t_1)$. We then compute

$$\begin{aligned} p(x_\alpha, t_2) &= p(x_\alpha, t_2; x_\alpha, t_1) + \sum_{\beta \neq \alpha} p(x_\alpha, t_2; x_\beta, t_1) \\ &= \left(1 - \sum_{\beta \neq \alpha} p(x_\beta, t_2 | x_\alpha, t_1)\right)p(x_\alpha, t_1) \\ &\quad + \sum_{\beta \neq \alpha} p(x_\alpha, t_2 | x_\beta, t_1)p(x_\beta, t_1), \end{aligned} \quad (15.104)$$

and set $t_1 = t$ and $t_2 = t + \Delta t$. For small time differences $\Delta t = t_2 - t_1$, we can approximate $p(x_\alpha, t + \Delta t | x_\beta, t) \approx w_{\alpha\beta}(t)\Delta t$ and obtain

$$p(x_\alpha, t + \Delta t) \approx p(x_\alpha, t) + \left[\sum_{\beta \neq \alpha} w_{\alpha\beta}(t)p(x_\beta, t) - w_{\beta\alpha}(t)p(x_\alpha, t) \right] \Delta t. \quad (15.105)$$

In the limit $\Delta t \rightarrow 0$, this yields the Master equation

$$\frac{dp(x_\alpha, t)}{dt} = \sum_{\beta \neq \alpha} [w_{\alpha\beta}(t)p(x_\beta, t) - w_{\beta\alpha}(t)p(x_\alpha, t)], \quad (15.106)$$

for the time-dependent probability of state x_α . The terms in the bracket can be seen as probability flows between different states: the positive term captures all transitions from other states to x_α , while the negative term captures all transitions from x_α to other states.

In practice, the number of states, and thus the number of equations in a system, can be very large. In a chemical system with m molecule species, each described by particle numbers between 0 and n , and states describing the molecule numbers, we obtain m^{n+1} possible states, each requiring its own differential equation. While the Master equation for systems of monomolecular reactions may be solved analytically, its solution for realistic biological models can be hard or impossible [7]. However, the behavior of the random process may still be computed by stochastic simulation or, in some cases, from moment-generating functions (see Section 15.4.8).

15.4.7 Continuous Random Processes

15.4.7.1 Langevin Equations

Continuous Markov processes like the Wiener process can be used to model Brownian motion and the dynamics of chemical systems described by real-valued substance levels. The *Langevin equation*

$$dX(t) = f(X(t))dt + \sigma dW(t) \quad (15.107)$$

describes a continuous Markov process, where $dX(t)$ is the stochastic time increment of the process itself, $dW(t)$ is the time increment of the Wiener process, and σ is a scaling factor. The equation can be formally written as

$$\frac{dX(t)}{dt} = f(X(t)) + \sigma \xi(t), \quad (15.108)$$

which appears like an ordinary differential equation with a stochastic noise term $\xi(t)$. This noise term is given by *Gaussian white noise*, a hypothetical random process with mean value 0 and the covariance function

$$\text{cov}(\xi(t_1), \xi(t_2)) = \langle \xi(t_1)\xi(t_2) \rangle = \delta(t_1 - t_2). \quad (15.109)$$

Even in the form (15.108), a Langevin equation does not refer to a mathematical function, but to a random process. The equivalence of Eqs. (15.107) and (15.108) relies on the fact that the Wiener process can be seen as a time integral over the white noise process: accordingly, the Wiener process $W(t)$ solves the simple Langevin equation

$$\frac{dW(t)}{dt} = \xi(t). \quad (15.110)$$

The fact that white noise, integrated over time, yields Gaussian-distributed random increments allows us to simulate Langevin equations by the *stochastic Euler method*: as in the simulation of Brownian motion in Section 15.4.3, the Langevin Eq. (15.108) can be approximated by a time-discrete process

$$X(t_{i+1}) \approx X(t_i) + f(X(t_i))\Delta t + \sigma \Delta W, \quad (15.111)$$

in which each time step corresponds to a fixed interval Δt . The random term ΔW , as the time increment of a Wiener process after time Δt , is a Gaussian random number with mean 0 and variance Δt . Equation (15.111) approximates the term $f(X(t))$ by a constant value within each time interval and is therefore not exact; however, its accuracy can be improved by choosing small interval lengths Δt .

15.4.7.2 The Fokker–Planck Equation

What would be the equivalent of a Master equation for continuous Markov processes or, in other words, how will the probability density $p(x, t)$ of a continuous random process $X(t)$ change over time? For a Langevin equation with system states $x \in R^n$, this question is answered by the *Fokker–Planck equation*

$$\frac{\partial p(x, t)}{\partial t} = \left[- \sum_{i=1}^n \frac{\partial}{\partial x_i} g_i(x, t) + \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2}{\partial x_i \partial x_j} D_{ij}(x, t) \right] p(x, t), \quad (15.112)$$

a partial differential equation specified by a drift vector $g(x)$ and a diffusion tensor $D(x)$. The two quantities stem from terms in the Langevin equation: for instance,

consider a Langevin equation for a vectorial process $X(t)$

$$\frac{dX(t)}{dt} = \mathbf{f}(X(t), t) + \mathbf{B}(X(t), t)\xi(t), \quad (15.113)$$

with a vector $\mathbf{f}(\mathbf{x}, t)$, a matrix $\mathbf{B}(\mathbf{x}, t)$, and uncorrelated white noise inputs $\xi_i(t)$ satisfying $\langle \xi_i(t_1)\xi_k^T(t_2) \rangle = \delta_{ik}\delta(t_2 - t_1)$. The resulting probability density $p(\mathbf{x}, t)$ will be governed by a Fokker–Planck equation (Eq. 15.112) with drift and diffusion terms

$$\begin{aligned} \mathbf{g}(\mathbf{x}, t) &= \mathbf{f}(\mathbf{x}, t) \\ \mathbf{D}(\mathbf{x}, t) &= \frac{1}{2} \mathbf{B}(\mathbf{x}, t)\mathbf{B}^T(\mathbf{x}, t). \end{aligned} \quad (15.114)$$

If we imagine the probability distribution $p(\mathbf{x}, t)$ as a cloud in state space, the cloud center will move in time according to the deterministic part of the Langevin equation, while the width of the cloud will increase according to the noise term. For simple Brownian motion in one dimension (Wiener process for a particle coordinate X), the Fokker–Planck equation reads

$$\frac{\partial}{\partial t} p(x, t) = D \frac{\partial^2}{\partial x^2} p(x, t), \quad (15.115)$$

with diffusion constant D and a vanishing drift term. If we consider many such particles, their concentration profile in space will reflect the single-particle probability. Therefore, Eq. (15.115) is closely related to the diffusion equation (7.37) for substrate concentrations.

15.4.8

Moment-Generating Functions

If a state space is very large, the Master equation cannot be integrated numerically. Sometimes, however, solutions $p(x, t)$ or their statistical moments can be obtained from *moment-generating functions*. Generally, a probability distribution $p(x)$ can be described by its *statistical moments* $\mu_n = \langle X^n \rangle$. The first moment $\langle X \rangle$ is the expectation value, the second moment $\langle X^2 \rangle$ determines the variance $\text{var}(X) = \langle X^2 \rangle - \langle X \rangle^2$, and higher moments specify the shape of the distribution more precisely. For a discrete random variable with values $n \in N$ and probabilities p_n , the generating function $G(s)$ is defined by the power series

$$G(s) = \sum_n p_n s^n, \quad (15.116)$$

in which the probabilities p_n appear as coefficients.

The function argument s has no direct interpretation. However, if a generating function is known, its derivatives at $s = 1$ allow us to compute the statistical moments of

Example 15.29 Generating Functions for Binomial Distribution and Poisson Distribution

The generating function of the binomial distribution

$$p_n = \binom{N}{n} q^n (1-q)^{N-n}$$

reads

$$G(s) = \sum_n \binom{N}{n} q^n (1-q)^{N-n} s^n = (sq + (1-q))^N.$$

The generating function of the Poisson distribution

$$p_n = \frac{\lambda^n}{n!} e^{-\lambda}$$

is given by $G(s) = e^{(s-1)\lambda}$.

our random variable, for instance

$$\begin{aligned} G(s)|_{s=1} &= \sum_n p_n = 1 = \mu_0, \\ \frac{dG(s)}{ds}|_{s=1} &= \sum_n p_n n s^{n-1}|_{s=1} = \sum_n n p_n = \langle n \rangle = \mu_1, \\ \frac{d^2 G(s)}{ds^2}|_{s=1} &= \sum_n p_n n(n-1) s^{n-2}|_{s=1} = \mu_2 - \mu_1, \end{aligned} \quad (15.117)$$

and so on. Moreover, the probabilities p_n follow from the derivatives of G at $s = 0$,

$$p_n = \frac{1}{n!} \frac{d^n G}{ds^n}|_{s=0}, \quad (15.118)$$

In order to study random processes, we simply consider the generating function of the time-dependent state distribution $p(x, t)$.

15.5

Control of Linear Dynamical Systems

Summary

Linear control theory describes how linear dynamical systems respond to external perturbations and how their dynamics can be stabilized or controlled. Linear differential equation systems can be characterized by their responses to pulse-like or harmonic input signals, called impulse response and frequency response functions. Temporal random signals, which can be characterized by mean values and spectral densities, are another possible type of inputs. In linear systems, the spectral density of

Example 15.30 Moment-Generating Function for Degradation Process

Consider a random model for the degradation of molecules with discrete molecule numbers n and an initial number n_0 (compare Section 7.3). With a constant degradation rate w per molecule, we obtain the propensity $a^-(n) = wn$ of stochastic degradation events, and the Master equation reads

$$\frac{dp(n,t)}{dt} = w(n+1)p(n+1,t) - wnp(n,t).$$

From the time-dependent distribution $p(n,t)$, we can construct the generating function

$$G(s,t) = \sum_n p(n,t)s^n.$$

Its time evolution reads

$$\frac{\partial G}{\partial t} = \sum_{n=0}^{n_0} \frac{dp(n,t)}{dt} s^n = w \sum_{n=0}^{n_0} [(n+1)p(n+1,t) - np(n,t)]s^n,$$

where we used $p_{n_0+1}(t) = 0$. Rewriting the two terms in brackets as

$$(n+1)p_{n+1}(t)s^n = \frac{\partial}{\partial s} p(n+1,t)s^{n+1}$$

$$np(n,t)s^n = s \frac{\partial}{\partial s} np(n,t)s^n$$

leads to a partial differential equation for $G(s,t)$

$$\frac{\partial G}{\partial t} = w \left[\frac{\partial G}{\partial s} - s \frac{\partial G}{\partial s} \right] = w(1-s) \frac{\partial G}{\partial s}.$$

From the initial distribution (with exactly n_0 molecules present at $t = 0$)

$$p(n,0) = \begin{cases} 1 & : n = n_0 \\ 0 & : n \neq n_0 \end{cases},$$

we obtain the initial condition $G(s,0) = s^{n_0}$, and the solution of Eq. (15.119) with this initial condition is given by

$$G(s,t) = [1 - (1-s)e^{-wt}]^{n_0}.$$

The generating function determines the probability distribution for all times t . The derivatives at $s = 1$ determine the mean value and variance

$$\langle N(t) \rangle = n_0 e^{-wt}$$

$$\text{var}(N(t)) = n_0 e^{-wt} (1 - e^{-wt}),$$

while the distribution itself is given by

$$p(n,t) = \binom{n_0}{n} (1 - e^{-wt})^{n_0-n} e^{-nwt}$$

$$= \binom{n_0}{n} (1 - q(n))^{n_0-n} q(t)^n.$$

For each time point t , this yields a binomial distribution with the time-dependent parameter $q(t) = e^{-wt}$.

output signals can be directly obtained from the spectral density of the input and from the system's frequency response. Control theory also provides efficient methods for the reduction and optimal control of linear systems.

15.5.1 Linear Dynamical Systems

Control theory is concerned with the question of steering dynamical systems by external interventions, for instance, in order to stabilize their dynamics against external perturbations. An important application is the design of feedback systems that keep system close to optimal states. We consider linear differential systems of the form

$$\begin{aligned} \frac{dx(t)}{dt} &= Ax(t) + Bu(t), \\ y(t) &= Cx(t) + Du(t), \end{aligned} \quad (15.119)$$

specified by constant matrices A , B , C , and D . In the model – just like in real systems – state variables are not necessarily visible from the outside, but they can be observed via an output vector y and be influenced via an input vector u . In the system Eq. (15.119), changes of the state variables x_i depend on the state variables themselves and on the input values u_j . The output variables y_l are linear combinations of state variables x_i and inputs u_j (see Figure 15.13).

Linear systems like Eq. (15.119) can be obtained by linearizing a nonlinear kinetic model around its steady state. The matrices A , B , C , D will result from linear approximations of the local dynamics (see Section 7.3). Particular solutions of the system depend on the initial condition $x(0) = x_0$ and on the external input $u(\cdot)$. Here the symbol $f(\cdot)$ denotes a mathematical function, in contrast to a particular function value $f(t)$ at time t . If the input variables vanish (constant input $u = 0$), the system (15.119) has a fixed point at $x = 0$. By adding a constant term in the differential equation, the fixed point could be shifted to other values, for instance positive values representing the steady state of a metabolic model. We usually assume that the system (15.119) is asymptotically stable,

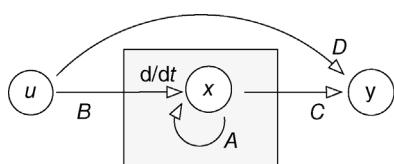


Figure 15.13 Linear dynamical system. According to Eq. (15.119), the time derivative of the internal state vector x depends on x itself and on an input vector u . The input can represent external regulation or random perturbations. The output vector y , observable from the outside, is computed from x and u . The system is specified by matrices A , B , C , and D .

that is, all eigenvalues of \mathbf{A} have strictly negative real parts.

15.5.2

System Response and Linear Filters

The temporal output $\mathbf{y}(\cdot)$ of a linear system (15.119) depends on two factors: on the initial values \mathbf{x}_0 and on the temporal input $\mathbf{u}(\cdot)$. Due to linearity, the general dynamics can be split in two terms: (i) a behavior given $\mathbf{x}(0)$, but assuming a vanishing input $\mathbf{u}(\cdot) = 0$; and (ii) a behavior arising from $\mathbf{x}(0) = 0$ and a given time course $\mathbf{u}(\cdot)$. In particular, if we set $\mathbf{x}(0) = 0$, the system (15.119) maps each input time course $\mathbf{u}(\cdot)$ to an output time course $\mathbf{y}(\cdot)$. This *input–output relation* is completely determined by the matrices \mathbf{A} , \mathbf{B} , \mathbf{C} , and \mathbf{D} . For simplicity, we assume that $\mathbf{x}(0) = 0$ and $\mathbf{D} = 0$ unless otherwise specified.

For the linear system (15.119) with $\mathbf{x}(0) = 0$, the responses to different input signals are additive, so we can linearly combine them. If the system maps an input $\mathbf{u}^{(1)}(\cdot)$ to an output $\mathbf{y}^{(1)}(\cdot)$ and an input $\mathbf{u}^{(2)}(\cdot)$ to $\mathbf{y}^{(2)}(\cdot)$, it will also map the linear combination $\alpha\mathbf{u}^{(1)}(\cdot) + \beta\mathbf{u}^{(2)}(\cdot)$ to the combination $\alpha\mathbf{y}^{(1)}(\cdot) + \beta\mathbf{y}^{(2)}(\cdot)$. Response to complicated inputs can be combined from responses to simpler inputs. It is therefore helpful to know the responses to simple input functions, namely pulses and harmonic oscillations.

15.5.2.1 Impulse Input

The response to a very short input pulse is described by the *impulse response function*. Assume that an input variable u_j shows an infinitesimally short pulse at $t = 0$ with a time integral of 1, while all other input variables vanish. Such a pulse can be represented by a Gaussian function of width ϵ , centered at $t = 0$. In the limit $\epsilon \rightarrow 0$, this function approximates a Dirac delta distribution $\delta(t)$ and the system's response (with initial condition $\mathbf{x}(0) = 0$ and $\mathbf{D} = 0$) is given by the *impulse response function* (also called kernel or Green function)

$$\mathbf{K}(t) = \mathbf{C}e^{\mathbf{A}t}\mathbf{B}, \quad (15.120)$$

where the matrix exponential is defined by a Taylor series

$$e^{\mathbf{A}t} = \mathbf{I} + \mathbf{A}t + \frac{1}{2}(\mathbf{A}t)^2 + \dots \quad (15.121)$$

If the impulse response (15.120) is known, system responses to bounded integrable inputs $\mathbf{u}(\cdot)$ of arbitrary shape (and starting at $t = -\infty$) can be written as a convolution integral

$$\mathbf{y}(t) = \int_{-\infty}^t \mathbf{K}(t-t')\mathbf{u}(t')dt'. \quad (15.122)$$

15.5.2.2 Oscillatory Input

As a second case, we consider an oscillatory input

$$\mathbf{u}(t) = \tilde{\mathbf{u}}e^{i\omega t} \quad (15.123)$$

with circular frequency $\omega = 2\pi/T$ (where T is the oscillation period) and $t \in \mathbb{R}$. The input (15.123) will induce forced oscillations

$$\mathbf{x}(t) = \tilde{\mathbf{x}}e^{i\omega t}, \quad \mathbf{y}(t) = \tilde{\mathbf{y}}e^{i\omega t} \quad (15.124)$$

of the same frequency ω . The complex amplitudes in the vectors $\tilde{\mathbf{u}}$, $\tilde{\mathbf{x}}$, and $\tilde{\mathbf{y}}$ describe amplitudes and phases of the oscillating variables. We can compute $\tilde{\mathbf{y}}$ in two ways: First, using (15.122), we obtain

$$\mathbf{y}(t) = \int_{-\infty}^t \mathbf{K}(t-t')\mathbf{u}e^{i\omega t'}dt', \quad (15.125)$$

and by a substitution $\tau = t - t'$

$$\mathbf{y}(t) = \int_0^\infty \mathbf{K}(\tau)\mathbf{u}e^{i\omega(t-\tau)}d\tau. \quad (15.126)$$

Then, we have

$$\mathbf{y}(t) = \left(\int_0^\infty \mathbf{K}(\tau)e^{-i\omega\tau}d\tau \right) \mathbf{u}e^{i\omega t}. \quad (15.127)$$

Second, by inserting Eq. (15.124) as an ansatz into the system Eq. (15.119) with $\mathbf{D} = 0$, we obtain

$$\tilde{\mathbf{y}} = \mathbf{C}(i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}\tilde{\mathbf{u}}. \quad (15.128)$$

A comparison between Eqs. (15.127) and (15.128) shows that the matrix

$$\mathbf{H}(i\omega) = \mathbf{C}(i\omega\mathbf{I} - \mathbf{A})^{-1}\mathbf{B}, \quad (15.129)$$

called *frequency response function*, is exactly the Fourier transform of the impulse response function

$$\mathbf{H}(i\omega) = \int_0^\infty \mathbf{K}(\tau)e^{-i\omega\tau}d\tau. \quad (15.130)$$

The convolution integral (15.122) describes the input–output relation for temporal signals; in frequency space, these signals are described by their Fourier transforms. This allows us to avoid the complicated calculation of the convolution integral. Since convolutions in the time domain correspond to multiplications in the frequency domain, we can compute the Fourier components of $\mathbf{y}(\cdot)$ by multiplying the Fourier components of \mathbf{u} with the frequency response function

$$\tilde{\mathbf{y}}(\omega) = \mathbf{H}(i\omega)\tilde{\mathbf{u}}(\omega). \quad (15.131)$$

This is exactly Eq. (15.128), and the Fourier transform of Eq. (15.122). Equation (15.131) has a simple interpretation: a time-invariant linear system can be seen as a *filter*

that specifically dampens or amplifies Fourier components in a signal.

15.5.3 Random Fluctuations and Spectral Density

To see how linear systems respond to random noise inputs, we choose stationary Gauss–Markov random processes as an input signals \mathbf{u} (see Section 15.4.4). An example of such processes is white noise, which also appears in the chemical Langevin equation (see Section 7.2.4). We specify $\mathbf{x}(t_0) = 0$ at time point t_0 as an initial condition and consider the limit $t_0 \rightarrow -\infty$. With a random input \mathbf{u} , the output \mathbf{y} will also follow a random process. Both processes can be described by mean values and cross-correlations or, alternatively, by their spectral density functions.

By averaging over all realizations of a random process, we obtain its ensemble average $\langle x(t) \rangle$. If the process is stationary, as we assume here, this average is constant in time. Individual realizations will deviate from this average behavior: the deviations $\Delta x(t) = x(t) - \langle x(t) \rangle$ describe fluctuations in time for a single realization. However, in the ensemble of realizations, $\Delta x(t)$, for each time point t , becomes a random variable, and deviations $\Delta x(t)$ at time points t and $t + \tau$ will be correlated. In a stationary process, these correlations only depend on the time difference τ . We can quantify the strength of fluctuations by the matrix-valued covariance function

$$\mathbf{C}_x(\tau) = \langle \Delta \mathbf{x}(\tau) \Delta \mathbf{x}^T(0) \rangle. \quad (15.132)$$

Again, the process can be described in the frequency domain: the Fourier transform of the covariance function,

$$\Phi_x(\omega) = \int_{-\infty}^{\infty} \mathbf{C}_x(\tau) e^{-i\omega\tau} d\tau, \quad (15.133)$$

called *spectral density matrix*, describes our random process in terms of fluctuations at different frequencies. The covariance function can be reobtained from the spectral density by an inverse Fourier transformation

$$\mathbf{C}_x(\tau) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{i\omega\tau} \Phi_x(\omega) d\omega, \quad (15.134)$$

and the stationary covariance

$$\mathbf{C}_x(0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \Phi_x(\omega) d\omega \quad (15.135)$$

contains the noise contributions of all frequencies.

For a linear model (15.119) with random input $\mathbf{u}(\cdot)$, the frequency spectra of input and output variables are given

by spectral density matrices $\Phi_u(\omega)$ and $\Phi_y(\omega)$, and so are the correlations between them (for an example, see Section 9.4.3). Again, the system acts as a linear filter: the spectral density of the outputs can be computed by the formula

$$\Phi_y(\omega) = \mathbf{H}(i\omega) \Phi_u(\omega) \mathbf{H}(i\omega)^\dagger \quad (15.136)$$

from the frequency response function and from the spectral density of the inputs. The symbol \dagger denotes the matrix adjoint (i.e., the conjugate transpose). If a system is driven by white noise (with spectral density $\Phi_u(\omega) = 1$), the output spectral density $\Phi_y(\omega) = \mathbf{H}(i\omega) \mathbf{H}(i\omega)^\dagger$ is directly determined by the frequency response function.

15.5.4 The Gramian Matrices

To control a linear system by a feedback controller, two central questions must be addressed: to what extent can the system be steered by manipulating its inputs? And, to what extent can the system's state \mathbf{x} be inferred from observations of its outputs? Ideally, a system should be both controllable and observable. A system is *controllable* if it can be steered from any initial state $\mathbf{x}^{(0)}$ to any final state $\mathbf{x}^{(1)}$ within some finite time interval by an appropriate input signal $\mathbf{u}(\cdot)$. A system is *observable* if its input values $\mathbf{u}(\cdot)$ and output values $\mathbf{y}(\cdot)$, observed in a time interval $[t_0, t_1]$ provide sufficient information to infer the initial internal state $\mathbf{x}(t_0)$.

For the system (15.119), controllability and observability are determined by two matrices, the *controllability Gramian* \mathbf{W}_c and the *observability Gramian* \mathbf{W}_o :

$$\begin{aligned} \mathbf{W}_c(t_1) &= \int_0^{t_1} e^{At} \mathbf{B} \mathbf{B}^T e^{A^T t} dt, \\ \mathbf{W}_o(t_1) &= \int_0^{t_1} e^{A^T t} \mathbf{C}^T \mathbf{C} e^{At} dt. \end{aligned} \quad (15.137)$$

The Gramian matrices describe which internal variables (or which linear combinations of them) can be controlled and observed, and to what extent. A system is controllable if $\mathbf{W}_c(t)$ is regular (i.e., invertible) for some (and hence any) time $t > 0$. Likewise, a system is observable if $\mathbf{W}_o(t)$ is regular for some (and hence any) time $t > 0$. The Gramian matrices are related to the so-called input and output energies. A given input signal is characterized by an *input energy*

$$E_u = \int_0^{t_1} \mathbf{u}^T(t) \mathbf{u}(t) dt. \quad (15.138)$$

In optimal control problems, the input energy can be seen as an effort for steering the system. Suppose that a system should be steered from state $\mathbf{x}(0) = 0$ to the state

$\mathbf{x}(t_1) = \mathbf{x}_1$ with a minimal energy effort. If the matrix \mathbf{W}_c is invertible, the optimal input is given by

$$\mathbf{u}(t) = \mathbf{B}^T e^{A^T(t_1-t)} \mathbf{W}_c^{-1}(t_1) \mathbf{x}_1, \quad (15.139)$$

and the necessary input energy is $\mathbf{x}_1^T \mathbf{W}_c^{-1}(t_1) \mathbf{x}_1$. In analogy, the *output energy*

$$E_y = \int_0^{t_1} \mathbf{y}^T(t) \mathbf{y}(t) dt \quad (15.140)$$

scores the output variables of a system that starts at $\mathbf{x}(0) = \mathbf{x}_1$ and has a vanishing input $\mathbf{u}(t) = 0$. The output energy can be written as $\mathbf{x}^T \mathbf{W}_o(t_1) \mathbf{x}$. For asymptotically stable systems, the infinite-time Gramian matrices \mathbf{W}_c and \mathbf{W}_o are defined by

$$\begin{aligned} \mathbf{W}_c &= \int_0^\infty e^{At} \mathbf{B} \mathbf{B}^T e^{A^T t} dt, \\ \mathbf{W}_o &= \int_0^\infty e^{A^T t} \mathbf{C}^T \mathbf{C} e^{At} dt, \end{aligned} \quad (15.141)$$

and can be computed from the Lyapunov equations

$$\begin{aligned} \mathbf{A} \mathbf{W}_c + \mathbf{W}_c \mathbf{A}^T + \mathbf{B} \mathbf{B}^T &= 0, \\ \mathbf{A}^T \mathbf{W}_o + \mathbf{W}_o \mathbf{A} + \mathbf{C}^T \mathbf{C} &= 0. \end{aligned} \quad (15.142)$$

The infinite-time controllability Gramian \mathbf{W}_c has a simple interpretation: it represents the covariance matrix of the internal states \mathbf{x} that would result from uncorrelated white-noise inputs $u_j(t) = \xi_j(t)$.

15.5.5 Model Reduction

If a system contains more internal state variables than input and output variables, there may be ways to reduce it to a smaller dimensionality $r \ll n$ while preserving the input–output relation to a good approximation. Various algorithms for such model reduction have been devised, for instance to reduce the large equation systems used to simulate electric circuits in computer chips. In linear model reduction, the dynamical system (15.119) is replaced by a lower-dimensional system with an r -dimensional state vector \mathbf{z}

$$\begin{aligned} \frac{d\mathbf{z}}{dt} &= \mathbf{A}' \mathbf{z} + \mathbf{B}' \mathbf{u}, \\ \mathbf{y}' &= \mathbf{C}' \mathbf{z} + \mathbf{D}' \mathbf{u}. \end{aligned} \quad (15.143)$$

The new variables (or “modes”) z_m are linear combinations $z_m = \sum_i T_{mi} x_i$ of the original variables x_i . If the transformation matrix \mathbf{T} is invertible, the original and the transformed system are equivalent and the input–output relation of Eq. (15.1) is exactly preserved. In this case, the state vector \mathbf{x} is transformed to a vector $\mathbf{z} = \mathbf{T}\mathbf{x}$ with the same size and the transformed system matrices read

$\mathbf{A}' = \mathbf{T}\mathbf{A}\mathbf{T}^{-1}$, $\mathbf{B}' = \mathbf{T}\mathbf{B}$, $\mathbf{C}' = \mathbf{C}\mathbf{T}^{-1}$, $\mathbf{D}' = \mathbf{D}$. Using such transformations as a starting point, we can reduce a system to a lower dimensionality r : We split

$$\mathbf{T} = \begin{pmatrix} \mathbf{T}_1 \\ \mathbf{T}_2 \end{pmatrix}, \quad \mathbf{T}^{-1} = (\mathbf{S}_1 \mathbf{S}_2) \quad (15.144)$$

with an $r \times n$ matrix \mathbf{T}_1 and an $n \times r$ matrix \mathbf{S}_1 and then simply neglect all modes z_m with $m > r$, that is, replacing the matrices \mathbf{S} and \mathbf{T} by \mathbf{S}_1 and \mathbf{T}_1 . The resulting transformation $\mathbf{z} = \mathbf{T}_1 \mathbf{x}$ with $\mathbf{A}' = \mathbf{T}_1 \mathbf{A} \mathbf{S}_1$, $\mathbf{B}' = \mathbf{T}_1 \mathbf{B}$, $\mathbf{C}' = \mathbf{C} \mathbf{S}_1$, $\mathbf{D}' = \mathbf{D}$, and $\mathbf{z}(0) = \mathbf{T}_1 \mathbf{x}_0$ yields a reduced r -dimensional model of the form (15.143). To ensure that output signals $\mathbf{y}'(\cdot)$ of the reduced system are good approximations of the original signals $\mathbf{y}(\cdot)$, we need to choose \mathbf{T} carefully: the higher modes, which are neglected, should contribute only little to the relevant behavior of the system.

There are various algorithms for linear model reduction. Proper orthogonal decomposition (also called *Karhunen–Loëve transform*) is based on simulated system trajectories. If the trajectories are seen as points in a high-dimensional space, the first principal components of the point cloud define a subspace that covers a maximal fraction of the data variance, and can be used as transformed variables. Another reduction method, called *balanced truncation* [8], employs a transformation to so-called balanced coordinates. As a guiding principle, transformed variables that are difficult to control should also be difficult to observe and vice versa. Such variables can be omitted (“truncated”) with little loss of accuracy in the input–output relation. The transformation in balanced truncation is chosen in a such a way that the transformed Gramians \mathbf{W}'_c and \mathbf{W}'_o become equal and diagonal

$$\mathbf{W}'_c = \mathbf{W}'_o = \text{Dg}(\sigma_1, \dots, \sigma_n) \quad (15.145)$$

with ordered diagonal entries σ_i , called *Hankel singular values* of the system. All higher modes $m > r$ are neglected, and the reduced system (15.143) will be asymptotically stable. An application to biochemical reaction systems has been described in Ref. [9]. Balanced truncation accounts for perturbations $\mathbf{u}(t)$ at all frequencies. Other methods, based on Krylov subspaces, can be tailored to yield optimal results for input signals within specific frequency ranges.

15.5.6 Optimal Control

The dynamics of linear systems can be shaped by feedback control. Consider a system

$$\frac{d\mathbf{x}}{dt} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \quad (15.146)$$

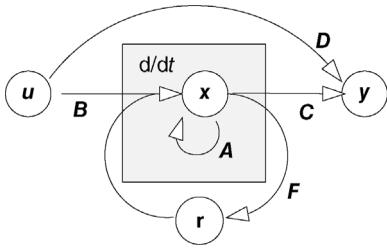


Figure 15.14 Closed-loop linear control system. The feedback via a vector r changes the system's dynamic properties.

with Jacobian matrix A (not necessarily stable) and input variables \mathbf{u} . To stabilize this system, we can add a feedback loop that measures the current state x , computes a linear function $\mathbf{r} = \mathbf{Fx}$, and adds it to the system input (see Figure 15.14). The resulting closed-loop system reads

$$\frac{dx}{dt} = Ax + B(u + r) = (A + BF)x + Bu. \quad (15.147)$$

With the right choice of F , the new Jacobian $(A + BF)$ can be made stable. In the linear-quadratic optimal control problem, we consider the system (15.119) with an objective function

$$I[\mathbf{u}(\cdot)] = \mathbf{x}^T(t_1)\mathbf{R}_0\mathbf{x}(t_1) + \int_{t_0}^{t_1} \mathbf{x}^T \mathbf{R}_1 \mathbf{x} dt + \int_{t_0}^{t_1} \mathbf{u}^T \mathbf{R}_2 \mathbf{u} dt. \quad (15.148)$$

The three terms, defined by matrices \mathbf{R}_0 , \mathbf{R}_1 , and \mathbf{R}_2 , score the final state, the time profiles of internal variables, and the time profiles of input variables. In the optimal control problem, we search for an input function $\mathbf{u}(\cdot)$ that minimizes $I[\mathbf{u}(\cdot)]$ where the initial state $\mathbf{x}(t_0)$ is predefined. With the objective function (15.148), the optimal input is given by

$$\mathbf{u}_{\text{opt}}(t) = -\mathbf{R}_2^{-1} \mathbf{B}^T \mathbf{Q}(t) \mathbf{x}(t). \quad (15.149)$$

The matrix \mathbf{Q} is the solution of the Riccati matrix equation

$$\begin{aligned} -\frac{d}{dt} \mathbf{Q} &= \mathbf{A}^T \mathbf{Q} + \mathbf{Q} \mathbf{A} + \mathbf{R}_1 - \mathbf{Q} \mathbf{B} \mathbf{R}_2^{-1} \mathbf{B}^T \mathbf{Q}, \\ \mathbf{Q}(t_1) &= \mathbf{R}_0. \end{aligned} \quad (15.150)$$

This equation must be solved backwards in time, starting from the final value \mathbf{R}_0 . The equations also apply if \mathbf{A} , \mathbf{B} , \mathbf{R}_1 , and \mathbf{R}_2 are time dependent. With an infinite time interval $t_1 = \infty$, the optimal input can be obtained from the stationary matrix $\overline{\mathbf{Q}}$, determined by the algebraic Riccati equation (i.e., setting the time derivative to 0)

$$0 = \mathbf{A}^T \overline{\mathbf{Q}} + \overline{\mathbf{Q}} \mathbf{A} + \mathbf{R}_1 - \overline{\mathbf{Q}} \mathbf{B} \mathbf{R}_2^{-1} \mathbf{B}^T \overline{\mathbf{Q}}. \quad (15.151)$$

If all state variables x_i are observable, optimal regulation can be implemented by the feedback matrix $\mathbf{F} = -\mathbf{R}_2^{-1} \mathbf{B}^T \overline{\mathbf{Q}}$.

15.6 Biological Thermodynamics

Molecules in cells diffuse, collide, change their conformations, and engage in chemical reactions. Even chemical equilibrium states appear very dynamic if we look at them on a molecular scale. For instance, even if the different protonation states of a substance are in equilibrium, this equilibrium emerges from a permanent dissociation and reassociation of ions and protons. How can we account for such microscopic dynamics in models? In *molecular dynamics*, microscopic processes are modeled in detail: individual molecules are treated as mechanical systems that follow Newton's laws of motion. However, molecular-dynamic simulations do not apply to entire cells: molecules in a cell are so numerous that all their microscopic motions can neither be measured nor computed. Moreover, since the dynamics is very sensitive to external perturbations, little thermal movements in the surroundings or small quantum mechanical uncertainties suffice to make molecule movements unpredictable.

Therefore, we need to find ways to disregard microscopic dynamics – but to account for it at the same time. *Phenomenological thermodynamics* describes systems by macroscopic variables like temperature or concentrations and deduces effective laws from conservation relations (e.g., for energy), phenomenological state equations, and the second law of thermodynamics. In *statistical mechanics*, these macroscopic laws are explained as *typical outcomes* of microscopic processes. Microscopic processes are not described precisely, but by random variables or probability distributions. Macroscopic variables like substance concentrations are obtained from time averages or averages over the ensemble of possible states. Thermodynamics plays a central role not only in physics and biochemistry, but has also informed information theory and statistics, for instance by contributing notions like entropy and computational methods such as simulated annealing (see Section 6.1).

15.6.1 Microstate and Statistical Ensemble

The state of a physical system, described to microscopic detail, is called the *microstate*. The microstate of a gas, for instance, can comprise the positions, orientation, and internal states of all molecules. Together with the corresponding momenta, these variables form a huge vector. The microstate of a system changes very fast, and its dynamics would have to be described by a very complicated, high-dimensional dynamics (e.g., a “billiard game” of 10^{23} gas molecules) or by a microscopic random dynamics (where the outcome of molecule collisions, for

instance, is thought to be inherently stochastic). In both cases, a precise description is practically impossible and the system is rather described by probability distributions.

Probability distributions can be used in different ways: in the *Boltzmann picture*, the distribution concerns individual components of a system, for example, the molecules in a gas, describing them in a statistical manner. We can see the many particles as realizations of one “ideal” particle, which, through interactions with the surrounding particles, jumps between possible states, realizing all these states with different probabilities. In the *Gibbs picture*, this idea of a random distribution of possible states is applied to the state of an entire (multiparticle) system. In both pictures, randomness in system states can arise from internal randomness (particle collisions with unpredictable outcome) or energy exchange with the environment, and the statistical distributions can be depicted as an ensemble, that is, as a large set of identical systems representing possible realizations of the random process. In the Boltzmann picture, it is the gas molecules that form the ensemble. In the Gibbs picture, the ensemble is formed by hypothetical versions of the entire gas in its different possible microstates.

15.6.1.1 Thermodynamic Equilibrium and Detailed Balance

Macroscopically, processes like diffusion, chemical reactions, or heat transport are described by state variables, such as temperature, pressure, and substance concentrations that can be measured or controlled in experiments. These variables constitute the *macrostate* of a system. To derive physical laws for macrostates (for instance, a transport equation for heat), we need to link them back to the underlying microstates. Since microstates are unknown and rapidly changing, one associates a macrostate with a whole set or statistical ensemble of microstates. A simple example is the microcanonical ensemble associated with a macrostate with precisely defined total energy E . In this microcanonical ensemble, all microstates with energy E have equal probability and all other microstates have zero probabilities.

Ensembles of microstates can have different interpretations: on the one hand, they may describe the microstates that a system will typically visit on its path through state space; on the other, it may reflect our knowledge of the system state in a given moment. If a system, according to its microscopic dynamics, can reach all of the microstates compatible with a given macrostate, then both views can be made to agree. In such cases, we may describe the system by an *ergodic random process*, a process in which the state distribution in one moment (for different realizations of the

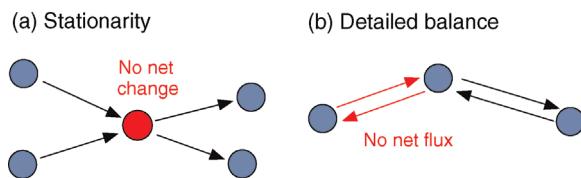


Figure 15.15 Balance conditions for thermodynamic equilibrium states. (a) Stationarity (condition for macroscopic steady state): to keep the probability of a microstate constant (red circle), incoming and outgoing probability flows must be balanced. (b) Detailed balance (condition for thermodynamic equilibrium). Between any two states, the probability flows in both directions must cancel out, so there is not net probability flow.

process) and the distribution of states over time (for each single realization) are identical.

An important kind of macrostates are *thermodynamic equilibrium states*, states in which all macroscopic variables remain constant and the macroscopic fluxes vanish. If a microscopic dynamics is described as a random jump process between microstates, the dynamics of equilibrium states has to satisfy two restrictions. To ensure stationarity, the distribution of microstates must be constant in time, so the probability flow into each state and the probability flow leaving this state must be balanced (see Figure 15.15a). To ensure an actual equilibrium, each individual flow (from state i to state j) must additionally be balanced with the reverse transition (from state j back to state i), such that both flows sum to zero (see Figure 15.15b). Note that this requirement, called *detailed balance*, is a condition for equilibrium states and distinguishes them from (nonequilibrium) steady states.

15.6.2 Boltzmann Distribution and Free Energy

15.6.2.1 Boltzmann Distribution

Due to detailed balance, all microscopic processes in equilibrium states fluctuate back and forth with equal rates in both directions. Individual molecules in a gas, for instance, will exchange energy by hitting each other, but the overall distribution of molecule energies remains practically constant. In the Boltzmann picture, we can focus on a single molecule as a representative and consider the probability distribution of its energy states. This distribution will be characterized by a fixed average energy, the total energy of the gas divided by the molecule number. The resulting *canonical ensemble* of molecule states is described by the *Boltzmann distribution*: the probability p to find a particle in a specific state x with energy $E(x)$ is proportional to the Boltzmann weight $w(x) = \exp(-E(x)/k_B T)$, where T is the absolute temperature T (in Kelvin) and $k_B \approx 1.38 \times 10^{-23} \text{ J K}^{-1}$ is

Boltzmann's constant, that is, the gas constant R divided by Avogadro's constant N_A . To obtain the probability

$$p(x) = \frac{1}{Z} e^{-E(x)/(k_B T)}, \quad (15.152)$$

the Boltzmann weights are normalized by the *partition function* $Z = \sum_x w(x')$, the sum over the weights of all states x' . The relative probabilities of molecule states follow directly from the ratios of their Boltzmann weights. A state with a higher energy has a lower probability: in the canonical ensemble, any microstate can be reached, but states of high energy are very unlikely. The temperature T in Eq. (15.152) determines the average energy of the system: at higher temperatures, the distribution is shifted toward higher energy states. The Boltzmann distribution also applies in the Gibbs picture, for the possible states of a (possibly complicated) system in thermal contact with an environment of temperature T .

15.6.2.2 Free Energy

The state x considered in the Boltzmann distribution (15.152) may comprise a large number of variables (e.g., atom positions) that characterize a microstate in full detail. Often, we are only interested in a few of those variables or in larger sets of such microstates. Figure 15.16 shows an example: a molecule (shown as a bar) can switch between three conformations: straight (1), bent (2), and pointed (3). In states 1 and 2, but not in state 3, a second molecule (shown by an ellipse) can bind. There are five states in total, but we are only interested in whether or not the second molecule is bound, irrespective of the molecule conformations. Thus, we can describe the system by two macrostates, U (unbound) and V (bound), each of which comprises several conformation substates.

In modeling, for example, when deriving gene regulation functions based on transcription factor binding, we would like to use the Boltzmann distribution, but only for the binding states, while other microscopic degrees of

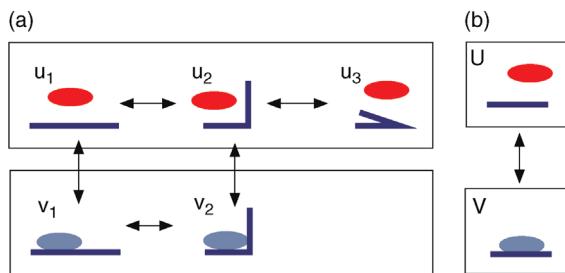


Figure 15.16 Microstates and macrostates. (a) Molecules with different binding and conformation states, drawn as a bendable bar and an ellipse. Arrows indicate possible state transitions. Macrostates ("unbound" and "bound") are shown by boxes. (b) The same model, approximated by an effective two-state model in which molecule conformations are disregarded.

freedom should be disregarded. Thus, our goal is to treat the binding states *as if* they were microstates without further internal structure. To assign occupation probabilities to the two states, we assume Boltzmann distribution $\exp(-F_X/(k_B T))$ of exactly the same form as Eq. (15.152), but replacing the energies by effective variables called *free energies* F . How can we choose these variables? In fact, for consistency, the Boltzmann weight $w(X)$ of a macrostate X must represent the Boltzmann weights $w(x)$ summed over all its substates x , or in other words: the partition function $Z_X = \sum_{x \in X} w(x)$ of state X . Equating this to $\exp(-F_X/(k_B T))$ and solving for the free energy F_X , we obtain the identity $F_X = -k_B T \ln Z_X$: the free energy of a macrostate effectively summarizes the energies and the Boltzmann probabilities of its substates. If a macrostate X contains N_X microstates x with identical energies E_X , we obtain

$$\begin{aligned} F_X &= -k_B T \ln \sum_{x \in X} \\ w(x) &= -k_B T \ln(N_X \exp(-E_X/(k_B T))) \\ &= E_X - T k_B \ln N_X. \end{aligned} \quad (15.153)$$

The second term, although showing the unit of an energy, represents a simple counting of substates. Equation (15.153) is an example of the general relationship $F = E - TS$, where $S = k_B \ln N_X$ is the statistical entropy. In general cases, with microstates having different energies, the entropy has a more complicated form.

Replacing several microstates by one macrostate as in our example is a form of model reduction. Assuming fast transitions between conformations, but slow transitions between binding states, we can assume, for each binding state, a quasi-equilibrium between all conformations. From the two Boltzmann distributions and the (possibly conformation-dependent) known rate constants for binding and unbinding events, we can compute the slow binding and unbinding rates, assuming that these transitions affect the conformation equilibria only weakly. If the entire system is in equilibrium, the approximation yields the exact result: the Boltzmann weights of the conformation states are precisely represented by the free energies. In nonequilibrium states, the approximation is justified if the dynamics toward a conformation equilibrium is much faster than the typical rate of binding and unbinding events.

15.6.3 Entropy

15.6.3.1 The Second Law of Thermodynamics

Many physical processes occur spontaneously in one direction, but never in the opposite direction. For instance, the mixing of water colors occurs spontaneously

while the reverse process, that is, a spontaneous separation, is never observed. Even if colors could be unmixed by a machine (e.g., a centrifuge separating color particles by their size), this machine would rely on some energy supply from other irreversible process (e.g., burning of coal to generate electricity); again, the overall process would not spontaneously proceed in reverse direction (involving carbon dioxide being spontaneously turned into coal and oxygen again). Thus, there is a general directionality of time, and since this directionality applies to *all* macroscopic processes, it has been given the status of a physical law, the second law of thermodynamics. The first law of thermodynamics, in contrast, states that energy is a conserved quantity and that it can be exchanged between systems in the form of work (associated with macroscopic processes like mechanic compression) and heat (i.e., an “uncontrolled” exchange of energy through microscopic degrees of freedom).

In Sommerfeld’s formulation of the second law, every thermodynamic system is characterized by an extensive state variable called entropy. In reversible processes (which only exist as an idealized picture), the entropy S of a system is increased by the amount ΔQ of absorbed heat divided by the absolute temperature: $\Delta S = \Delta Q/T$. In irreversible processes (i.e., all real thermodynamic processes), there is an additional positive amount of entropy, which arises spontaneously from processes within the system. Due to such internal processes, the entropy of an isolated system will increase until a thermodynamic equilibrium, that is, a state of maximal entropy, has been reached. By converse, the only way to decrease a system’s entropy is by coupling it to processes outside the system that produce even more entropy. To summarize, any spontaneous thermodynamic process involves entropy production.

15.6.3.2 Statistical Entropy

In the axiomatic definition given above, entropy was introduced as a phenomenological state variable, whose precise meaning remained unclear. In Boltzmann’s approach of statistical thermodynamics, however, entropy is given a specific interpretation: the entropy of a macrostate is a function that characterizes the probability distribution $p(x)$ of the microstates x behind it. For a system with W discrete microstates and a uniform distribution $p(x) = 1/W$, entropy reads

$$S = k \log W \quad (15.154)$$

with a constant prefactor k . This is the entropy term we encountered in Eq. (15.153). If two physical systems (with W_1 and W_2 states, respectively) are independent, the combined system has $W_1 \cdot W_2$ states, and due to the logarithm in Eq. (15.154), their entropies will be additive.

More generally, the statistical entropy for systems with discrete states x_i and state probabilities $p(x_i)$ is given by [10–12]

$$S = -k \sum_{i=1}^n p(x_i) \log p(x_i). \quad (15.155)$$

Entropy can be seen as a measure of statistical variety: it is maximal ($S = \log n$) for uniform distributions and minimal ($S = 0$) if all probability is concentrated in one state. Distributions of real-valued random variables, with probability density $p(x)$, are characterized by the differential entropy

$$S = -k \int p(x) \log p(x) dx, \quad (15.156)$$

where x can be one- or multidimensional. The differential entropy can be positive or negative, and it vanishes for the uniform distribution in an interval of width 1.

The concept of statistical entropy, initially introduced for distributions of thermodynamic states, applies to all kinds of probability distributions. If a distribution describes what we *know* about a random variable, its entropy quantifies the uncertainty in this variable or, in other words, the information that we would gain by measuring the variable precisely. In information theory [10], the concept of statistical entropy is used to define the information content of signals transmitted through noisy channels. The prefactor k in Eqs. (15.155) and (15.156), and accordingly the basis of the logarithm, is a matter of unit conventions. In thermodynamics, the natural logarithm and the Boltzmann constant $k_B = R/N_A \approx 1.3 \times 10^{-23} \text{ J K}^{-1}$ are used. In information theory, Shannon entropy (measured in units of bits or Shannons) is defined with $k = 1$ and using the binary logarithm \log_2 .

Example 15.31 Shannon Entropy of a Picture

Consider pictures consisting of 100×100 pixels, where each pixel can be black or white. In total, there are $W = 2^{10000}$ possible pictures (most of which will look completely random; some of them show you, reading this book); to store one such picture in a file, a storage of 10 000 bits (one for each pixel) will be needed. Accordingly, the set of all possible pictures, with equal probabilities for each picture, will have a Shannon entropy of $\log_2 W = 10000$ Shannon. If some of the pictures in the ensemble are more likely than others, the entropy (in Shannon) will be lower, while the size in bits will remain the same. Using file compression, the size in bits may be decreased, but with lossless file compression, it can never fall below the value set by the Shannon entropy.

15.6.3.3 Principle of Maximal Entropy

In statistical thermodynamics, a macrostate is typically associated with a statistical ensemble of microstates. On what grounds should this ensemble be chosen? As an example, consider a single gas molecule within a gas of fixed total energy. The energy of our molecule will change due to collisions, but its average energy, due to symmetry reasons, will be given by the total energy of the gas divided by the number of gas molecules. How will the energy values be distributed?

A helpful and common assumption is that the system considered is in thermodynamic equilibrium. According to the second law of thermodynamics, the distribution should then show a maximal entropy Eq. (15.156), given the restrictions imposed by the macrostate. The maximal entropy criterion can also be justified in terms of information [13]: when choosing a statistical ensemble of unknown microstates, we should impose as little prior information as possible. This principle can be used in the context of statistics or machine learning, for instance to choose probability distributions for unknown model parameters (see Section 10.1).

Using the principle of maximal entropy, we can obtain the microcanonical and the canonical ensemble. (i) *Microcanonical ensemble*. In a system with fixed energy E_0 (a macroscopic variable), microstates with energies $E \neq E_0$ must have zero probability, and all microstates with energy E_0 must have equal probabilities because this distribution maximizes the entropy. This is the probability distribution of the microcanonical ensemble. This is intuitive: if nothing is known about a system aside from its total energy, there is no reason why any of the allowed states should be more likely than the others. (ii) *Canonical ensemble*. Instead of a fixed total energy, we can consider a *fixed average energy*: under this constraint, the entropy is maximized by the Boltzmann distribution, and we obtain a canonical ensemble.

15.6.4 Equilibrium Constant and Energies

Thermodynamics predicts some important relations between molecular energies, state probabilities or concentrations, and fluxes in biochemical systems. Central among such relations is the *mass-action ratio* of reactions, the product of product concentrations, divided by the product of substrate concentrations. In equilibrium states this ratio has a constant value called equilibrium constant of the reaction. For reactions with different numbers of substrates and products, it can be convenient to use unitless equilibrium constants, achieved by multiplying or dividing the equilibrium

mass-action ratio by the standard concentration (typically assumed to be 1 mM).

The equilibrium constant of a reaction is directly determined by the energy difference between products and substrates. To see this, consider a two-state system that can randomly switch between states A and B with rates $w_{A \rightarrow B}$ and $w_{B \rightarrow A}$. For the occupation probabilities p_A and $p_B = 1 - p_A$ in thermodynamic equilibrium, detailed balance implies that

$$p_A w_{A \rightarrow B} = p_B w_{B \rightarrow A}. \quad (15.157)$$

Since detailed balance must hold between any two states, circular probability flows as in $A \rightarrow B \rightarrow C \rightarrow A$ are excluded. Taken together, detailed balance and the Boltzmann distribution (15.152) yield a relationship between the energies of two states and the transition rates between them. Consider two states with free energies F_A and F_B representing, for instance, a molecule with two conformations. The equilibrium distribution at temperature T is a Boltzmann distribution given by $p(A) = e^{-\beta F_A}/Z$ and $p(B) = e^{-\beta F_B}/Z$ with $\beta = 1/(k_B T)$ and the partition function $Z = e^{-\beta F_A} + e^{-\beta F_B}$. As a condition for detailed balance, the transition rates must satisfy

$$\frac{w_{A \rightarrow B}}{w_{B \rightarrow A}} = \frac{p_B}{p_A} = e^{-\beta(F_B - F_A)}, \quad (15.158)$$

so the ratio of forward and backward rates depends only on the energy difference. This identity has important consequences for kinetic models, as we can easily see. An ensemble of two-state systems can be described by the average concentrations s_A and s_B of systems in the two states. These concentrations follow a kinetic model with a reaction $A \rightleftharpoons B$ and a mass-action rate

$$v = k_{A \rightarrow B} s_A - k_{B \rightarrow A} s_B. \quad (15.159)$$

Like in unimolecular reactions, the macroscopic rate constants $k_{i \rightarrow j}$ are given by the microscopic transition rates $w_{i \rightarrow j}$, so their ratio is determined by the energy difference

$$\frac{k_{A \rightarrow B}}{k_{B \rightarrow A}} = e^{-\Delta F/(kT)}. \quad (15.160)$$

At the same time, the rate in Eq. (15.159) must vanish in chemical equilibrium, so the ratio $K_{eq} = s_B^{eq}/s_A^{eq}$, called equilibrium constant, is given by

$$\frac{s_B^{eq}}{s_A^{eq}} = \frac{k_{A \rightarrow B}}{k_{B \rightarrow A}}. \quad (15.161)$$

The fact that the equilibrium constant is fixed and identical for all equilibrium states justifies its name. The relation between concentrations and energies does not only hold for two-state systems, but for chemical systems in general. Any reaction event in a chemical reaction system, together with its reverse event, moves the system

between states characterized by different molecule numbers and, therefore, different amounts of free energy. Our previous considerations about two-state systems also hold for these states, and therefore for any chemical reaction events. Accordingly, the mass-action ratios of all reactions in thermodynamic equilibrium are determined by the free energy differences between reaction substrates and products.

An important consequence of this is that the equilibrium constant in enzyme-catalyzed reactions depends only on the reaction's sum formula and it is independent of enzymes catalyzing the reactions. Enzymes can accelerate reactions in both directions – which allows them to influence the metabolite concentrations in (nonequilibrium) steady states. The only way in which enzymes can affect equilibrium states is by binding substrate molecules and thereby reducing the probability of unbound particle states.

15.6.5 Chemical Reaction Systems

Substance concentrations and other macroscopic variables of chemical systems follow macroscopic laws based on thermodynamics, which summarize and hide the microscopic dynamics. In principle, the transition from microscopic to macroscopic chemical models works exactly as we saw it in Section 7.2 on stochastic biochemical models. However, the usage of thermodynamic notions like the Gibbs energy make the link between macroscopic variables (e.g., concentrations, fluxes, or temperature) and microscopic properties (e.g., molecule energies) more transparent.

15.6.5.1 Temperature and Pressure as Free Variables

The principle of maximal entropy holds for isolated systems, that is, systems that exchange no work, heat, or matter with their environment. Such systems would consist of a fixed amount of matter in a fixed volume and with a fixed amount of energy. In real biological systems, in contrast, the controllable macroscopic variables are not entropy and volume, but temperature and pressure, while entropy, energy, and volume emerge dynamically. To describe such systems, we first consider a system with total energy E in a volume V . In equilibrium, the system will have an entropy $S(E, V)$, which follows from the equilibrium distribution of microstates. By inverting this function, we can obtain the energy as a function $E(S, V)$, and pressure and temperature can be defined by derivatives

$$T(S, V) = \frac{\partial E(S, V)}{\partial S}, \quad p(S, V) = -\frac{\partial E(S, V)}{\partial V}. \quad (15.162)$$

Therefore, the differential of the energy reads $dE = TdS - pdV$. State variables like the energy E , which allow us to compute thermodynamic variables by derivatives, are called thermodynamic potentials.

In models of biological systems, temperature and pressure should be treated as free variables, whereas entropy and volume should be derived from relations similar to (15.162). To obtain equations of this form, we need to find a new thermodynamic potential that assumes the role of the energy E after our change of variables. Using a *Legendre transformation*, we introduce T as a free variable, express the entropy as a function $S = S(T, V)$, and introduce the *free energy* $F(T, V) = E(S(T, V), V) - TS(T, V)$ as our new thermodynamic potential. Its differential reads $dF = dE - TdS - SdT = -SdT - pdV$. A second Legendre transformation brings us from volume V to pressure p and yields the *Gibbs energy* $G(p, T) = E + pV - TS$, with the differential $dG = -SdT + Vdp$. The Gibbs free energy is the right potential for describing systems at given temperature and pressure: as we can see from the differential, entropy and volume follow from the relationships

$$S(p, T) = -\frac{\partial G(p, T)}{\partial T}, \quad V(p, T) = \frac{\partial G(p, T)}{\partial p}. \quad (15.163)$$

In isolated systems (i.e., systems with fixed amounts of matter and energy), internal processes like heat transport, substance diffusion, and chemical reactions are accompanied by entropy production. When entropy has reached a maximum, the system is in equilibrium: the substance concentrations remain constant and the net reaction rates vanish. For systems at given temperature and pressure, analogous conditions for spontaneous processes and equilibrium states follow from a principle of minimal Gibbs energy.

15.6.5.2 Gibbs Energy and Chemical Potentials

Macroscopically, a well-mixed solution of metabolites is described by pressure p , temperature T , and the mole numbers m_i of substances i . The system's Gibbs energy is a function $G(p, T, m_1, m_2, \dots)$ of these variables, and its derivatives $\mu_i = \partial G / \partial m_i$ (in kJ mol^{-1}) are called the *chemical potentials* of the substances. As an extensive variable, the Gibbs energy can be written as a sum

$$G(p, T, m) = \sum_i \mu_i m_i \quad (15.164)$$

of chemical potentials for all molecule species. In ideal mixtures, the chemical potential of a substance i with concentration s_i is given by

$$\mu_i(p, T, s_i) = \mu_i^{(0)}(p, T) + RT \ln s_i / s^\circ, \quad (15.165)$$

where $\mu_i^{(0)}$ denotes the chemical potential of the substance at standard concentration s° (usually 1 mM) and $R \approx 8.314 \text{ J mol}^{-1} \text{ K}^{-1}$ is Boltzmann's gas constant. Different conventions about s° lead to different nominal values of $\mu_i^{(0)}(p, T)$; s° is often omitted in the formula, assuming that s_i is formally dimensionless, but refers to mM. Molecule interactions in real mixtures can be captured by an additional term $RT \ln f_i^0$, with the activity coefficient f_i^0 . If molecules are charged (e.g., ions transported across cell membranes), their electric energy can be included in the energy function; accordingly, their chemical potentials are replaced by *electrochemical potentials* $\eta_i = \mu_i + z_i F \Phi$, where z_i is the number of elementary charges per molecule (negative for negatively charged molecules), F is Faraday's constant (i.e., 1 mole of elementary charges), and Φ is the electric potential at the place of the molecule.

15.6.5.3 Reaction Gibbs Energy

Chemical reaction events will change the substance concentrations and thereby a system's Gibbs energy. A reaction j with stoichiometric coefficients n_{ij} causes a Gibbs energy difference (in kJ per mole of reaction events)

$$\Delta_r G_j = \sum \mu_i(p, T) n_{ij}. \quad (15.166)$$

Assuming a standard state (typically $p = 1.015 \text{ bar}$, $T = 298.15 \text{ K}$, and concentrations $s_i = s^\circ = 1 \text{ mM}$ for all reactants), we obtain the standard reaction Gibbs energies $\Delta_r G_j^{(0)}$, which can be written as

$$\Delta_r G_j^{(0)} = \sum_i n_{ij} \mu_i^{(0)} = \sum_i n_{ij} G_i^{(0)}, \quad (15.167)$$

in terms of Gibbs energies of formation $G_i^{(0)}$. In systems with controlled temperature and pressure, chemical reactions are driven by a Gibbs energy difference between substrates and products. In equilibrium, the system's total Gibbs energy reaches a local minimum and the $\Delta_r G_j$ values of all reactions vanish. We can use this to confirm the link between equilibrium concentrations and energies. By setting Eq. (15.166) to zero and inserting (15.165) and (15.167), we obtain an expression for the equilibrium constant

$$K_{\text{eq},j} = \prod_i (s_i^{\text{eq}})^{n_{ij}} = \exp(-\Delta_r G_j^{(0)} / RT), \quad (15.168)$$

where the vector \mathbf{s}^{eq} contains the metabolite concentrations in a chemical equilibrium state. Equation (15.168) links the equilibrium constants to standard Gibbs free energies of formation. It is analogous to Eq. (15.160) for microscopic states at given temperature and volume.

15.6.5.4 Wegscheider Conditions and Haldane Relationships

Equilibrium thermodynamics yields constraints not only for the states, but also for the parameters of biochemical

network models: the equilibrium constants must respect Wegscheider conditions, and the rate constants and equilibrium constant in each reaction must satisfy a Haldane relationship.

- 1) **Wegscheider conditions** The equilibrium constant is defined as the ratio between product and substrate concentrations in equilibrium. Accordingly, the equilibrium constants, multiplied along a closed, isolated loop of chemical reactions, must yield a value of 1, and the logarithmic equilibrium constants must sum to 0. This is an example of a *Wegscheider condition*, showing that equilibrium constants cannot be generally treated as independent model parameters. Wegscheider conditions, in general reaction networks, are related to flux loops, hypothetical stationary flux distributions with a zero production and consumption of (both internal and external) metabolites. We can see this as follows. By definition, equilibrium constants satisfy the relationship

$$\ln K_{\text{eq},j} = \ln \left[\prod_i s_i^{\text{eq}}^{n_{ij}} \right] = \sum_i n_{ij} \ln s_i^{\text{eq}}. \quad (15.169)$$

In matrix form, Eq. (15.169) becomes $\ln \mathbf{K}_{\text{eq}} = \mathbf{N}^{\text{tot}^T} \mathbf{x}$, where $\mathbf{x} = \ln \mathbf{s}^{\text{eq}}$ and \mathbf{N}^{tot} is the stoichiometric matrix containing all (internal and external) metabolites. The fact that all equilibrium constants are derived from the same vector \mathbf{x} can make them linearly dependent: the vector of equilibrium constants must satisfy the condition

$$\mathbf{Q}^T \ln \mathbf{K}_{\text{eq}} = 0, \quad (15.170)$$

where \mathbf{Q} , a right nullspace matrix of \mathbf{N}^{tot} , contains loop fluxes (i.e., stationary flux distribution without a net production of *any* metabolite). Each such loop flux induces a Wegscheider condition [14,15]. Notably, Wegscheider conditions do not hold only for equilibrium constants, but for any quantities that represent differences along chemical reactions, for example, the reaction Gibbs energies ΔG_l . An analog of a Wegscheider condition in electric circuits is Kirchhoff's loop rule, which states that the sum of directed voltages along closed loops in a circuit must vanish.

- 2) **Haldane relationships** A second type of constraints, the Haldane relationships [16], relate the rate constants of a reversible rate law to the equilibrium constant of the reaction. For the mass-action rate law $v = k^f s - k^r p$, the Haldane relationship reads simply $K_{\text{eq}} = k^f / k^r$. To derive such relationships, we consider a chemical equilibrium state, set the reaction rate to zero, and solve for the mass-action ratio.

15.6.5.5 Data for Thermodynamic Calculations

The equilibrium constants are key quantities in biochemical models. In principle, according to Eq. (15.168) they can be obtained from the Gibbs energies of formation $G_i^{(0)}$. Practical calculations, however, have to overcome several problems. First, substances can exist in different protonation states, and models usually lump these states in a single molecule species. The lumped species can be characterized by transformed standard Gibbs free energies $G_i^{(0)}$, effective energies that subsume the standard Gibbs energies of different protonation states and their relative abundances. Since the abundances vary with state variables like pH or ionic strength, the transformed values are not constant, but depend on these state variables [17,18]. The pH-dependence can be important for transport reactions, where substances experience different biochemical conditions on the two sides of a membrane [19].

Second, thermodynamic quantities have only been measured for some substances [20]. To construct large-scale models, many values need to be inferred by numerical estimation, for instance by the group contribution method [21] or the component contribution method [22]. Equilibrium constants and Gibbs energies for various compounds can be obtained from the tool eQuilibraTor (equilibrator.weizmann.ac.il) [23].

15.6.6

Nonequilibrium Reactions

Life is based on nonequilibrium states and therefore depends on a dissipation of Gibbs free energy. Among all nonequilibrium states, steady states play a special role. As suggested by their German name *Fließgleichgewicht* (flow equilibrium) [24,25], they represent an equilibrium between fluxes in which all variables remain constant in time, but there may still be a net conversion of matter or energy – which would not be the case in equilibrium states. In biochemical network models, processes are often assumed to be in steady state; a cell in chemical equilibrium, in contrast, would be dead.

Generally, reactions in nonequilibrium must produce entropy and dissipate Gibbs energy. A chemical reaction produces a certain amount of entropy per time and volume; this value, given by $\sigma = Av/T$, must be positive. Therefore, any reaction flux v requires a thermodynamic driving force $A = -\Delta_r G$ with identical sign:

$$v_j \neq 0 \Rightarrow \text{sign}(v_j) = \text{sign}(A_j). \quad (15.171)$$

As a consequence, fluxes must always lead from higher to lower chemical potentials. If a potential difference vanishes, also the net flux will vanish, and we obtain a chemical equilibrium. Equation (15.171) is used in some types

of flux analysis as a criterion for feasible flux distributions. In such models, chemical potentials μ_i must be chosen such that the resulting thermodynamic forces $A_j = -\Delta_r \mu_j$ satisfy all sign constraints. This criterion excludes certain flux cycles for which no such chemical potentials can exist. Furthermore, Eq. (15.165) relates the chemical potentials to metabolite concentrations. Therefore, given flux directions will limit the possible ranges of metabolite concentrations, and given concentration ranges will limit the possible flux directions (see Chapter 3).

There is a second explanation for the sign constraint (15.171). Each reaction flux is a difference $v = v^+ - v^-$ of microscopic forward and backward fluxes, which can be resolved, in principle, using isotope-labeled metabolites. The flux ratio is given by

$$\frac{v^+}{v^-} = e^{A/RT} \quad (15.172)$$

with the thermodynamic driving force $A = -\Delta_r G$ [26], which therefore defines the net flux direction. While enzymes can increase the forward and backward fluxes, the ratio Eq. (15.172) is predetermined by thermodynamics (and dependent on the reactant concentrations). The thermodynamic force tells us whether a reaction is close to equilibrium ($A \approx 0, v^f \approx v^r$) or to being irreversible (large $A, v^f \gg v^r$).

15.6.6.1 Variational Principle for Flux States

Like many other physical laws, the flux–force relation (15.172) can be derived from a variational principle [27]. Consider an FBA problem in which metabolites are balanced by external exchange fluxes \mathbf{v}_e (e.g., dilution fluxes in a growing cell, or transport reactions across the cell membrane). The exchange fluxes must be balanced by a fixed net metabolite production $\mathbf{b} = -N_e \mathbf{v}_e$ within the network. To meet this demand, the internal forward and backward fluxes (in vectors \mathbf{v}_f and \mathbf{v}_r) must satisfy the stationarity condition

$$\mathbf{Nv}_f - \mathbf{Nv}_r = \mathbf{b}. \quad (15.173)$$

Fleming *et al.* showed that solutions to this problem, that is, stationary, thermodynamically feasible flux distributions, can be obtained by solving the optimality problem [27]

$$\begin{aligned} (\mathbf{v}_f^*, \mathbf{v}_r^*) = \text{argmin}_{(\mathbf{v}_f, \mathbf{v}_r) > 0} & [\mathbf{v}_f^T (\ln \mathbf{v}_f + \mathbf{c} - \mathbf{1}) \\ & + \mathbf{v}_r^T (\ln \mathbf{v}_r + \mathbf{c} - \mathbf{1})], \end{aligned} \quad (15.174)$$

with an arbitrary vector \mathbf{c} , $\mathbf{1}$ being a vector of ones, and satisfying the side constraint (15.173) with a given feasible vector \mathbf{b} . Any solution will yield feasible fluxes satisfying relation (15.172), where the Lagrange multipliers \mathbf{y} associated with the constraint (15.173) define the chemical potentials by $\boldsymbol{\mu} = -2\mathbf{y}/RT$. Moreover, any thermodynamically feasible solution to an FBA problem is a

solution to the optimality problem Eq. (15.174) with appropriate vectors \mathbf{b} and \mathbf{c} . Fleming's variational principle accounts for mass and energy conservation and for the second law of thermodynamics. Moreover, it establishes a practical way to compute thermodynamically feasible flux distributions, namely by convex optimization.

15.6.6.2 Consequences of the Flux–Force Relation

The general relation (15.172) between thermodynamic forces and flux ratios has important consequences for metabolic models: (i) Forces determine flux directions, as we already knew from Eq. (15.171). (ii) Only a fraction $v/v^+ = 1 - e^{\Delta G/RT}$ of the forward flux contributes to the net flux, while the rest must compensate the backward flux. The percentage of enzymatic capacity wasted due to backward fluxes is directly determined by the driving force. The closer a reaction comes to equilibrium, the smaller the relative net flux will be, and the more abundant (or catalytically active) the enzymes must be to reach a given net flux. In equilibrium states, this enzyme effort becomes infinite. (iii) Reversible rate laws share the general form

$$v = E \frac{q^+ \prod_i s_i^{n_i^s} - q^- \prod_i s_i^{n_i^p}}{D(\mathbf{s})}, \quad (15.175)$$

where n_i^s and n_i^p represent the stoichiometric coefficients of substrates and products; the denominator $D(\mathbf{s})$ can vary between rate laws. The form (15.175) comprises many reversible rate laws as special cases, including reversible mass-action and Michaelis–Menten kinetics and generalized rate laws such as thermodynamic–kinetic modeling rate laws (TKM) [28], modular rate laws [29], and separable rate laws [30]. The two terms in the numerator correspond to forward and backward fluxes and their ratio yields

$$\frac{v^+}{v^-} = \frac{q^+ \prod_i s_i^{n_i^s}}{q^- \prod_i s_i^{n_i^p}} = e^{\ln(q^+/q^-) - \sum_i n_i \ln s_i} = e^{A/RT}. \quad (15.176)$$

where $K_{\text{eq}} = q^+/q^-$ is the equilibrium constant. This formula agrees with Eq. (15.172), which justifies rate laws of the form (15.175).

15.6.6.3 Reaction Energetics and Flux Control

By imposing constraints on kinetics, thermodynamics has an impact on the control properties of metabolic systems. In reactions close to equilibrium, a change in the enzyme level will have little effect on the pathway flux; such reactions exert less flux control than strongly forward-driven reactions in the same pathway. For unbranched pathways with simple rate laws, this can be shown directly. If all enzymes operate in their linear range (with

rates approximated by mass-action rate laws), the scaled flux control coefficients are proportional to [31]

$$C_{v_j}^j \sim \frac{e^{A/RT} - 1}{\prod_{m=1}^j e^{A/RT}}. \quad (15.177)$$

On the contrary, if all enzymes are substrate-saturated, the control coefficients are proportional to [32]

$$C_{v_j}^j \sim e^{A/RT} - 1. \quad (15.178)$$

In both cases the absolute values can be determined from the summation theorem $\sum_j C_{v_j}^j = 1$. Given the thermodynamic forces (or, equivalently, the imbalance ratios between mass-action ratios and equilibrium constant), the control coefficients will be fully determined without any additional dependence on metabolite levels. Since strongly forward-driven reactions exert relatively high flux control, they are also plausible targets for allosteric or transcriptional regulation.

15.6.7

The Role of Thermodynamics in Systems Biology

What are the main implications of thermodynamics for systems biology? Since biochemical systems rely on the dynamics of molecules, they are inherently governed by thermodynamics. A main insight from thermodynamics is that life relies on nonequilibrium processes. The complex structures of organisms would rapidly be destroyed by thermal processes if they were not constantly renewed. This renewal requires an input of Gibbs energy, which can be supplied in the form of chemical energy (or, in photosynthetic organisms, sunlight). In the ATP/ADP system for energy transmission, the Gibbs energy taken up is first used to establish a nonequilibrium ratio between ADP and ATP. As long as the mass-action ratio is kept below the equilibrium constant, the conversion of ATP to ADP can be used a source of Gibbs energy that can drive a variety of irreversible processes in the cell. Aside from their general importance, thermodynamic laws have also some major practical consequences for biochemical models:

- 1) *Chemical potentials* Each compound i has a chemical potential μ_i , defined as the derivative of the system's Gibbs energy by the compounds' mole number. For ideal mixtures, the chemical potentials are given by $\mu = \mu^{(0)} + RT \ln s$ with gas constant R , absolute temperature T , and concentration s given in standard units.
- 2) *Thermodynamic forces* The negative difference $A = -\Delta\mu$ of chemical potentials along reactions, called

reaction affinity, can be seen as the thermodynamic force driving the reaction. A flux v , at a driving force A , produces entropy at a rate $v \cdot A/T$. In order to produce positive entropy, all fluxes must flow in the direction of the driving force.

- 3) *Partial fluxes* Chemical reactions are reversible and can be split into forward and backward fluxes satisfying the flux–force relation Eq. (15.172). The reaction affinity determines the relative backward flux, and this relationship predetermines the numerator of reversible rate laws.
- 4) *Equilibrium states* In chemical equilibrium, forward and backward fluxes are balanced and both driving forces and net fluxes vanish. The mass-action ratio in such states, called equilibrium constant, is given by $K_{\text{eq}} = \exp(-\Delta\mu^{(0)}/RT)$.
- 5) *Constraints on model parameters* Equilibrium constants and enzymatic rate constants must satisfy Wegscheider conditions and Haldane relationships.

Models that violate thermodynamic conditions run the risk of describing a chemical *perpetuum mobile*. This may be acceptable in some cases, for instance, if reactions involve cofactors that are deliberately neglected in the model. However, usage of thermodynamic laws can make models more realistic, provide additional knowledge in parameter fitting (e.g., in parameter balancing, Section 6.1), and help bridge the gap between flux analysis and kinetic models (e.g., in elasticity sampling, Section 10.1.3).

15.7 Multivariate Statistics

Summary

The analysis of genome-wide gene expression data involves basic concepts from multivariate statistics. Most applications can be subdivided in two groups: the first group consists of case-control studies comparing a certain transcriptome state of the biological system (e.g., disease state and perturbed state) against the control situation; the second group of applications consist of multiple case studies involving different states (e.g., drug response time series, and groups of patients). The analysis of case-control studies involves testing of statistical hypotheses. Here, expression changes are observed that deviate from a predefined hypothesis and this deviation is judged for significance. The basic methods for multicase studies are clustering and classification. Here, groups of genes are identified that enable the identification of functionally related groups of genes or experiments. These types of analysis result in the identification of marker

genes and their related interactions that are the basis for further network studies.

15.7.1

Planning and Designing Experiments for Case-Control Studies

The analysis of fold changes is a central part of transcriptome analysis. Questions of interest are whether there are genes that can be identified as being differentially expressed when comparing two different conditions (e.g., a normal versus a disease condition). Whereas early studies of fold change analysis were based on the expression ratio of probes derived from the expression in a treatment and a control target sample, it has been a working standard to perform experimental repetitions and to base the identification of differentially expressed genes on statistical testing procedures judging the null hypothesis $H_0 : \mu_x = \mu_y$ versus the alternative $H_0 : \mu_x \neq \mu_y$, where μ_x, μ_y are the population means of the treatment and the control sample, respectively. Strikingly, it is still very popular to present the expression ratio in published results without any estimate of the error and studies that employ ratio error bounds are hard to find (for an exception see [33]). It should be noted that the use of the fold change without estimates of the error bounds is of very limited information. For example, probes with low expression values in both conditions can have tremendous ratios but these ratios are meaningless because they reflect only noise. A simple error calculation can be done as follows. Assume that we have replicate series for control and treatment series x_1, \dots, x_n and y_1, \dots, y_m . A widely used error of the sample averages is the *standard error of the mean*

$$S_x = \sqrt{\frac{1}{(n-1)n} \sum_{i=1}^n (x_i - \bar{x})^2} \quad \text{and} \\ S_y = \sqrt{\frac{1}{(m-1)m} \sum_{i=1}^m (y_i - \bar{y})^2}. \quad (15.179)$$

The *standard error of the ratio* can then be calculated as

$$\frac{\bar{x}}{\bar{y}} \pm \frac{1}{\bar{y}^2} \sqrt{\bar{x}^2 S_y^2 + \bar{y}^2 S_x^2}. \quad (15.180)$$

An important question in the design of such an experiment is how many replicates should be used and what level of fold change can be detected. This is – among other factors – dependent on the experimental noise. Experimental observations indicate that an experimental noise of 15–25% can be assumed in a typical microarray experiment. The experimental noise can be interpreted as

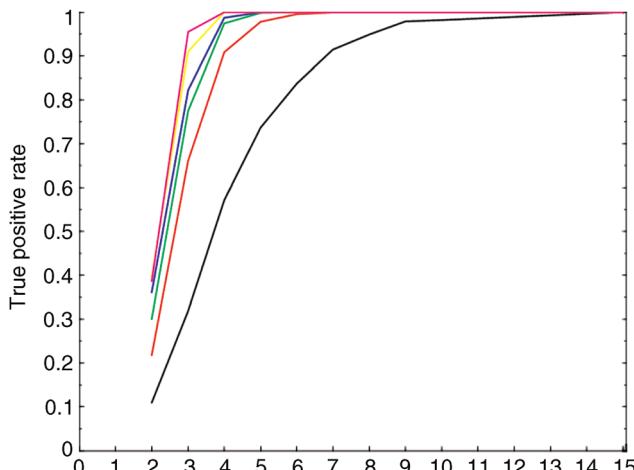


Figure 15.17 Simulation of the dependency of fold change detection from the sample size. Experimental error is assumed to be 20%, that is, CV of replicated control and treatment series equals 0.2. Samples are drawn from Gaussian distributions with mean equal to 1 for the control series and mean equal to 1.5 (black), 2 (red), 2.5 (green), 3 (blue), 5 (yellow), and 10 (magenta) for the treatment samples, respectively in order to simulate the fold changes. Sampling is repeated 1000 times and the proportion of true positive test results ($P < 0.05$) is plotted (Y-axis) over the sample size (X-axis).

the mean CV of replicated series of expression values of the probes. The dependence of the detectable fold change on the number of experimental repetitions and on the experimental error has been discussed in several papers [34,35].

Figure 15.17 shows a simple simulation of this fact. Replicate series are sampled from a Gaussian distribution with mean $\mu = 1$ and $\sigma^2 = 0.04$ (i.e., $CV = 0.2$) for the control series. In order to simulate fold changes the mean of the treatment series is changed subsequently holding the CV constant (e.g., $\mu = 2$ and $\sigma^2 = 0.16$ (i.e., $CV=0.2$) if a fold change of factor two is simulated). Then, replicates are sampled from that distribution. A Welch test is performed and it is marked whether the P -value is significant (<0.05) or not. The sampling is repeated 1000 times and the number of positive test results is denoted. The curves show the dependency of the true positive rate on the sample size. For example, a 1.5-fold change is detectable in only 32% of all cases when three repetitions are used. This number increases to 95% when eight replicates are used (black line). The simulation suggests that a fold-change analysis should be performed with at least four independent replicates.

15.7.2 Tests for Differential Expression

15.7.2.1 DNA Arrays

Let x_1, \dots, x_n and y_1, \dots, y_m be two independent samples derived from replicated measurements of the same probe across two conditions (treatment and control). Differential expression of the gene represented by the probe in the two conditions can be judged by location tests. Here, deviations from the null hypothesis of equal means are judged by test statistics. These test statistics are real-valued functions on the data samples

$$f(x_1, \dots, x_n, y_1, \dots, y_m). \quad (15.181)$$

Test statistics typically follow a certain probability distribution under the null hypothesis, and so, for each given value of the test statistic the probability of an even more extreme observation can be calculated by integrating the respective area under the probability density function. This probability is computed as the P -value that judges the significance of the observation, for example a certain fold change, given the null hypothesis. Thus, observations that have a low P -value indicate that the null hypothesis of equal location of the two samples is very unlikely and should be rejected. These observations are typically called significant results.

It is notable that the P -value is only valid if the distributional assumptions are valid. For example, if a t -test is applied to a single gene observation resulting in a P -value of 0.01, this value is only true if both series are Gaussian distributed and have equal variances. Furthermore, the test assumes that the replicates are independent of each other. Strikingly, there are many studies around that miss this fact entirely, for example, applying a Gaussian-based t -test without checking the validity of the distributional assumptions. Thus, replicates on the same array and replicates in different experiments should not be mixed, since they have different characteristics and cannot be treated as independent replicates. Important issues are given below:

- Are the distributional assumptions valid?
- Are the replicates independent of each other?
- Is the number of replicates sufficient to detect the fold change that you are interested in?
- Are outliers removed from the samples?

Most commonly, modifications of four different tests are applied in microarray data analysis. These tests are standardly implemented in statistical software packages such as R/Bioconductor or SAS:

- 1) Student's t -test
- 2) Welch's test

- 3) Wilcoxon's rank sum test
 4) Permutation tests

While the first two tests assume Gaussian-distributed data and the P -values are calculated by a probability distribution, the latter two are nonparametric and the P -values are calculated with combinatorial arguments.

Example 15.32

In a microarray study incorporating approximately 15 000 different cDNAs and four independent hybridization experiments, the early differentiation event in human blastocysts has been investigated, that is, the formation of the trophectoderm and the inner cell mass [36]. *HMBG1* is a specific gene of interest, because it has been published as a potential stemness gene in human stem cell lines, that is, a gene that is relevant for remaining pluripotency of cells. *HMBG1* is a member of the high mobility group of transcription factor encoding proteins that act primarily as architectural facilitators in the assembly of nucleoprotein complexes, for example, the initiation of transcription factor target genes.

The four measurements are for the trophectoderm and ICM, respectively:

32 612	46 741	29 238	32 671
49 966	58 037	94 785	122 044

P -values are 0.037 for Student's t -test, 0.068 for the Welch test, and 0.029 for the Wilcoxon test. This example shows how a high variance (ICM sample) can mislead the Gaussian-based tests, whereas the rank-based test is fairly stable. Note that ranking separates the groups perfectly.

15.7.2.2 Next-Generation Sequencing

While the differential analysis of DNA arrays has been well developed through recent years, differential analysis of next-generation sequencing data is still under construction. In principle, these analyses can be compared to EST data analyses and many of the methods proposed there can be extrapolated.

Consider, for a given gene, the measurement of a sequence reads in condition A (treatment) and b sequence reads in condition B (control).

We can organize the data in a 2×2 contingency table as shown in Table 15.1.

The question, whether or not the difference in read counts for the particular gene in the two conditions is significant, can be answered by different statistical approaches [37]:

- 1) Fisher's exact test

Table 15.1 Contingency table for tag-based statistical analysis.

	Tags in condition A	Tags in condition B	Total
Tags for this particular gene	a	b	$a+b$
Tags for all other genes	c	d	$c+d$
Total	$a+c$	$b+d$	$a+b+c+d$

- 2) Chi-square test
 3) Bayesian methods [38]

The latter method has been originally proposed in the analysis of EST libraries. It follows a Poisson assumption of the occurrence of a tag for a particular gene in a large population of tags and uses a Bayesian approach to compute the corresponding conditional probabilities. The probability of observing b reads of the gene in condition B (under the null hypothesis of equal distribution) given that a reads have been observed in condition A is expressed as

$$P(b|a) = \left(\frac{b+d}{a+c} \right)^b \frac{(a+b)!}{a!b!(1 + ((b+d)/(a+c)))^{a+b+1}}. \quad (15.182)$$

Resulting P -values are computed by

$$P = \min \left\{ \sum_{k=0}^b P(k|a), \sum_{k=b}^{\infty} P(k|a) \right\}. \quad (15.183)$$

15.7.3 Multiple Testing

The single gene analysis described above has statistically a major drawback. We cannot view each single test separately but have to take into account the fact that we perform thousands of tests in parallel (for example, for each gene on an array). Thus, a global significance level of $\alpha = 0.05$, for example, performed with $n = 10,000$ cDNAs will imply a false positive rate of 5%. This means that we must expect that 500 (!) individual tests are false positive results and thus that many cDNAs are falsely identified as potential targets. Inclusion of such false positives in the further analysis steps can be extremely costly. Therefore, corrections for multiple testing are commonly applied to microarray studies that assure a global significance rate of 5%.

Let α_g be the global significance level and α_s be the significance level at the single-gene level. It is clear that we cannot assure a global significance level α_g without adjusting the single-gene levels. For example, the probability of making the correct decision given that we reject

the null hypothesis (i.e., the probability of selecting a truly differentially expressed gene) is

$$p_s = (1 - \alpha_s). \quad (15.184)$$

The probability of making the correct decision on the global level is the product of the probabilities on the individual levels

$$p_g = (1 - \alpha_s)^n. \quad (15.185)$$

The probability of drawing the wrong conclusion in either of the n different tests is

$$P(\text{wrong}) = \alpha_g = 1 - (1 - \alpha_s)^n. \quad (15.186)$$

For example, if we have 100 different genes on the array and we set the gene-wise significance level to 0.05, we will have a probability of 0.994 of making a type-I error. This is so-called *family-wise error rate* (FWER) of the experiment, that is, the global type-I error rate. Multiple testing corrections try to adjust the single-gene level type-I error rate in a way that the global type-I error rate is below a given threshold. In practice, it means that the calculated P -values have to be corrected.

The most conservative correction is the *Bonferroni correction*. Here, we approximate (15.186) by the first terms of the binomial expansion, that is,

$$(1 - \alpha_s)^n = \sum_{i=0}^n \binom{n}{i} (-\alpha_s)^i \approx n\alpha_s. \quad (15.187)$$

Thus, we rewrite

$$\alpha_g = 1 - \sum_{i=0}^n \binom{n}{i} (-\alpha_s)^i \approx n\alpha_s \Rightarrow \alpha_s = \frac{\alpha_g}{n}. \quad (15.188)$$

The Bonferroni correction of the single-gene level is the global level divided by the number of the tests performed. This is far too conservative. For example, using an array of $n = 10\,000$ probes and an experiment FWER of 0.01, then only those observations would be judged as “significantly differentially expressed” whose P -value is below $1.0e-06$. Fairly, few genes would meet this requirement. The result would therefore consist of many true negatives.

The Bonferroni correction is too strict in the sense that we apply the same significance level to all genes. Consider now the following step-wise procedure:

For a given global significance level α_g , sort the probes in increasing order after their P -values calculated on the single-gene basis. If $p_1 < \frac{\alpha_g}{n}$, then adjust the remaining $n-1$ P -values by comparing the next P -value $p_2 < \frac{\alpha_g}{n-1}$ and so on. If m is the largest integer for which $p_m < ((\alpha_g)/(n-m+1))$, then we call genes $1, \dots, m$ significantly differentially expressed. This procedure is

called *Holm's stepwise correction* and it assures that the global significance level is valid. Although it is more flexible than the Bonferroni correction, it is still too strict for practical purposes.

A widely used method for adjusting P -values is the *Westfall and Young step-down correction* [39]. This procedure is essentially based on permutations of the data.

- 1) Perform $d=1, \dots, D$ permutations of the sample labels, and let p_i be the gene-wise P -value of the i th probe
- 2) For each permutation, compute the P -value p_{id} from the d th permutation for the i th probe
- 3) Adjust the P -value of probe i by $\tilde{p}_i = \frac{\#\{d: \min_d p_{id} \leq p_i\}}{D}$

The advantage of this resampling method is that, unlike the approaches above, it takes into account data dependencies.

An alternative to controlling the FWER is the computation of the *false discovery rate* (FDR). The FDR is defined as the expected number of type-I errors among the rejected hypotheses [40].

The procedure follows the scheme

- 1) As in the case of Holm's procedure, sort the probes in increasing order after their P -values calculated on the single-gene basis. Select a level α_g for the FDR.
- 2) Let $j^* = \max\{j; p_j \leq j\alpha_g/n\}$.
- 3) Reject the hypotheses for $j = 1, \dots, j^*$.

Recent variations of controlling the FDR with application to microarray data have been published in Refs. [41,42].

The practical use of multiple testing is not entirely clear. Whereas it is useful to select false positive results from true positive results on the one hand, it will on the other hand discard a lot of potentially useful targets and the experimentalist might lose important biological information.

15.7.4 ROC Curve Analysis

In Section 15.3 we introduce the basic types of errors of a statistical test procedure. If, in practice, a training sample is available (for example, a set of gene probes known to be differentially expressed and a set of probes that is known to be unchanged), we can for each test result calculate the true and false positive rates. The performance of a specific test, normalization method, and so on can then be display using a ROC (*receiver operating characteristic*) curve. The purpose and result of a ROC curve analysis is, for example, to evaluate several normalization and test procedures and to choose the best methods.

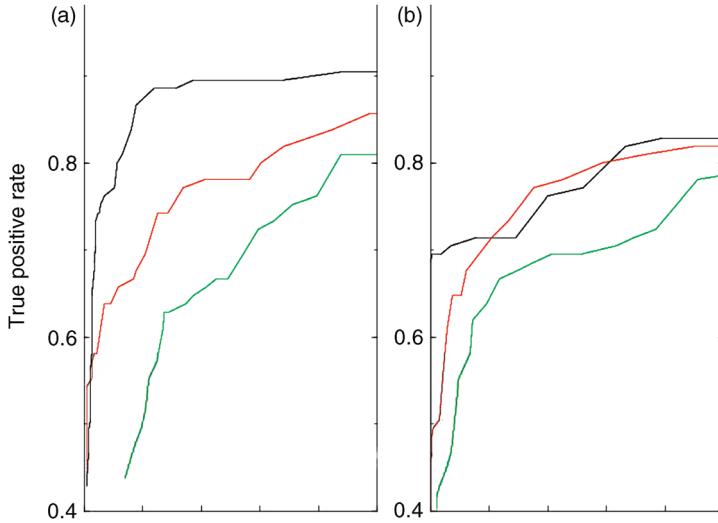


Figure 15.18 ROC curve for visualizing performance of normalization methods and test procedures. Six independent hybridization experiments were performed with wild-type zebrafish embryos (control) and lithium-treated embryos (treatment). The true positive sample was identified by 105 cDNAs that were verified by an independent experimental technique (*in-situ* hybridization), the false positive sample was estimated by 2304 copies of an *Arabidopsis thaliana* cDNA whose complementary sequence was spiked to the treatment and control target samples. (a) The van der Waerden test is used for judging differential expression on three different normalization methods: global median normalization (black), variance stabilization (red), and linear regression (green). (b) Student's *t*-test is used for judging differential expression using the same normalization methods. ROC analysis reveals that the nonparametric test outperforms the Gaussian-based test and furthermore, that the global normalization performs best with both test methods compared to the other methods.

Figure 15.18 shows a typical example of a ROC curve analysis. Here, we map the false positive rate (X-axis) and the true positive rate (Y-axis) and compare the performance of three normalization procedures and two statistical tests on an experimental test set with known expression changes. Ideally, the ROC curve has an integral of one and is a straight line (no false positives, maximal sensitivity) and those procedures are preferable that give the highest overall integral. Alternatively, one might select a specific area of interest (for example, a false positive rate below the experimental significance level) and chose that procedure that shows the highest performance in the selected area. Similar ROC curve analysis has been used to compare different normalization strategies [43].

15.7.5

Clustering Algorithms

Clustering algorithms are a general group of tools from multivariate explorative statistics. They are used to group data objects according to their pairwise similarity with respect to a set of characteristics measured on these objects. Clustering algorithms are widely used in order to identify coregulated genes with microarray experiments. There is a simple assumption behind that strategy – the concept of *guilt-by-association*. The ratio behind this concept is that those genes whose probes show a similar

profile through a set of experimental conditions will share common regulatory rules. Thus, gene expression clusters are used to identify common functional characteristics of the genes.

Clustering algorithms are explorative statistical methods that group together genes with similar profiles and separate genes with dissimilar profiles, whereby similarity (or dissimilarity) is defined numerically by a real-valued pairwise (dis)similarity function. Consider p experiments have been performed on n different genes on the array, then the profile of gene i is a p -dimensional vector $x_i = (x_{i1}, \dots, x_{ip})$ and a *pairwise similarity measure* can be any function $d : \mathcal{R}_p \times \mathcal{R}_p \rightarrow \mathcal{R}$. Intuitively, one would prefer functions that reflect some kind of geometric distance such as the *Euclidean distance*, or more general, *Minkowsky-* or ℓ^q -distances defined by

$$d_q(x_n, x_m) = \left(\sum_{i=1}^p |x_{ni} - x_{mi}|^q \right)^{\frac{1}{q}}. \quad (15.189)$$

Note that for $q=1$ we have the *Manhattan distance* and for $q=2$ we have the Euclidean distance. Another class of pairwise similarity measures is correlation measures such as Pearson's- or Spearman's correlation coefficient.

A practical problem occurs with *missing values*, since there may be some measurements that yield an unreliable

value for a given probe. However, one wants to keep the other reliable measurements of that probe and use its profile in further analysis. The fact that the profile now consists only of $p - 1$ values has to be taken into account. The treatment of missing values is a characteristic of the pairwise similarity measure. For example, one could try to estimate the distance of two vectors with missing values by the valid values. Assume two vectors, $\mathbf{x}_n, \mathbf{x}_m$, then the squared Euclidean distance is given by $d_2^2(\mathbf{x}_n, \mathbf{x}_m) = \sum_{i=1}^p (x_{ni} - x_{mi})^2$ if both vectors have no missing values. If there are missing values, count the number of coordinate pairs that include at least one missing value, k , compute the distance on the remaining coordinate pairs and estimate the distance by a multiplicative factor proportional to the amount missing pairs, that is, $d_2^2(\mathbf{x}_n, \mathbf{x}_m) = \frac{p}{p-k} \sum_i (x_{ni} - x_{mi})^2$. If for example, half of the data is missing the remaining distance is multiplied with 2. Such and other adjustments for missing data can be found in the book of Jain and Dubes [44].

Example 15.33

In practice, it might be useful to transform data prior to computing pairwise distances. Consider the profiles $\mathbf{x}_1 = (100, 200, 300)$, $\mathbf{x}_2 = (10, 20, 30)$, and $\mathbf{x}_3 = (30, 20, 10)$. Euclidean distance will assign a higher similarity to the pair $\mathbf{x}_2, \mathbf{x}_3$ than to the pair $\mathbf{x}_1, \mathbf{x}_2$, because it only takes into account the geometric distance of the three data vectors. Correlation measures would assign a higher similarity to the pair $\mathbf{x}_1, \mathbf{x}_2$ than to the pair $\mathbf{x}_2, \mathbf{x}_3$ since they take into account whether the components of both vectors change in the same direction. For example, if these data were derived from a time series measurement one would argue that both vectors $\mathbf{x}_1, \mathbf{x}_2$ increase with time (although on different levels of expression) whereas \mathbf{x}_3 decreases with time. In many applications, thus, it makes sense to transform the data vectors before calculating pairwise similarities. A straightforward geometric data transformation would be to divide each component x_j of a p -dimensional data vector $\mathbf{x} = (x_1, \dots, x_p)^T$ by its Euclidian norm, that is, perform the transformation $\tilde{\mathbf{x}}_j = (x_j / \|\mathbf{x}\|)$. The resulting effect is that after transformation each data vector $\tilde{\mathbf{x}}$ has an Euclidean norm of one and is mapped to the unit sphere.

The choice of the similarity measure is important since it influences the output of the clustering algorithm. It should be adapted to the question of interest. Figure 15.19 shows two different clustering results using two different distances.

Two classes of clustering algorithms are commonly distinguished, *hierarchical* and *partitioning* methods [44]. In contrast to partitioning methods that try to find the “best” partition given a fixed number of clusters, hierarchical methods calculate a full series of partitions starting from n clusters each of which contains one single data point and ending with one cluster that contains all n data points (or vice versa); in each step of the procedure, two clusters are merged according to a prespecified rule. In the following, we describe the classical hierarchical algorithm and commonly used partitioning methods (SOM and K -means).

15.7.5.1 Hierarchical Clustering

Let $\mathbf{x}_1, \dots, \mathbf{x}_n$ be the p -dimensional data points (expression profiles of n gene representatives across the p experiments). The process of hierarchical algorithms requires a dissimilarity measure, d , between pairs of clusters (related to a dissimilarity measure, \tilde{d} , between pairs of data points) and an update procedure for recalculation of the merged clusters. It has then the following scheme:

- 1) For $\nu = n$ start with the finest possible partition.
- 2) Calculate a new partition by joining two clusters that minimize d .
- 3) Update the distances of the remaining clusters and the joined cluster.
- 4) Stop, if $\nu = 1$, that is, all data points are in one cluster otherwise repeat steps 1–3.

Several cluster dissimilarity measures are in use:

$$\text{Single linkage } d(C_k^{(\nu)}, C_l^{(\nu)}) = \min_{x_i \in C_k^{(\nu)}, x_j \in C_l^{(\nu)}} \tilde{d}(\mathbf{x}_i, \mathbf{x}_j). \quad (15.190)$$

$$\text{Complete linkage } d(C_k^{(\nu)}, C_l^{(\nu)}) = \max_{x_i \in C_k^{(\nu)}, x_j \in C_l^{(\nu)}} \tilde{d}(\mathbf{x}_i, \mathbf{x}_j). \quad (15.191)$$

$$\text{Average linkage } d(C_k^{(\nu)}, C_l^{(\nu)}) = \frac{1}{|C_k^{(\nu)}||C_l^{(\nu)}|} \sum_{x_i \in C_k^{(\nu)}, x_j \in C_l^{(\nu)}} d(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_j). \quad (15.192)$$

Here, $C_i^{(\nu)}$, denotes the i th cluster at the ν th iteration step ($i = k, l$). In the single-linkage procedure, the distance of two clusters is given by the minimal pairwise distance of the members of the first and second cluster. In the complete-linkage procedure, the distance of two clusters is given by the maximal pairwise distance of the members of the first and second cluster. In the average-linkage procedure, the distance of two clusters is given by the pairwise distance of the arithmetic means of the clusters. In all three procedures, those two clusters with the minimal cluster distance over all possible pairs of clusters are merged.

Once two clusters have been merged to a new cluster, the distances to all other clusters must be recomputed.

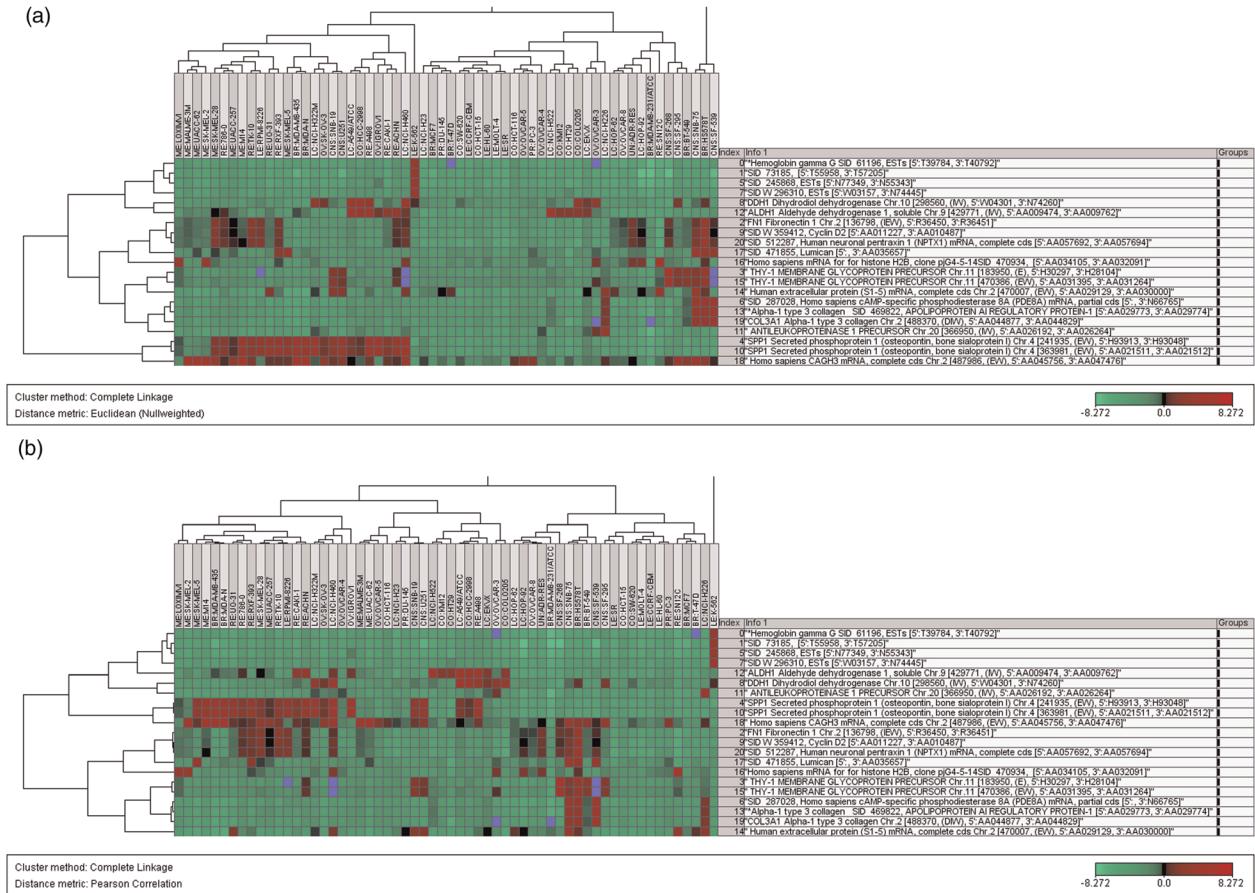


Figure 15.19 Influence of similarity measure on clustering. Two dendrograms of a subgroup of genes using the microarray expression data of Ross *et al.* [45] were generated using hierarchical clustering with Euclidean distance (a) and Pearson correlation (b) as pairwise similarity measure. Although all other parameters are kept constant, results show differences in both gene and cancer cell line groupings. Clustering is performed with the *J-Express Pro* software package (Molmine, Bergen Norway).

This is usually implemented using the following recursive formula:

$$d(C_m^{(\nu-1)}, C_k^{(\nu)} \cup C_l^{(\nu)}) = \alpha_k d(C_m^{(\nu)}, C_k^{(\nu)}) + \alpha_l d(C_m^{(\nu)}, C_l^{(\nu)}) \\ + \beta d(C_k^{(\nu)}, C_l^{(\nu)}) + \gamma |d(C_m^{(\nu)}, C_l^{(\nu)}) - d(C_m^{(\nu)}, C_k^{(\nu)})|, \quad (15.193)$$

where the parameters depend on the cluster distance measure. The parameters for the update procedure are summarized in Table 15.2.

Table 15.2 Parameters in hierarchical clustering.

Method	α_i ($i = k, l$)	β	γ
single linkage	0.5	0	-0.5
complete linkage	0.5	0	0.5
Average linkage	$\frac{ C_i^{(\nu)} }{ C_k^{(\nu)} + C_l^{(\nu)} }$	0	0

The parameter β is zero in these examples but there exist other update methods (for example, the centroid and the Ward method) that incorporate a positive β .

Hierarchical methods have been applied in the context of clustering gene-expression profiles [46]. They are memory intensive with increasing data size because all pairwise distances must be calculated and stored. Hierarchical methods suffer from the fact that they do not “repair” false joining of data points from previous steps, indeed they follow a determined path for a given rule. Figure 15.20 displays this problem. In a recent study [33], the gene-expression profiles of chromosome 21 mouse orthologues were studied in nine different mouse tissues from a mouse model for trisomy 21 (TS65Dn mouse). Among genes predominantly active in the brain they found *DSCAM*, a cell surface protein acting as an axon guidance receptor. A hierarchical clustering using average linkage as an update rule was performed and the resulting dendrogram is displayed. This cluster is significantly

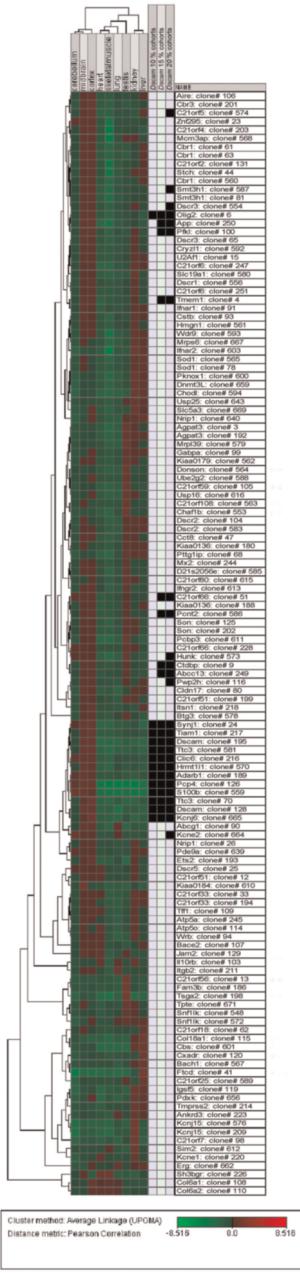


Figure 15.20 Practical example of a dendrogram from nine different mouse tissues. For each cDNA the logarithm (base 2) of the ratio between the normalized intensity in the specific tissue and the average of intensities of this cDNA across the nine control tissues was calculated. Ratios were represented with a color gradient spanning from green (underexpressed) to red (overexpressed). Hierarchical clustering was performed with the average-linkage update rule and Pearson correlation as similarity measure (J-Express, Molmine, Bergen Norway). In addition, clones with the most similar expression profiles to *Dscam* (with respect to the Pearson correlation) are displayed: 10%-closest (13 clones, left column), 15%-closest (20 clones, middle column), and 20%-closest (26 clones, right column). Note that in hierarchical clustering procedures, clones with similar expression profiles can be split to different parts of the dendrogram (e.g., *Olig2*), and vice-versa (e.g., *Abcg1*).

nonrandom; however, several of the profiles numerically close to *DSCAM* are missing (black bars) due to false joining in previous steps.

Another problem with hierarchical clustering methods is that it may be difficult to decide on a representative member for each cluster, especially when using the single-linkage algorithm. In contrast when a partitioning algorithm is used, the center of each cluster is a natural representation of the cluster's feature.

15.7.5.2 Self-Organizing Maps (SOMs)

Clustering methods are implicitly used in the construction of self-organizing maps (SOMs), a method in the neural network framework introduced by Kohonen [47]. Kohonen's algorithm tries to find an illustrative display of n -dimensional data points in a given lattice, L , of points, usually in two or three dimensions such that the high dimensional data structure (neighborhoods, topological ordering, and clusters) are preserved and can be detected in this low dimensional structure. The points $r_i \in L$ are called nodes (neurons). Each node r_i has a representation in the n -dimensional space of the data points; this representation is called reference vector, c_j , (or weight vector of the neuron). Basically, there are two main steps that are repeated for each data vector for a number of iterations, in the order of tens of thousands iterations.

- 1) Randomly initialize the reference vector $c_j^{(1)}$ for each node
- 2) For each iteration step $\nu + 1$ do:
 - a) Randomly pick an input data vector $x_{\nu+1}$. Denote by $c_j^{(\nu)}$ the weight vector of the j th node at iteration ν . The matching node is defined by $c_{j_0}^{(\nu)} \in \arg \min \{d_2(x_{\nu+1}, c_j^{(\nu)}) ; j\}$, where d_2 is defined in (15.189).
 - b) Update the reference vector of the matching node and its neighbors by the update formula $c_j^{(\nu+1)} = c_j^{(\nu)} + \eta^{(\nu)} h_{j_0j}^{(\nu)} (x_{\nu+1} - c_j^{(\nu)})$.
- 3) Assign each data vector to the cluster with the most similar reference vector.

$0 < \eta^{(\nu)} < 1$ is called the learning function, which monotonically decreases with the number of iterations; $0 < h_{j_0j}^{(\nu)} < 1$ is called the neighborhood function, which decreases monotonically with the distance of the nodes.

The main task of the neighborhood function is to provide learning, that is, updating of the weights, not only for the best matching node, but also for its neighbors. The task of the learning function is to shrink in time as iterations increase.

The result of Kohonen's algorithm is that units that are spatially close tend to develop similar weight vectors. Of course, the rate at which the neighborhood shrinks is critical. If the neighborhood is large and it shrinks slowly, the cluster centers will tend to stick close to the overall mean of all of the samples.

Commonly used neighborhood functions are $h_{j_0j}^{(v)} = e^{-(d_2(r_{j_0}, r_j)^2)/(2\sigma^2(v))}$ and $h_{j_0j}^{(v)} = \begin{cases} 1, & d_2(r_{j_0}, r_j) < \sigma(v) \\ 0, & d_2(r_{j_0}, r_j) \geq \sigma(v) \end{cases}$, where

r_{j_0} is the matching node, r_j is the adapted node whose reference vector is updated and $\sigma^2(v)$ is the neighborhood radius, that is decreasing with the number of iterations. Self-organizing maps have been used in the context of clustering gene-expression profiles in Refs. [48,49].

15.7.5.3 K-Means

K -means algorithms are a fast and large-scale applicable clustering method. The main idea behind these techniques is the optimization of an objective function usually taken up as a function of the deviates between all patterns of the data points from their respective cluster centers. The most commonly used optimization is the minimization of the within-cluster sum of squared Euclidean distances utilizing an iterative scheme, which starts with a random initialization of the cluster centers, then alters the clustering of the data to obtain a better value of the objective function.

K -means algorithms alternate between two steps until a stopping criterion is satisfied. These steps are a pairwise distance measure of the data vectors and the cluster centers related to the optimization criterion and an update procedure for the cluster centers.

Euclidean distance has been used in most cases as pairwise similarity measure because of its computational simplicity. The cluster center at each iteration can be calculated in a straightforward manner by the arithmetic mean of the data vectors currently assigned to the cluster, which is known to minimize the within-cluster sum of squared Euclidean distances.

The original K -means algorithm reads:

- 1) Start with an initial partition of the data points in K cluster with cluster centers $\mathbf{c}_1^{(1)}, \dots, \mathbf{c}_K^{(1)}$ and let $W^{(1)}$ be the value of the initial objective function.
- 2) At the v th step of the iteration, assign each data point to the cluster with the lowest pairwise distance.
- 3) Recompute the cluster centers $\mathbf{c}_1^{(v+1)}, \dots, \mathbf{c}_K^{(v+1)}$ by minimizing $W^{(v+1)}$.
- 4) If for all k , $|\mathbf{c}_k^{(v)} - \mathbf{c}_k^{(v+1)}| < \epsilon$ stop; else go to step 2.

- 5) Assign each data vector to the nearest cluster center.

If the pairwise distance is defined as the Euclidean distance, the algorithm minimizes the within-cluster sum of squares of the K clusters. In this case, the cluster centers at every iteration are recomputed as the arithmetic means of the respective data points. Other pairwise distance measures are the l_1 -metric (K -median clustering) and the l_∞ -metric (K -midranges clustering). A common criticism on K -means algorithms focuses on the fact that the number of centers has to be fixed from beginning of the procedure. Furthermore, the results are highly dependent on the initialized set of centers. Alternative algorithms have been published that do not require to determine the number of clusters in advance and thus overcome this criticism [50].

A simple approach of refining the K -means algorithm employs two thresholding parameters (*sequential K-means*). The original idea dates back to MacQueen [51]. A parameter ρ controlling the distance within the clusters is used to define new cluster centers and a parameter σ controlling the distance between cluster centers is used to merge cluster centers. The algorithm reads as follows:

- 1) Initialize K cluster centers $\mathbf{c}_1^{(1)}, \dots, \mathbf{c}_K^{(1)}$.
- 2) Select a new data point \mathbf{x}_i at the $v + 1$ th step.
- 3) Compute the distances to all cluster centers from the previous step $\mathbf{c}_1^{(v)}, \dots, \mathbf{c}_K^{(v)}$. Let $w_1^{(v)}, \dots, w_K^{(v)}$ be the weights of the clusters in that step, that is, the number of data points already assigned to the cluster centers.
If $\min\{d(\mathbf{c}_j^{(v)}, \mathbf{x}_i); j = 1, \dots, K\} < \rho$ then
 - a) assign \mathbf{x}_i to the cluster center with the minimal distance, $\mathbf{c}_{j_0}^{(v)}$ and update the centroid and its weight by $w_{j_0}^{(v+1)} = w_{j_0}^{(v)} + 1$ and $\mathbf{c}_{j_0}^{(v+1)} = \frac{w_{j_0}^{(v)} \mathbf{c}_{j_0}^{(v)} + \mathbf{x}_i}{w_{j_0}^{(v+1)}}$.
 - b) Compute the distance of the updated center to each of the other cluster centers. While $\min\{d(\mathbf{c}_j^{(v)}, \mathbf{c}_{j_0}^{(v+1)}); j = 1, \dots, K\} < \sigma$, merge the center with the minimal distance and update again according to (a). Repeat this step until for all centers there is $d(\mathbf{c}_j^{(v)}, \mathbf{c}_{j_0}^{(v+1)}) \geq \sigma$.
If $\min\{d(\mathbf{c}_j^{(v)}, \mathbf{x}_i); j = 1, \dots, K\} \geq \rho$, initialize a new cluster center by $\mathbf{c}_{K+1}^{(v+1)} = \mathbf{x}_i$ and $w_{K+1}^{(v+1)} = 1$.
- 4) Reclassify the data points.

The above algorithm iteratively allows to join clusters that are similar to each other and to initialize new cluster

centers in each step of the iteration and is thus a very flexible alternative. It should be pointed out that K -means algorithms are not very stable in their solutions, that is, running the same algorithm with different parameters will lead to different results. Thus, this algorithm should be applied not just one time on the data set but rather several times with several initializations of cluster centers. In a postprocessing step, the stable clusters can then be retrieved.

15.7.6 Cluster Validation

Many clustering algorithms are currently available each of which claims special merits and has some interpretation that makes it suitable for a class of applications. However, it is important to compare the output of cluster algorithms in order to decide which one gives best results for the current problem. For that purpose, *cluster validation measures* are used. In principle, two groups of measures can be separated, external and internal measures. *External validation measures* incorporate a priori knowledge on the clustering structure of the data, for example in simulation experiments when the true partition of the data is known, or in real experiments when specific gene clusters are known. Typically, an external cluster validation measure is a numerical function that evaluates two different groupings of the same data set. This is done by the following scheme:

Assume that we have n p -dimensional data vectors $\mathbf{x}_1, \dots, \mathbf{x}_n$ and that a clustering result generates a partition of this data set in disjoint subsets. This is implicitly done with partitioning algorithms, whereas with hierarchical algorithms the dendrogram has to be cut in a suitable postprocessing step. Each partition can be represented by a binary $n \times n$ -partitioning matrix, $C = (c_{ij})$, with

$$c_{ij} = \begin{cases} 0, & \text{if data vectors } i \text{ and } j \text{ are not in the same cluster} \\ 1, & \text{if data vectors } i \text{ and } j \text{ are in the same cluster} \end{cases}$$

Let, \mathbf{C} and \mathbf{T} be two partitioning matrices computed from two different clustering algorithms, then most external indices are defined as numerical functions on the 2×2 contingency table (Table 15.3).

Here, n_{11} denotes the number of pairs that are in a common cluster in both partitions, $n_{1.}$ and $n_{.1}$ are the marginals of the partition matrices \mathbf{T} and \mathbf{C} , respectively. Likewise, the other cell entries are defined. A commonly used index is, for example, the *Jaccard coefficient*

$$J(T, C) = \frac{n_{11}}{n_{11} + n_{01} + n_{10}}, \quad (15.194)$$

Table 15.3 Contingency table for judging clustering quality.

C/T	0	1	Total
0	n_{00}	n_{01}	$n_{0.}$
1	n_{10}	n_{11}	$n_{1.}$
Total	$n_{.0}$	$n_{.1}$	n^2

that measures the data pairs clustered together proportionally to the marginals. Other examples are Hubert's Γ statistic, the goodness-of-fit statistic or measures based on information theory [44].

Internal validation measures compare the quality of the calculated clusters solely by the data itself. Indices of quality are topological concepts, for example, compactness or isolation, that are computed by numerical functions, information theoretic concepts that quantify, for example, high informative clusters and variance concepts that quantify the overall variance explained by the cluster. A widely used topological measure is the *Silhouette index* [55]. Consider a clustering of n data vectors that results in K clusters, S_1, \dots, S_K . For each data vector, \mathbf{x}_i , we can calculate two topological values. Let S_l be the cluster that is assigned to \mathbf{x}_i , then the *compactness* value describes the average distance of \mathbf{x}_i to all other data points in the same cluster, that is,

$$a_i = \frac{1}{|S_l| - 1} \sum_{\mathbf{x}_k \in S_l, k \neq i} d(\mathbf{x}_i, \mathbf{x}_k), \quad (15.195)$$

where d is a suitable distance measure. The *isolation* value describes the minimal average distance to all other clusters, that is,

$$b_i = \min \left\{ \frac{1}{|S_j|} \sum_{\mathbf{x}_k \in S_j} d(\mathbf{x}_i, \mathbf{x}_k); j = 1, \dots, K, j \neq l \right\}. \quad (15.196)$$

The compactness (isolation) of a cluster is defined as the average compactness- (isolation-) values of its cluster members. Apparently, clusters of high quality are compact and isolated. The Silhouette index combines compactness and isolation by

$$SI(\mathbf{x}_i) = \frac{b_i - a_i}{\max\{a_i, b_i\}}. \quad (15.197)$$

The value of the Silhouette Index is bound to the interval $[-1, 1]$. Negative values indicate that this data vector

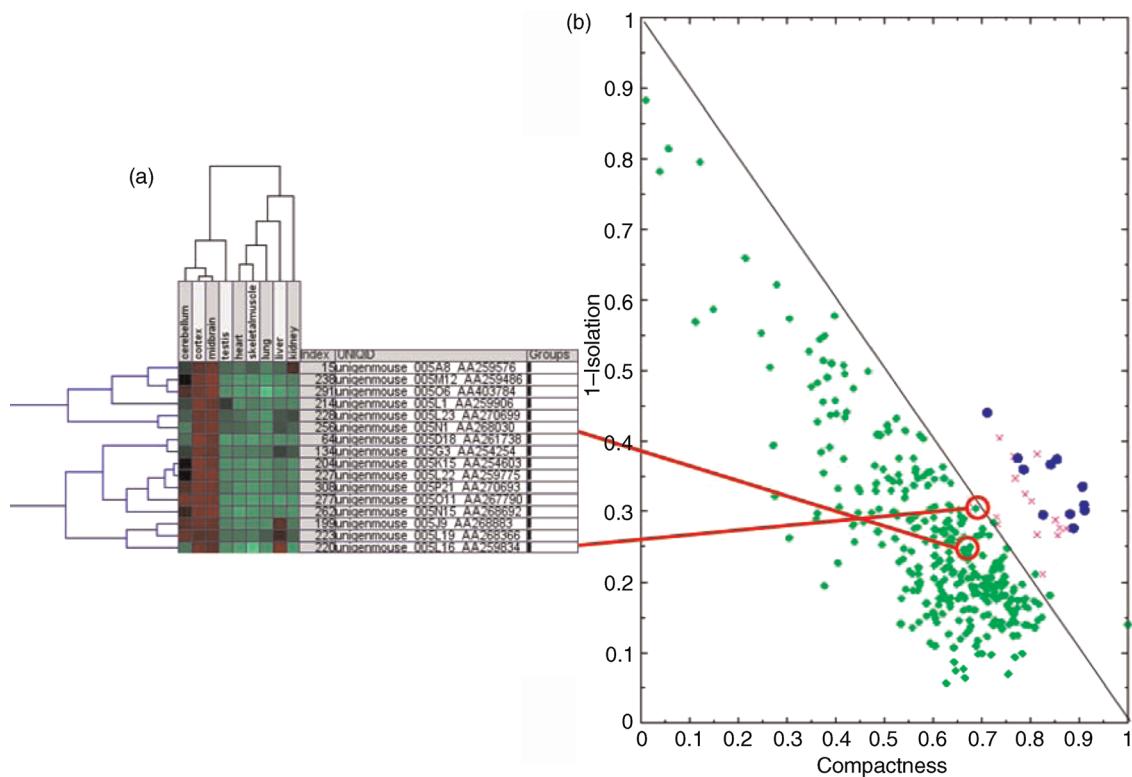


Figure 15.21 Visualization of cluster quality. (a) cluster of cDNA sequences that have a tissue-specific expression in brain. In this study, nine different tissues have been compared in the mouse using a whole-genome approach. Genes in that specific cluster show a high expression in three brain regions (cortex, cerebellum, and midbrain). (b) Compactness (X-axis) and isolation (Y-axis) can be used to visualize the cluster quality in a two-dimensional plot. Crosses represent the brain-specific cDNAs, and circles represent another cluster of liver-specific sequences. Green diamonds represent random assignments of compactness and isolation. The visualization allows the identification of false-positives in each cluster what can enhance follow-up analyses, for example, with respect of promoter searches.

should belong to a different cluster rather than the computed one. Figure 15.21 shows a visualization of the above measures.

Cluster validation is an important topic that has drawn insufficient attention in gene expression analysis. Currently, the situation is somewhat troublesome for the user of clustering software packages. On the one hand, there is the choice between a multitude of sophisticated algorithms, multiple algorithmic parameters, and visualization tools. However, each of these methods will generate a different result contributing to the confusion and frustration of the user and there are too few tools that validate and compare results and select the best one. Thus, future research will focus on the comparison and integration of different methods in order to reduce the bias of the individual methods.

15.7.7 Overrepresentation and Enrichment Analyses

Essentially, case-control studies (Section 15.7.2) and multicase studies (Section 15.7.6) result in lists of genes that

are significant for the study under analysis and functional categorization of these lists of genes is a fundamental issue. Adding functional attributes to the selected genes gives a first impression of the molecular interactions that are involved in the problem. Since more and more functional annotations are available for genes, for example, through the GeneOntology Consortium [53] or through many different pathway databases, these annotations are used to validate clustering results.

Measuring the statistical significance of functional categorization is essentially based on a simple statistical method. Consider a cluster of n genes out of which k genes belong to a certain functional class and $n - k$ belong to different classes. Let N be the total number of genes under analysis and K be the total number of genes annotated for that class. Is the observed number untypical or does it rather express a random distribution of the specific functional class? This problem can be translated into an urn model. If the n genes were randomly drawn from the total of N genes, then the probability of having exactly k out of K genes from the functional class would be given by the

Example 15.34

In a recent work, a core set of 213 marker genes with respect to type-2 diabetes mellitus has been identified with a meta-analysis across different microarray resources [54]. Enrichment analyses based on the hypergeometric distribution were carried out in order to assess whether the candidate list is overrepresented with respect to specific biochemical pathways.

The results show that the fundamental pathways associated with type-2 diabetes mellitus (annotated in the KEGG) database are significantly enriched by the selected marker sets such as PPAR signaling and Insulin signaling. These kinds of overrepresentation analyses are often used to judge a specific selection of marker genes or – vice versa – to identify new pathways potentially relevant for the disease under study (Table 15.4).

Table 15.4 Overrepresentation analysis.

Pathway ID	SigSet	Set	Sig	All	P-value	Q-value	Pathway description
path:mmu03320	13	69	213	15 274	1.02E-11	1.37E-09	PPAR signaling pathway
path:mmu04920	12	73	213	15 274	3.46E-10	1.66E-08	Adipocytokine signaling pathway
path:mmu04930	10	44	213	15 274	3.69E-10	1.66E-08	Type-II diabetes mellitus
path:mmu04910	13	128	213	15 274	2.70E-08	9.09E-07	Insulin signaling pathway
path:mmu04612	6	38	213	15 274	1.30E-05	0.000 351	Antigen processing and presentation
path:mmu00280	6	44	213	15 274	3.11E-05	0.000 697	Valine, leucine and isoleucine deg.
path:mmu04610	7	67	213	15 274	3.98E-05	0.000 764	Complement and coagulation casc.

All are the genes under consideration, Sig the number of candidate genes, Set is the number of genes in the pathway under study, and SigSet the overlap of genes in the pathway and the candidate genes. P-values were computed with the upper tail of the hypergeometric distribution indicating the probability of observing this overlap by chance. Q-values are the multiple testing corrected P-values.

hypergeometric distribution

$$P(k) = \frac{\binom{K}{k} \binom{n-K}{m-k}}{\binom{n}{m}}. \quad (15.198)$$

The P-value for the cluster can then be calculated as the probability of having more than the observed number of hits of that functional group using (15.200), that is,

$$p = \sum_{j \geq k} P(j). \quad (15.199)$$

Overrepresentation analysis described in (15.200) does not take into account the specific alterations of a gene in the experiments so that each gene in the list is weighted equally. In contrast, gene enrichment analysis weights the genes according to the experimental observations (fold-change, expression differences, etc.). A simple approach is, for example, a weighing of the gene according to its fold-change and significance P-value (cf. Section 15.7.2). Let p_i be the P-value and r_i be the ratio of the gene with respect to a case-control study, then the quantity

$$|\log_{10}(p_i)| |\log_2(r_i)| \quad (15.200)$$

is an indicator for the influence of the gene in the study. These scores can be used for gene-enrichment analysis in order to extrapolate the importance of entire pathways for the study. An example is shown in Figure 15.22.

Gene-enrichment scores have been proposed in order to extrapolate differential analysis of genes to the differential analysis of entire pathways. A statistical argument based on a nonparametric, robust hypothesis test has been proposed in Ref. [36]. Here, array data were used to test whether specific pathways showed differential expression in human blastocyst differentiation. Pathways were taken from the KEGG database. Consider for each pathway i , the set of related genes $(x_{i1}, y_{i1}), \dots, (x_{in_i}, y_{in_i})$. Here, x_{ij}, y_{ij} denote the expression level of the j th gene in two different samples (case control). Wilcoxon's matched pairs signed rank test was used to calculate a Z-score for the differences $d_{ij} = x_{ij} - y_{ij}$ for each pathway i . These differences were ranked, and the ranks of differences with negative signs, R_{neg} , and those with positive signs, R_{pos} , were summed. The test statistic is the smaller of the two numbers,

$$R = \min\{R_{\text{pos}}, R_{\text{neg}}\}. \quad (15.201)$$

If the pathway is not affected by the treatments R_{neg} and R_{pos} will be fairly equal, but, if there is a trend of

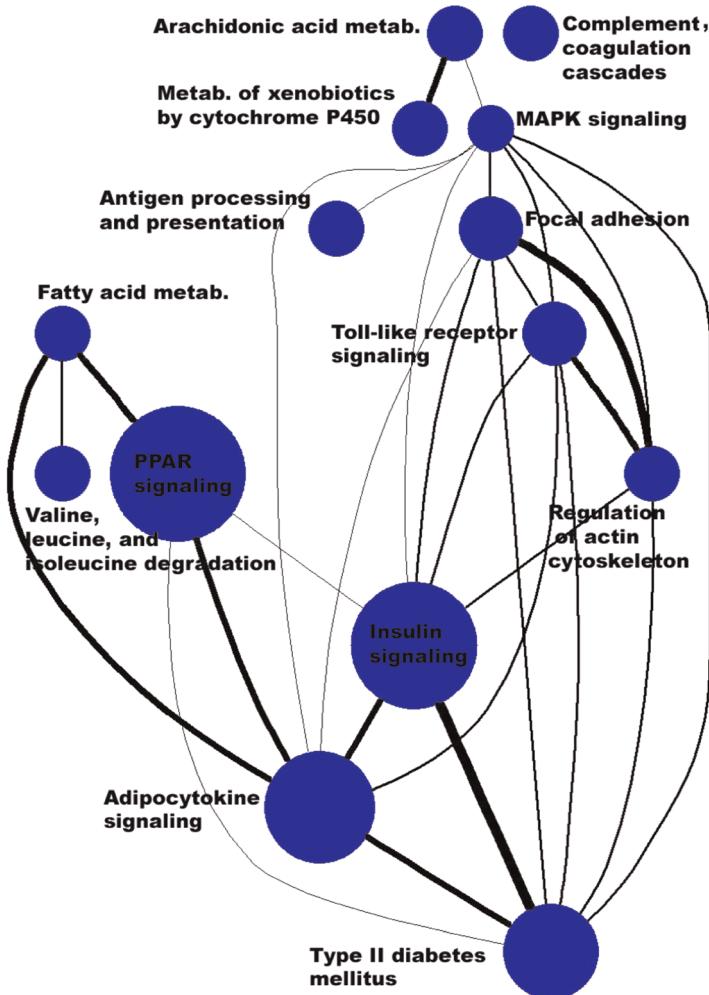


Figure 15.22 Pathway crosstalk with respect to the type-II diabetes mellitus candidate gene set. Identified by Rasche *et al.*, 2008. Pathways were derived from the KEGG database. Each pathway has been weighted according to the total disease score reflected by the size of the nodes. Only pathways with a total score >20 were selected for display. The thickness of the edges between the different pathway nodes reflects the overlap score derived from the sum of the scores of the overlapping genes. The graph was generated with the graphviz package (www.graphviz.org).

under- or overexpression, the test statistic will be small. The Z-score is defined as

$$z = \frac{|R - E(R)|}{\sqrt{\text{Var}(R)}}, \quad (15.202)$$

where E is the expectation and Var is the variance of R . These were calculated as

$$E(R) = \frac{n_i(n_i + 1)}{4} \quad (15.203)$$

and

$$\text{Var}(R) = \frac{n_i(n_i + 1)(2n_i + 1)}{24}, \quad (15.204)$$

respectively.

15.7.8 Classification Methods

An important medical application of microarray analysis is the diagnosis of diseases and subtypes of a disease, for example, cancer. Normal cells can evolve into malignant cancer cells by mutations of genes that control cell cycle, apoptosis, and other processes [55]. To determine the exact cancer type and stage is essential for the correct medical treatment of the patient. The task of sample diagnostics cannot be treated by the methods discussed so far. This task defines a complementary set of mathematical algorithms for gene expression – classification procedures. Recall that the purpose of clustering is to partition genes (and possibly conditions) into

coexpression groups by a suitable optimization method based on the expression matrix. The purpose of classification is to assign a given condition (for example, a patient's expression profile across a set of genes) to pre-existing classes of conditions (for example, groups of patient samples from known disease stages). The clustering methods discussed so far do not utilize any supporting tissue annotation (e.g., tumor vs. normal). This information is only used to assess the performance of the method. Such methods are often referred to as *unsupervised*. In contrast, *supervised* methods, attempt to predict the classification of new tissues based on their gene-expression profiles after training on examples that have been classified by an external "supervisor."

The practical problems underlying the classification of patients to disease subtypes are

- 1) new/unknown disease classes have to be identified;
- 2) marker genes have to be found that separate the disease classes; and
- 3) patients have to be classified by assigning them to one of the classes.

The general classification problem can be stated as follows: Let T be a set of n training samples consisting of pairs (\mathbf{x}_i, z_i) , $T = \{(\mathbf{x}_1, z_1), \dots, (\mathbf{x}_n, z_n)\}$, where \mathbf{x}_i is a p -dimensional vector and $z_i \in \{-1, 1\}$ is a binary label (class label). Each vector consists of the expression profile of the patient sample across the p marker genes and each label assigns this vector to one of the classes. Given a new query, $\mathbf{x} \in \Re_p$, the classification method (*classifier*) has to predict the group label, z , of \mathbf{x} given the training set.

Thus, each classification method can be interpreted as a function $F : \Re_p \times T \rightarrow \{-1, 1\}$.

15.7.8.1 Support Vector Machines

Support vector machines (SVMs) are the most widely used group of methods for classification [56]. Different studies have been published using SVMs in recent years, in particular for cancer diagnostics [57,58].

The intuition of support vector machines is that of a linear decision rule. Consider two different groups of vectors in \Re_p . We want to find a hyperplane that separates these two samples by making the least possible error. The usual problem is that there are many such separating hyperplanes so that we have to define some kind of optimization criterion. Figure 15.23 illustrates the problem with the simple case of a two-dimensional space spanned by the expression levels of the patients according to two marker genes (A and B). Hyperplanes 1 and 2 are both separating the two samples perfectly. However, given a new patient profile (red square) both methods would lead to different classification results. The problem here is that both hyperplanes are geometrically too close to one of the samples and thus risk misassignment of a future data.

The idea behind SVMs is to select a hyperplane that is more likely to generalize on future data. This is achieved by finding a hyperplane that maximizes the minimum distances of the closest points and thus to maximize the width of the margin between the two classes. The hyperplane is specified by the boundary training vectors (*support vectors*).

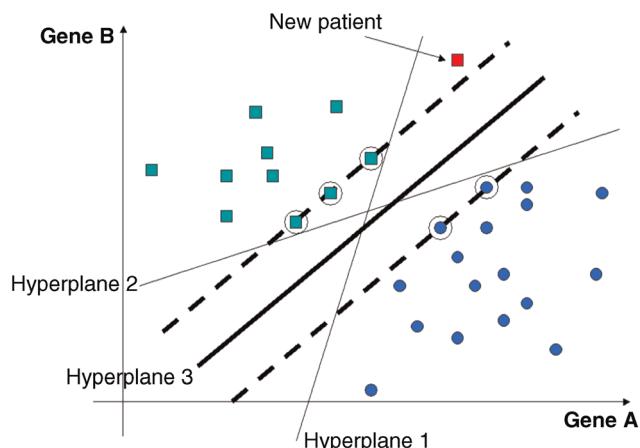


Figure 15.23 Support vector machines. Two classes of patient data are separated in the plane that is spanned by the expression levels according to two genes. Hyperplanes 1 and 2 yield two perfect linear separations of the groups; however, when classifying a new patient they disagree in the classification. Hyperplane 1 assigns the patient to group 1 (circles), whereas hyperplane 2 assigns the patient to group 2 (squares) due to the fact that both hyperplanes are geometrically too close to each of the subsets. The support vector machine classifier tries to maximize the margin between the two groups by defining support vectors (circled data points) and hyperplane that maximizes the minimum distances to these support vectors (hyperplane 3).

Recall the classification problem from the previous section. A hyperplane can be described by $H(\mathbf{w}, b) = \{\mathbf{x}; \mathbf{w}\mathbf{x} + b = 0\}$, with a vector $\mathbf{w} \in \mathfrak{R}_p$ that determines the orientation of the hyperplane and a scalar b that determines the offset of the hyperplane from the origin. Here, $\mathbf{w}\mathbf{x}$ denotes the inner- or dot-product of the two vectors. A hyperplane in two dimensions is given by a straight line (Figure 15.23) and in three dimensions by a plane. We say that a hyperplane supports a class if all points in that class fall on one side. Thus, we would like to find a pair \mathbf{w}, b so that $\mathbf{w}\mathbf{x}_i + b \geq 1$ for the points with class label $z_i = 1$ and $\mathbf{w}\mathbf{x}_i + b \leq -1$ for the points with class label $z_i = -1$. To compute the hyperplane with the largest margin, we search two supporting hyperplanes for the two classes. The support planes are pushed apart until they fall into a specific number of data vectors from each class (the support vectors marked with a circle in Figure 15.23). Thus, the solution depends only on these support vectors. The distance between the supporting hyperplanes $\mathbf{w}\mathbf{x} + b = 1$ and $\mathbf{w}\mathbf{x} + b = -1$ is equal to $\frac{2}{\|\mathbf{w}\|}$, where $\|\cdot\|$ denotes the Euclidean norm. Thus, maximizing the margin is equivalent of the following problem:

$$\text{Minimize } \|\mathbf{w}\|^2 \text{ and } b$$

$$\text{subject to } z_i(\mathbf{w}\mathbf{x}_i + b) \geq 1 \text{ for } i = 1, \dots, n \quad (15.205)$$

This problem can be represented by the Langrangian dual problem

$$\begin{aligned} \text{Minimize } \alpha \text{ values } & \frac{1}{2} \sum_{i,j} z_i z_j \alpha_i \alpha_j \mathbf{x}_i \mathbf{x}_j - \sum_i \alpha_i \\ \text{subject to } & \sum_i z_i \alpha_i = 0 \text{ and } \alpha_i \geq 0. \end{aligned} \quad (15.206)$$

Both problems lead to the same solution, that is, a hyperplane with the property that $\mathbf{w} = \sum_i z_i \alpha_i \mathbf{x}_i$. The classification rule found by the algorithm then reads for any new data

$$F_T(\mathbf{x}) = \text{sign} \left(\sum_i z_i \alpha_i \mathbf{x}_i \mathbf{x} + b \right). \quad (15.207)$$

It should be noted that this classification rule depends only on the support vectors, since the dual problem assigns values $\alpha_i = 0$ to all other data vectors.

In practice, it might be the fact that in the original dimension, p , no linear separation can be performed on the training data. SVMs then map the data to a higher dimensional space where a linear separation is possible, using a map $\Phi : \mathfrak{R}_p \rightarrow \mathfrak{R}_m, m > p$. Since the optimization problem involves only inner products of the vectors an optimal separating hyperplane in the projected space can be found by solving the problem for the inner products $\Phi(\mathbf{x}_i)\Phi(\mathbf{x}_j)$. Fortunately, (due to Mercer's theorem that is beyond the scope of this book) it is known that for

certain mappings and any two vectors the inner product in the projected dimension can be calculated using a *kernel function* $K : \mathfrak{R}_p \times \mathfrak{R}_p \rightarrow \mathfrak{R}$ such that

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)\Phi(\mathbf{x}_j). \quad (15.208)$$

Two kernels are widely used

- linear kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i \mathbf{x}_j$, and
- polynomial kernel: $K(\mathbf{x}_i, \mathbf{x}_j) = (\gamma \mathbf{x}_i \mathbf{x}_j + \epsilon)^d$.

A further modification due to the fact that in practice misclassifications will occur is the introduction of an error parameter, C , in the optimization problem (15.208) and the dual problem (15.208). This error parameter represents the trade-off between the training set misclassification error and the size of the margin. The optimization problem involving the kernel and this parameter reads

$$\begin{aligned} \text{Minimize } \alpha \text{ values } & \frac{1}{2} \sum_{i,j} z_i z_j \alpha_i \alpha_j K(\mathbf{x}_i \mathbf{x}_j) - \sum_i \alpha_i \\ \text{subject to } & \sum_i z_i \alpha_i = 0 \text{ and } C \geq \alpha_i \geq 0, \end{aligned} \quad (15.209)$$

and the classification is based on

$$F_T(\mathbf{x}) = \text{sign} \left(\sum_i z_i \alpha_i K(\mathbf{x}_i \mathbf{x}) + b \right). \quad (15.210)$$

Further reading on SVMs can be found in Ref. [59]. SVMs seem to be the method of choice for classifying samples according to gene-expression profiles. They have proven to outperform other procedures in particular in cancer diagnosis by several independent studies [57,58].

15.7.8.2 Other Approaches

A very simple classification approach is the *k-nearest neighbor* classification method [60]. Consider a training data set of n pairs (\mathbf{x}_i, z_i) , $i = 1, \dots, n$, of p -dimensional expression profiles and group labels. If a query, \mathbf{x} , is to be classified, then it is likely that the group label of \mathbf{x} equals the group label of the most similar training data. Thus, the classification rule is given by

$$F_T(\mathbf{x}) = z_{i_0}, \text{ where}$$

$$i_0 \in \arg \max \{S(\mathbf{x}, \mathbf{x}_i); \mathbf{x}_i \in T\}. \quad (15.211)$$

S is a suitable similarity function between the expression profiles, for example, the Pearson correlation coefficient. Taking into account the high error rates in microarray experiments, it is not reasonable to base the classification on just the nearest neighbor of the query in the training set but rather on the *k*-nearest neighbors. Thus, the result of the classification of the query is defined as the majority vote of these k data vectors. Further refinements are

performed by weighting the group labels of the data vectors according to their similarity to the query. K -nearest neighbor methods yield surprisingly good results in many classification procedures and can be used as a borderline test for more sophisticated algorithms.

Classification can also be combined with clustering methods using *clustering-based classification* [61]. If we consider the n training samples (tumor subtypes, cell lines, etc.) as expression vectors whose coordinates are the expression levels of some genes, that is, essentially transposing the expression matrix discussed in Section 15.7.5, then we can perform a clustering of the training

sample in two or more clusters. This yields groups of samples that are similar to each other based on the selected set of genes. Clustering-based classification method simply clusters the query sample together with the training samples and assigns the label of the highest confidence calculated from all training labels in the same cluster to the query.

A third group of algorithms is based on *boosting* [62]. The idea of boosting is to construct a good classifier by repeated calls of weak learning procedures. An example for a Boosting algorithm is the AdaBoost algorithm by Freund and Shapire.

Exercises

1) Robust statistical testing:

Let x_1, \dots, x_n and y_1, \dots, y_m be two independent samples of observations. Wilcoxon's rank test is a robust alternative to Gaussian-based tests. The test statistic T is based on the ranks of the first sample across the combined samples, that is,

$$T(x_1, \dots, x_n, y_1, \dots, y_m) = \sum_{i=1}^n R(x_i),$$

where $R(x_i)$ is the rank of x_i in the combined sample of observations.

Write a computer program that computes the exact P -values for this test.

Hint: Implement a recursive algorithm. Let $w(z, n, m)$ be the number of possible rank orderings that result in a value of T equal to z . This number is a sum of the number of possible rank orderings of T containing the highest rank, $m+n$, and those that do not, what can be described as

$$\begin{aligned} w(z, n, m) &= w(z - (m+n), n-1, m) \\ &\quad + w(z, n, m-1). \end{aligned}$$

The P -value can be derived by counting all combinations of rank orderings that yield a more extreme value of T divided by the total number of possible rank orderings.

2) Comparison of tests:

Expression of a specific gene was measured in two different patient groups yielding the following series of observations:

Group 1:	2434	2289	5599	2518	1123	1768	2304	2509	14 820	2489	1349	1494
Group 2:	3107	3365	4704	3667	2414	4268	3600	3084	3997	3673	2281	3166

This can be deduced from the *Cauchy-Schwartz inequality*:

$$\sum_{i=1}^p |a_i b_i| \leq \left(\sum_{i=1}^p a_i^2 \right)^{\frac{1}{2}} \left(\sum_{i=1}^p b_i^2 \right)^{\frac{1}{2}}, \quad (15.214)$$

for any two series of real values.

4) Clustering (practical exercise):

The Gene Expression Omnibus has a large collection of publicly available expression data. Search for expression data generated on the NCI-60 panel of cancer cell lines [63]. Download the preprocessed and normalized data and perform hierarchical clustering. Use different metrics and observe changes in cluster composition. Identify groups of genes that are specific for certain cancer types. Identify the most variable and the most constant expression patterns in the data set.

5) Urn models:

In Section 15.7.7, we have introduced the hypergeometric distribution in the context of overrepresentation analyses. This, and many other practical problems, can be described with so-called *urn models*. Consider an urn containing N balls out of which K are red and $N-K$ are black. The experiment consists of n draws from that urn. If the ball is replaced in the urn after each draw, we call the experiment *drawing with replacement* otherwise *drawing without replacement*. Of practical interest is the calculation of the probability of having exactly k red balls among the n balls drawn. This is given by $P(k) =$

$$\binom{n}{k} p^k (1-p)^{n-k} \quad (\text{Binomial distribution}), \quad p = \frac{K}{N},$$

we draw with replacement, that is, each ball is placed back in the urn after drawing, and $P(k) =$

$$\binom{K}{k} \binom{N-K}{n-k} / \binom{N}{n}$$

(Hypergeometric distribution), if we draw without replacement.

- a) Compute the expectation and variances of both distributions. What differences do you observe in the drawing with and without replacement?
- b) Limit theorem for the hypergeometric distribution:

Let $q_k = \binom{K}{k} \binom{N-K}{n-k} / \binom{N}{n}$ and show that, for large N , the hypergeometric distribution can be approximated by the binomial distribution by proving

$$\begin{aligned} \binom{n}{k} \left(p - \frac{k}{N} \right)^k \left((1-p) - \frac{n-k}{N} \right)^{n-k} \\ \leq q_k \leq \binom{n}{k} p^k (1-p)^{n-k} \left(1 - \frac{n}{N} \right)^{-n}. \end{aligned}$$

What can you deduce from this calculation?

Section 15.6

- 6) *Entropy and number of microstates*. (a) Show that Eq. (15.154) is a special case of Eq. (15.155). (b) Consider a molecule with n conformations of equal energy E and derive the entropy of its Boltzmann distribution.
- 7) *Heat from the environment as an energy source*. Imagine a hypothetical organism that uses heat from the environment as an energy supply. Invent a biological mechanism that respects the second law of thermodynamics.
- 8) *Equilibrium and steady state*. Discuss the differences between (i) mechanical equilibrium, (ii) thermodynamic equilibrium, and (iii) steady state. Find examples for each of them.
- 9) *Boltzmann distribution*. (a) The Boltzmann distribution for an ensemble of noninteracting particles with states x , at temperature T , reads

$$p(x) = \frac{1}{Z} e^{-E(x)/(k_B T)}.$$

- Explain the meaning of $E(x)$ and Z . (b) A particle in a quadratic potential well has the potential energy $E(x) = (a/2)x^2$, where x is the (scalar) particle position. Determine the Boltzmann probability for finding the particle at position x . How does the distribution change with temperature?
- 10) *Loop flux*. Consider chemical reactions $A \rightleftharpoons B, B \rightleftharpoons C$, and $C \rightleftharpoons A$ forming a loop without cofactors. Based on thermodynamic laws, show that the stationary flux in this loop must vanish.

- 11) *Principle of maximal entropy*. For a system with continuous states $x \in \mathbb{R}$ and energies $E(x)$, show that the Boltzmann distribution has the highest differential entropy among all distributions with fixed mean energy.
- 12) *Metabolic systems and electric circuits*. Explain similarities and differences between Kirchhoff's laws for electric currents and the laws for metabolic fluxes. Is there a metabolic analog to electric resistance?

References

- 1 Bronstein, I.N., Semendjajev, K.A. (1987) Taschenbuch der mathematik für ingenieure und Studenten Technischer Hochschulen, B.G. Teubner, Leipzig.
- 2 Kahlem, P., Sultan, M., Herwig, R., Steinfath, M., Balzereit, D., Eppens, B. *et al.* (2004) Transcript level alterations reflect gene dosage effects across multiple tissues in a mouse model of down syndrome. *Genome Res.*, 14 (7), 1258–1267.
- 3 Jolliffe, I.T. (1986) *Principal Component Analysis*, Springer, New York.
- 4 Dürr, D. and Teufel, S. (2009) *Bohmian Mechanics: The Physics and Mathematics of Quantum Theory*, Springer.
- 5 Honerkamp, J. (1994) *Stochastic Dynamical Systems*, John Wiley and Sons.
- 6 Trigg, G.L. (2005) *Mathematical Tools for Physicists*, Wiley-VCH Verlag.
- 7 Jahnke, T. and Huisenga, W. (2007) Solving the chemical master equation for monomolecular reaction systems analytically. *J. Math. Biol.*, 54, 1–26.
- 8 Moore, B.C. (1981) Principal component analysis in linear systems: controllability, observability, and model reduction. *IEEE Trans. AC*, AC-26, 17–32.
- 9 Liebermeister, W., Baur, U., and Klipp, E. (2005) Biochemical network models simplified by balanced truncation. *FEBS J.*, 272 (16), 4034–4043.
- 10 Shannon, C.E. (1948) A mathematical theory of communication. *AT&T Tech. J.*, 27, 379–423.
- 11 Thomas, J.A. and Cover, T.M. (1991) *Elements of Information Theory*, John Wiley & Sons, Inc.
- 12 Kullback, S. (1959) *Information Theory and Statistics*, John Wiley and Sons, New York.
- 13 Jaynes, E.T. (1957) Information theory and statistical mechanics. *Phys. Rev.*, 106, 620–630.
- 14 Wegscheider, R. (1902) Über simultane gleichgewichte und die beziehungen zwischen thermodynamik und reactionskinetik homogener systeme. *Z. Phys. Chem.*, 39, 257–303.
- 15 Schuster, S. and Schuster, R. (1989) A generalization of Wegscheider's condition. Implications for properties of steady states and for quasi-steady-state approximation. *J. Math. Chem.*, 3, 25–42.
- 16 Haldane, J.B.S. (1930) *Enzymes*, Longmans, Green and Co., London, (republished in 1965 by MIT Press, Cambridge, MA).
- 17 Alberti, R.A. and Cornish-Bowden, A. (1993) The pH dependence of the apparent equilibrium constant, K° , of a biochemical reaction. *Trends Biochem. Sci.*, 18 (8), 288–291.
- 18 Alberti, R.A. (1998) Calculation of standard transformed Gibbs energies and standard transformed enthalpies of biochemical reactants. *Arch. Biochem. Biophys.*, 353 (1), 116–130.
- 19 Jol, S.J., Kümmel, A., Hatzimanikatis, V., Beard, D.A., and Heinemann, M. (2010) Thermodynamic calculations for biochemical transport and reaction processes in metabolic networks. *Biophys. J.*, 99, 3139–3144.
- 20 Alberti, R.A. (2003) *Thermodynamics of Biochemical Reactions*, Wiley.
- 21 Mavrovouniotis, M. (1990) Group contributions for estimating standard Gibbs energies of formation of biochemical compounds in aqueous solution. *Biotechnol. Bioeng.*, 36, 1070–1082.
- 22 Noor, E., Haraldsdottir, H.S., Milo, R., and Fleming, R.M.T. (2013) Consistent estimation of Gibbs energy using component contributions. *PLOS Comp. Biol.*, 9, e1003098.
- 23 Flamholz, A., Noor, E., Bar-Even, A., and Milo, R. (2012) equilibrator – the biochemical thermodynamics calculator. *Nucleic Acids Res.*, 40 (D1), D770–D775.
- 24 von Berthalanffy, L. (1932) *Theoretische Biologie I. Band: Allgemeine Theorie, Physikochemie, Aufbau und Entwicklung des Organismus*, Gebrüder Borntraeger, Berlin.
- 25 von Berthalanffy, L. (1953) *Biophysik des Fließgleichgewichts. Einführung in die Physik offener Systeme und ihre Anwendung in der Biologie*, Vieweg & Sohn, Braunschweig.
- 26 Beard, D.A. and Qian, H. (2007) Relationship between thermodynamic driving force and one-way fluxes in reversible processes. *PLoS One*, 2 (1), e144.
- 27 Fleming, R.M.T., Maes, C.M., Saunders, M.A., Ye, Y., and Palsson, B.O. (2012) A variational principle for computing nonequilibrium fluxes and potentials in genome-scale biochemical networks. *J. Theor. Biol.*, 292, 71–77.
- 28 Ederer, M. and Gilles, E.D. (2007) Thermodynamically feasible kinetic models of reaction networks. *Biophys. J.*, 92, 1846–1857.
- 29 Liebermeister, W., Uhlendorf, J., and Klipp, E. (2010) Modular rate laws for enzymatic reactions: thermodynamics, elasticities, and implementation. *Bioinformatics*, 26 (12), 1528–1534.
- 30 Noor, E., Flamholz, A., Liebermeister, W., Bar-Even, A., and Milo, R. (2013) A note on the kinetics of enzyme action: a decomposition that highlights thermodynamic effects. *FEBS Lett.*, 587 (17), 2772–2777.
- 31 Kacser, H. and Burns, J.A. (1973) The control of flux. *Symp. Soc. Exp. Biol.*, 27, 65–104.
- 32 Noor, E., Bar-Even, A., Flamholz, A., Reznik, E., Liebermeister, W., and Milo, R. (2014) Pathway thermodynamics uncovers kinetic obstacles in central metabolism. *PLoS Comp. Biol.*, 10, e100348.
- 33 Kahlem, P., Sultan, M., Herwig, R., Steinfath, M., Balzereit, D., Eppens, B. *et al.* (2004) Transcript level alterations reflect gene dosage effects across multiple tissues in a mouse model of down syndrome. *Genome Res.*, 14 (7), 1258–1267.
- 34 Herwig, R., Aanstad, P., Clark, M., and Lehrach, H. (2001) Statistical evaluation of differential expression on cDNA nylon arrays with replicated experiments. *Nucleic Acids Res.*, 29 (23), E117.
- 35 Zien, A., Fluck, J., Zimmer, R., and Lengauer, T. (2003). Microarrays: how many do you need? *J. Comput. Biol.*, 10, 653–667.
- 36 Adjaye, J., Huntriss, J., Herwig, R., BenKahla, A., Brink, T.C., Wierling, C. *et al.* (2005) Primary differentiation in the human blastocyst: comparative molecular portraits of inner cell mass and trophectoderm cells. *Stem Cells*, 23 (10), 1514–1525.
- 37 Man, M.Z., Wang, X., and Wang, Y. (2000). POWER_SAGE: comparing statistical tests for SAGE experiments. *Bioinformatics*, 16 (11), 953–959.
- 38 Audic, S. and Claverie, J.M. (1997) Significance of digital gene expression profiles. *Genome Res.*, 7, 986–995.
- 39 Westfall, P.H. and Young, S.S. (1993) *Resampling-Based Multiple Testing: Examples and Methods for p-Value Adjustment*, John Wiley & Sons, New York.
- 40 Benjamini, Y. and Hochberg, Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Royal Statist. Soc. B*, 57, 289–300.
- 41 Tusher, V.G. *et al.* (2001) Significance analysis of microarrays applied to the ionizing radiation response. *PNAS*, 98, 5116–5121.
- 42 Storey, J.D. and Tibshirani, R. (2003) Statistical significance for genome-wide studies. *PNAS*, 100, 9440–9445.
- 43 Irizarry, R.A., Bolstad, B.M., Collins, F., Cope, L.M., Hobbs, B., and Speed, T.P. (2003) Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res.*, 31, e15.
- 44 Jain, A.K. and Dubes, R.C. (1988) *Algorithms for Clustering Data*, Prentice Hall, Englewood Cliffs, NJ.

- 45 Ross, D.T., Scherf, U., Eisen, M.B., Perou, C.M., Rees, C., Spellman, P. *et al.* (2000) Systematic variation in gene expression patterns in human cancer cell lines. *Nat. Genet.*, 24, 227–235.
- 46 Eisen, M.B., Spellman, P.T., Brown, P.O., and Botstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA*, 95, 14863–14868.
- 47 Kohonen, T. (1997) *Self-Organizing Maps*, Springer, Berlin.
- 48 Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dimitrovsky, E. *et al.* (1999) Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA*, 96, 2907–2912.
- 49 Törönen, P., Kolehmainen, M., Wong, G., and Castren, E. (1999) Analysis of gene expression data using self-organizing maps. *FEBS Lett.*, 451, 142–146.
- 50 Herwig, R., Poustka, A.J., Muller, C., Bull, C., Lehrach, H., and O'Brien, J. (1999) Large-scale clustering of cDNA-fingerprinting data. *Genome Res.*, 9 (11), 1093–1105.
- 51 MacQueen, J.B. (1967) Some methods for classification and analysis of multivariate observations, in *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability. 1* (eds LM. LeCam and J. Neymann), UCLA Press, Los Angeles.
- 52 Rousseeuw, P.J. (1984) Least median of squares regression. *J. Am. Stat. Assoc.*, 79, 871–880.
- 53 Consortium, G.O. (2003) The gene ontology (GO) database and information resource. *Nucleic Acids Res.*, 32, D258–D261.
- 54 Rasche, A., Al-Hasani, H., and Herwig, R. (2008) Meta-analysis approach identifies candidate genes and associated molecular networks for type-2 diabetes mellitus. *BMC Genomics*, 9, 310.
- 55 Hanahan, D. and Weinberg, R.A. (2000) The hallmarks of cancer. *Cell*, 100, 57–70.
- 56 Vapnik, V.N. (1995) *The Nature of Statistical Learning Theory*, Springer, New York.
- 57 Furey, T.S., Cristianini, N., Duffy, N., Bednarski, D.W., Schummer, M., and Haussler, D. (2000) Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16, 906–914.
- 58 Statnikov, A., Aliferis, C.F., Tsamardinos, I., Hardin, D., and Levy, S. (2004) A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, Epub ahead of print.
- 59 Cristianini, N. and Shawe-Taylor, J. (2000) *An introduction to Support Vector Machines*, Cambridge University Press, Cambridge.
- 60 Duda, R.O. and Hart, P.E. (1973) *Pattern Classification and Scene Analysis*.
- 61 Alon, U., Barkai, N., Notterman, D.A., Gish, G., Ybarra, S., Mack, D. *et al.* (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, 96, 6745–6750.
- 62 Freund, J. and Shapire, R. (eds) (1996) Experiments with a new boosting algorithm. *Machine Learning: Proceedings to the 13th International Conference*; Morgan Kaufmann, San Francisco.
- 63 Ross, D.T., Scherf, U., Eisen, M.B., Perou, C.M., Rees, C., Spellman, P. *et al.* (2000) Systematic variation in gene expression patterns in human cancer cell lines. *Nat. Genet.*, 24 (3), 227–235.

Summary

With the rapidly increasing generation and availability of biological data, it has become more and more important to organize and structure the data in such a way that information can easily be retrieved and compared with each other. As a result, the number of databases has grown rapidly over the past few years. Most of these databases are accessible through the World Wide Web and can be accessed from everywhere in the world, which is an enormously important service for the scientific community. In the following, various databases are introduced that cover different aspects of systems biology, such as pathway and model databases, databases of biology-related numbers and kinetic parameters, or databases integrating diverse omics data, such as sequencing and expression data.

16.1 General-Purpose Data Resources

Each year in January the journal *Nucleic Acids Research* offers a database issue dedicated to factual biological databases. In addition to this, each year in July the journal also publishes a web server issue presenting web-based services. These issues of the journal present most recent developments and updates on many relevant databases and web tools. Another journal of interest is the journal “Database: The Journal of Biological Databases and Curation” that is published by Oxford Journals.

Within this chapter we introduce several databases and resources that are frequently used in systems biology for data mining beginning with two data resources of more general use, the websites PathGuide and BioNumbers.

16.1.1 PathGuide

Pathway-related data and information are of major interest for systems biology. PathGuide is a pathway resource list

16.1 General-Purpose Data Resources

- PathGuide
- BioNumbers

16.2 Nucleotide Sequence Databases

- Data Repositories of the National Center for Biotechnology Information
- GenBank/RefSeq/UniGene
- Entrez
- EMBL Nucleotide Sequence Database
- European Nucleotide Archive
- Ensembl

16.3 Protein Databases

- UniProt/Swiss-Prot/TrEMBL
- Protein Data Bank
- PANTHER
- InterPro
- iHOP

16.4 Ontology Databases

- Gene Ontology

16.5 Pathway Databases

- KEGG
- Reactome
- ConsensusPathDB
- WikiPathways

16.6 Enzyme Reaction Kinetics Databases

- BRENDA
- SABIO-RK

16.7 Model Collections

- BioModels
- JWS Online

16.8 Compound and Drug Databases

- ChEBI
- Guide to PHARMACOLOGY

16.9 Transcription Factor Databases

- JASPAR
- TRED
- Transcription Factor Encyclopedia

16.10 Microarray and Sequencing Databases

- Gene Expression Omnibus
- ArrayExpress

References

giving an overview of web-accessible biological pathway and network databases [1]. PathGuide currently lists 547 resources providing information about biological pathways and molecular interactions. These include databases on protein interactions, metabolic and signaling pathways, transcription factors and gene regulatory networks, protein–compound interactions, and pathway diagrams. PathGuide also gives details on the availability and supported exchange standards of the respective data repositories.

URL: www.ncbi.nlm.nih.gov/geo/

Latest version: August 2013

Availability: Free for all Free for academic Commercial

16.1.2

BioNumbers

For many biological properties, numerical values are sometimes difficult to find in the literature. Most quantitative properties in biology depend on the context or the method of measurement, the organism, or the cell type. Often, however, the order of magnitude is already a very helpful information for modeling. BioNumbers is a database of useful biological numbers [2]. It allows you to easily browse or search for many common biological numbers that are sometimes difficult to find but can be very important for numerical modeling, such as the rate of translation of the ribosome or the number of bacteria in the gut. BioNumbers is a community effort to make quantitative properties of biological systems easily available together with full references.

URL: www.bionumbers.hms.harvard.edu/

Latest version: Unknown

Availability: Free for all Free for academic Commercial

16.2

Nucleotide Sequence Databases

Major repositories of nucleotide sequence data and diverse related data and information are provided by the National Center for Biotechnology Information (NCBI) and the European Bioinformatics Institute (EMBL-EBI). These data providers and their major sequence databases are introduced in the following.

16.2.1

Data Repositories of the National Center for Biotechnology Information

The National Center for Biotechnology Information provides several databases widely used in biological research.

Most important are the molecular databases, offering information about nucleotide sequences, proteins, genes, molecular structures, and gene expression. Besides these, several databases comprising scientific literature are available as well. The NCBI also provides a taxonomy database that contains names and lineages of more than 455 000 organisms representing about 10% of the described species of life on earth. For more than 1000 organisms, whole genomes and corresponding gene maps are available, along with the tools for their inspection. A full overview of the databases provided by the NCBI can be found at <http://www.ncbi.nlm.nih.gov/Sitemap/index.html>. All these databases are searchable via the Entrez search engine accessible through the NCBI homepage.

URL: www.ncbi.nlm.nih.gov/

Availability: Free for all Free for academic Commercial

16.2.2

GenBank/RefSeq/UniGene

Among the nucleotide sequence databases, the Genetic Sequence database (GenBank), the Reference Sequence database (RefSeq), and UniGene can be found at NCBI. GenBank (Release 207.0, from April 2015) comprises 189 billion nucleotide bases from more than 182 million reported sequences. The RefSeq database is a curated, nonredundant set of sequences including genomic DNA, mRNA, and protein products for major research organisms [3]. In UniGene, expressed sequence tags (ESTs) and full-length mRNA sequences are organized into clusters, each representing a unique known or putative gene of a specific organism. For molecular biological analyses, for example, sequencing or expression profiling, the mRNA of expressed genes is usually translated into a complementary DNA (cDNA), since this is more stable and feasible for standard biotechnological methods. An EST is a short – approximately 200–600 bp long – sequence from either side of a cDNA clone that is useful for identifying the full-length gene, for example, for locating the gene in the genome. In addition to nucleotide sequences, protein sequences can be searched for at the NCBI site via Entrez Proteins. Searches are performed across several databases, including RefSeq, Swiss-Prot, and Protein Data Bank (PDB) (see Section 16.3).

URL: www.ncbi.nlm.nih.gov/genbank

URL: www.ncbi.nlm.nih.gov/refseq

URL: www.ncbi.nlm.nih.gov/unigene

Latest version: Updated every 2 months

Availability: Free for all Free for academic Commercial

16.2.3

Entrez

Entrez is the global query cross-database search system of the NCBI. Entrez offers diverse information about specific genetic loci (the location of a specific gene). Thus, Entrez provides a central hub for accessing gene-specific information of a number of species, such as human, mouse, rat, zebrafish, nematode, fruit fly, cow, and sea urchin.

One major part of Entrez are the literature databases PubMed and OMIM (Online Mendelian Inheritance in Man). PubMed is a database of citations and abstracts for the biomedical literature. Citations are from MEDLINE and additional life science journals. OMIM is a catalog of human genes and genetic disorders with textual information and copious links to the scientific literature.

Thus, the databases at the NCBI are one of the major resources for sequence data, annotations, and literature references. They can be used to determine what is known about a specific gene or its protein, or to get information about the sequences, its variants, or polymorphisms. In addition to this, the NCBI also offers a database on gene expression data (Gene Expression Omnibus (GEO); see Section 16.10).

URL: www.ncbi.nlm.nih.gov/gquery

Latest version: Unknown

Availability: Free for all Free for academic Commercial

16.2.4

EMBL Nucleotide Sequence Database

Similar to the NCBI, also the European Bioinformatics Institute offers several biologically relevant databases. This includes databases on nucleotide sequences, genes, and genomes (EMBL Nucleotide Database, Ensembl automatic genome annotation database), a database on alternative splicing sites (ASTD), a database of protein modifications (RESID), a database on protein families and protein domains (InterPro), a database on macromolecular structures (PDBe), and a database on gene expression data (ArrayExpress; see Section 16.10). The protein databases UniProt, Swiss-Prot, and TrEMBL (Translation from EMBL) will be discussed in Section 16.3.

URL: www.ebi.ac.uk/Databases

Availability: Free for all Free for academic Commercial

16.2.5

European Nucleotide Archive

The European Nucleotide Archive (ENA) incorporates, organizes, and distributes nucleotide sequences from public sources and synchronizes its data in a daily manner with the DNA Database of Japan (DDBJ) and GenBank, which are the two other nucleotide sequence databases most important worldwide [4].

URL: www.ebi.ac.uk/ena

Latest version: July 2015

Availability: Free for all Free for academic Commercial

16.2.6

Ensembl

The Ensembl project is developing and maintaining a system for the management and presentation of genomic sequences and its annotation for eukaryotic genomes [4,5]. Annotation, in this context, is the characterization of features of the genome using computational and experimental methods. In the first place, this is the prediction of genes, including structural elements such as introns and exons, from the assembled genome sequence and the characterization of genomic features, such as repeated sequence motifs, conserved regions, or single nucleotide polymorphisms (SNPs). SNPs (pronounced “snips”) are common DNA sequence variations among individuals, where a single nucleotide is altered. Furthermore, annotation includes information about functional domains of the proteins encoded by the genes and the roles that the gene products fulfill in the organism.

The central component of Ensembl is a relational database storing the genome sequence assemblies and annotations produced by Ensembl’s automated sequence annotation pipeline, which utilizes the genome assemblies and data from external resources for this purpose. Ensembl provides genomic annotations for several vertebrates (e.g., human, chimp, mouse, rat, pufferfish, zebrafish, and chicken), arthropods (e.g., mosquito, honeybee, and fruit fly), and others. Annotations, such as genes with their intron/exon structure and SNPs, among others, can be viewed along the assembled sequence contigs using the Ensembl ContigView, which is accessible via the organism-specific webpages.

URL: www.ensembl.org

Latest version: May 2015

Availability: Free for all Free for academic Commercial

16.3 Protein Databases

In addition to several nucleotide sequence databases, also a variety of protein sequence databases exist, ranging from simple sequence repositories to expertly curated universal databases that cover many species and provide a lot of further information.

16.3.1

UniProt/Swiss-Prot/TrEMBL

One of the leading protein databases is UniProt, a collection of protein sequences and their annotations [6]. The UniProt Knowledgebase (UniProtKB) has two sections: a section of manually annotated and reviewed sequences known as Swiss-Prot, and a section of automatically annotated and nonreviewed sequences known as TrEMBL. UniProt is maintained by the Swiss Institute of Bioinformatics (SIB), the Protein Information Resource (PIR), and the European Bioinformatics Institute. As of June 2015, UniProtKB/Swiss-Prot contains about 550 000 annotated protein sequence entries. It offers a high level of annotation comprising information about the protein origin (gene name and species), amino acid sequence, protein function and location, protein domains and sites, quaternary structure, references to the literature, protein-associated diseases, and many further details. In addition, Swiss-Prot provides cross-references to several external data collections such as nucleotide sequence databases (DDBJ/EMBL-ENA/GenBank), protein structure databases, databases providing protein domain and family characterizations, and disease-related databases, among others.

Since the creation of fully curated Swiss-Prot entries is a highly laborious task, the UniProtKB/TrEMBL database was introduced that provides an automated annotation process. TrEMBL contains computer-annotated entries generated by *in silico* translation of all coding sequences (CDS) available in the nucleotide databases (DDBJ/EMBL/GenBank). As of June 2015, TrEMBL contains more than 48 million sequence entries. The entries offered at TrEMBL do not overlap with those found in Swiss-Prot.

URL: www.uniprot.org/uniprot

Latest version: Each entry has its own version history

Availability: Free for all Free for academic Commercial

16.3.2

Protein Data Bank

Biological macromolecules, that is, proteins and nucleic acids, fold into specific three-dimensional structures.

Using techniques such as X-ray crystallography or nuclear magnetic resonance (NMR), these structures can be solved and the three-dimensional coordinates of the atoms can be determined. Obviously, such information is extremely valuable for understanding the biological activity of the molecules and their interaction with possible reaction partners. The PDB database is the main repository for three-dimensional structures of biological macromolecules [7]. As of June 2015, the database holds more than 100 000 structures.

The PDB website offers extensive search and browse capabilities. In the most simple case, one can enter a PDB ID, a four-character alphanumeric identifier, to get straight to a specific structure. 1B06, for instance, brings up the information for the superoxide dismutase of *Sulfolobus acidocaldarius*, a thermophilic archaebacterium. The resulting page gives essential information about the protein, such as literature references, quality parameters for the crystal structure, and Gene Ontology (GO) terms (see Section 16.4). Via the available pull-down menus, further functions can be reached, such as a 3D view of the protein of interest and links to download the protein structure (PDB or mmCIF (macromolecular Crystallographic Information File) format) or sequence (FASTA format) files.

URL: www.rcsb.org

Latest version: Weekly updates

Availability: Free for all Free for academic Commercial

16.3.3

PANTHER

PANTHER is a comprehensive curated database of protein families, trees, subfamilies, and functions [8]. The main goal of the PANTHER database is the inference of gene and protein function using phylogenetic trees applied to large sequence databases. This includes also the detailed representation of evolutionary events in gene family histories.

URL: www.pantherdb.org

Latest version: Version 10.0, May 2015

Availability: Free for all Free for academic Commercial

16.3.4

InterPro

InterPro is a protein signature database comprising information about protein families, domains, and functional groups [9]. It combines many commonly used protein

signature databases and is a very powerful tool for the automatic and manual annotation of new or predicted proteins from sequencing projects. In addition, InterPro entries are mapped to Gene Ontology (see Section 16.4). InterPro is also used to annotate protein sequences of UniProtKB, the UniProt Knowledgebase.

URL: www.ebi.ac.uk/interpro

Latest version: Version 52, May 2015, updated every 8 weeks

Availability: Free for all Free for academic Commercial

16.3.5 iHOP

iHOP is a web service that allows to explore a network of gene and protein interactions by direct navigation of a pool of published scientific literature. iHOP integrates information on more than 80 000 biological molecules automatically extracted from key sentences of millions of PubMed documents [10,11].

URL: www.ihop-net.org

Latest version: Unknown

Availability: Free for all Free for academic Commercial

16.4 Ontology Databases

16.4.1 Gene Ontology

The accumulation of scientific knowledge is a decentralized, parallel process. Consequently, the naming and description of new genes and gene products is not necessarily systematic. Often gene products with identical functions are given different names in different organisms or the verbal description of the location and function might be quite different (e.g., protein degradation versus proteolysis). This, of course, makes it very difficult to perform efficient searching across databases and organisms.

This problem has been recognized, and in 1998 the Gene Ontology project was initiated as a collaborative effort of the *Saccharomyces* Genome Database (SGD), the Mouse Genome Database (MGD), and FlyBase. The aim of the GO is to provide a consistent, species-independent, functional description of gene products [12,13]. Since 1998, the GO project has grown considerably and now includes databases for plant, animal, and prokaryotic genomes. Effectively, GO consists of a controlled vocabulary (the GO terms) used to describe the biological

function of a gene product in any organism. The GO terms have a defined parent–child relationship and form a directed acyclic graph (DAG). In a DAG, each node can have multiple child nodes, as well as multiple parent nodes. Cyclic references, however, are forbidden. The combination of vocabulary and relationship between nodes is referred to as ontology. At the root of the GO are the three top-level categories, *molecular function*, *biological process*, and *cellular component*, which contain many levels of child nodes (GO terms) that describe a gene product with increasing specificity. The GO Consortium, in collaboration with other databases, develops and maintains the three top-level ontologies (the set of GO terms and their relationship) themselves, creates associations between the ontologies and the gene products in the participating databases, and develops tools for the creation, maintenance, and use of the ontologies.

Let us look at a practical example to see how the concept works. The enzyme superoxide dismutase, for instance, is annotated in FlyBase (the *Drosophila melanogaster* database) with the GO term *cytoplasm* in the cellular component ontology, with the GO terms *defense response* and *determination of adult life span* in the biological process ontology, and with the terms *antioxidant activity* and *copper, zinc superoxide dismutase activity* in the molecular function ontology. The GO term *cytoplasm* itself has the single parent *intracellular*, which has the single parent *cell*, which is finally connected to *cellular component*. The other GO terms for superoxide dismutase are connected in a similar hierarchical way to the three top categories.

To use the GO effectively, many different tools have been developed that are listed on the GO website. The repertoire encompasses web-based and stand-alone GO browsers and editors and programs for many specialized tasks. One of these tools is AmiGO (<http://amigo.geneontology.org>), which is a web-based GO browser maintained by the GO Consortium. AmiGO can be used to browse the terms of the ontologies. GO terms are uniquely identified by a seven-digit number, the GO-ID.

URL: www.geneontology.org

Latest version: Unknown

Availability: Free for all Free for academic^{a)} Commercial

a) Gene Ontology Consortium data and data products are licensed under the Creative Commons Attribution 4.0 Unported License.

16.5 Pathway Databases

The development of models of biochemical reaction networks requires information about the stoichiometry and

topology of the reaction network. Such information can be found in databases such as KEGG (Kyoto Encyclopedia of Genes and Genomes) and Reactome. Often pathway databases cover a specific scope, for example, metabolic pathways, signal transduction pathways, or gene regulatory networks. Some databases act as metadatabases that integrate pathway data from multiple sources building up a comprehensive resource of pathway and interaction data such as ConsensusPathDB.

16.5.1

KEGG

KEGG is a reference knowledge base offering information about genes and proteins, biochemical compounds and reactions, and pathways. The data are organized in three parts: the gene universe (consisting of the GENES, SSDB, and KO databases), the chemical universe (with the COMPOUND, GLYCAN, REACTION, and ENZYME databases that are merged as LIGAND database), and the protein network consisting of the PATHWAY database [14]. Besides this, the KEGG pathway database is hierarchically classified into categories and subcategories. The topmost categories are metabolism, genetic information processing, environmental information processing, cellular processes, organismal systems, and human diseases. Subcategories of metabolism are, for example, carbohydrate, energy, lipid, nucleotide, or amino acid metabolism. These are subdivided into the different pathways, such as glycolysis, citrate cycle, and purine metabolism, among others. To reflect groups of orthologous genes that have identical functions, KEGG is using internal identifiers, given by a K number (e.g., K00001 for alcohol dehydrogenase) across the different species.

The gene universe offers information about genes and proteins generated by genome sequencing projects. Information about individual genes is stored in the GENES database, which is semiautomatically generated from the submissions to GenBank, the NCBI RefSeq database, the EMBL database, and other publicly available organism-specific databases. K numbers are further assigned to entries of the GENES database. The SSDB database contains information about amino acid sequence similarities between protein-coding genes computationally generated from the GENES database. This is carried out for many complete genomes and results in a huge graph depicting protein similarities with clusters of orthologous and paralogous genes.

The chemical universe offers information about chemical compounds and reactions relevant to cellular processes. It includes more than 17 000 compounds (internally represented by C numbers, e.g., C00001 denotes water), a

separate database for carbohydrates (nearly 11 000 entries; represented by a number preceded by G, e.g., G10481 for cellulose), more than 9900 reactions (with R numbers, e.g., R00275 for the reaction that converts the superoxide radical into hydrogen peroxide), and more than 6500 enzymes (denoted by EC numbers as well as K numbers for orthologous entries). All these data are merged as LIGAND database [15]. Thus, the chemical universe offers comprehensive information about metabolites with their respective chemical structures and biochemical reactions.

KEGG's protein network provides information about protein interactions comprising pathways and protein complexes. The 476 KEGG reference pathway diagrams (maps), offered on the website, give clear overviews of important pathways. Organism-specific pathway maps are automatically generated by coloring of organism-specific genes in the reference pathways.

URL: www.genome.jp/kegg

Latest version: Version 75, July 2015

Availability: Free for all Free for academic Commercial

16.5.2

Reactome

Reactome is an open, online database of fundamental human biological processes. The Reactome project is managed as a collaboration of the Cold Spring Harbor Laboratory, the European Bioinformatics Institute, and the Gene Ontology Consortium [16,17]. The database is divided into several modules of fundamental biological processes that are thought to operate in humans. Each module of the database has one or more primary authors and is further peer reviewed by experts of the specific field. Each module can also be referenced by its revision date and thus can be cited like a publication.

The current version of Reactome (version 53) describes 8128 human proteins participating in 8369 reactions. Besides the description of molecular species, reactions, and pathways, the database also annotates a broad range of major disease processes at the molecular level, such as mutations leading to the loss or gain of function of the gene product or infectious agents such as a virus that introduces a novel gene product with a new reaction perturbing normal human processes.

The Reactome website offers several tools to analyze data, such as a tool for pathway overrepresentation analysis or a function to compare human processes with the processes of another species showing the overlap of the interaction network. Data from Reactome can be exported in various formats such as SBML and BioPAX.

URL: reactome.org

Latest version: Version 53, July 2015

Availability: Free for all Free for academic^{a)} Commercial

a) Reactome data and software are licensed under the Creative Commons Attribution 4.0 International License.

16.5.3

ConsensusPathDB

ConsensusPathDB is a database integrating human functional interactions [18–20]. Currently, the database integrates the content of 32 different interaction databases with heterogeneous foci comprising a total of about 155 000 distinct physical entities and about 435 000 distinct functional interactions covering nearly 4400 pathways. The database comprises protein–protein interactions, biochemical reactions, and gene regulatory interactions. ConsensusPathDB has a sophisticated interface for the visualization of the functional interaction networks. Furthermore, the database provides functionalities for overrepresentation analysis.

URL: consensuspathdb.org

Latest version: Version 30, January 2015

Availability: Free for all Free for academic Commercial

16.5.4

WikiPathways

Compared with KEGG and Reactome, WikiPathways is a public wiki for pathway curation [21]. WikiPathways serves as a repository for biological knowledge by the use of pathway diagrams. Entities such as genes, proteins, or metabolites can be annotated with many different identifier systems, such as Ensembl or ChEBI, and the entries can be linked to other external databases or information resources, such as genome browsers, experimental platforms, Gene Ontology, or Wikipedia.

URL: www.wikipathways.org

Latest version: Pages are updated individually

Availability: Free for all Free for academic Commercial

16.6

Enzyme Reaction Kinetics Databases

High-throughput projects, such as the international genome sequencing efforts, accumulate large amounts of

data at an amazing rate. These data are essential for the reconstruction of phylogenetic trees and gene-finding projects. However, for kinetic modeling, which is at the heart of systems biology, kinetic data of proteins and enzymes are needed. Unfortunately, this type of data are notoriously difficult and time-consuming to obtain since proteins often need individually tuned purification and reaction conditions. Furthermore, the results of such studies are published in a large variety of journals from different fields. In this situation, the databases BRENDA and SABIO-RK aim to be comprehensive resources of kinetic data.

16.6.1

BRENDA

Basically, BRENDA is a curated database that contains a large number of functional data for individual enzymes [22]. These data are gathered from the literature and made available via a web interface. The information that is collected ranges from K_M values (currently more than 127 000), over molecular weight, to specific activity as well as pH and temperature optima. One of the BRENDA's strengths is the multitude of ways the database can be searched. It is, for instance, easy to find all enzymes that are above a specific molecular weight, belong to *Caenorhabditis elegans*, or have a temperature optimum above 30 °C. Using the *Advanced Search* feature, it is possible to construct arbitrarily complex search queries involving the information fields shown on the website.

Sometimes it is necessary to search for all enzymes that are glycosylases without knowing the corresponding EC number, or to find all enzymes that exist in horses without knowing the exact scientific name. In this situation, the *ECTree browser* and the *TaxTree search* are helpful by providing a browser-like interface to search down the hierarchy of EC number descriptions or taxonomic names. A similar browser is also available for Gene Ontology terms, which were discussed earlier in this chapter. *Pathway Maps* are also a convenient way to search for enzymes. This feature makes it possible to select a pathway from a large list, which is then displayed in a graphical way very similar to the KEGG database.

BRENDA is also very well connected to other databases that can provide further information about an enzyme in question. Associated GO terms are directly linked to a GO browser, substrates and products of the catalyzed reactions can be displayed as chemical structures, sequence data can be obtained from Swiss-Prot, and, if crystallographic data exist, a link to PDB is provided. Of course, also relevant links to the literature are provided, from where the data originated.

If desired, the list of results can be downloaded as a tab-separated text file, but it is also possible to query BRENDA programmatically via a SOAP interface. Explicit examples for this are given in the tutorials (www.brenda-enzymes.org/tutorial.php).

URL: www.brenda-enzymes.org

Latest version: June 2015

Availability: Free for all Free for academic Commercial

16.6.2

SABIO-RK

Like BRENDA, also SABIO-RK is a database for enzyme kinetic data [23]. Basically, search details such as organism, enzyme name, organ, cellular location, parameter type, or year of publication are entered in a web-based search form, which allows to combine the search terms using “AND”, “OR”, and “NOT”. A search for “*Homo sapiens*” AND “alcohol dehydrogenase”, for instance, yields 358 hits that can be further examined. All information about the result is shown in a tabular format containing data about substrates, products, experimental conditions, literature reference, and, of course, the kinetic constants of the reaction. The presentation is very neat and readable, but in contrast to BRENDA the units of the kinetic parameters are not standardized; that is, K_m values can appear in mM or μM and k_{cat} can be given in s^{-1} or min^{-1} . Selected entries can then be exported in different formats such as tab-separated text file, Excel sheet, SBML, or BioPAX. Finally, like BRENDA, also SABIO-RK can be accessed programmatically (via RESTful Web Services).

URL: sabio.villa-bosch.de

Latest version: September 2014

Availability: Free for all Free for academic Commercial

16.7

Model Collections

A lot of different mathematical models of biological systems have already been developed in the past and are described in the literature. However, these models are usually not available in a computer-amenable format. During the past few years, huge efforts have been made on the gathering and implementation of existing models in databases. Two well-known databases in this respect are BioModels and JWS, which are described in more detail in the following.

16.7.1

BioModels

The BioModels project is an international effort to (i) define agreed-upon standards for model curation, (ii) define agreed-upon vocabularies for annotating models with connections to biological data resources, and (iii) provide a free, centralized, publicly accessible database of annotated, computational models in SBML, and other structured formats [24–26]. The 29th release of the BioModels database provides access to more than 144 000 models, of which a large number is automatically generated from pathway resources. One thousand two hundred ninety-six models correspond to models published in the literature, of which 575 models have been manually curated. Models can be browsed in the web interface, online simulations can be performed via the external simulation engine of JWS Online (see below), or they can be exported in several prominent file formats (e.g., SBML, CellML, and BioPAX) for external usage by other programs.

URL: biomodels.org

Latest version: Version 29, April 2015

Availability: Free for all Free for academic Commercial

16.7.2

JWS Online

Another model repository that provides kinetic models of biochemical systems is JWS Online [27]. As of July 2015, this model repository provides more than 190 models. Models in JWS Online can be interactively run and interrogated over the Internet. Besides acting as a model repository, JWS Online also provides methods for model building allowing the construction and annotation of new models. Furthermore, it integrates an interface to simulate, study, and analyze the dynamic behavior of the models directly via the website.

URL: jjj.biochem.sun.ac.za

Latest version: Unknown

Availability: Free for all Free for academic Commercial

16.8

Compound and Drug Databases

Small molecules serve as building blocks for larger biological compounds. These small molecules are also

subject to diverse enzymatic reactions modifying molecules in multiple ways. The complex but coordinated conversion of molecules by a living organism defines its metabolism. Several repository providers are collecting and integrating data about any kind of molecule related to living organisms including also information about, for example, drugs and their targets that are of particular interest to systems medicine.

16.8.1

ChEBI

ChEBI is a database of chemical entities of biological interest [28]. ChEBI provides a standardized description of molecular entities using a well-defined ontology that is aligned with other ontologies, such as Gene Ontology (see Section 16.4.1). ChEBI provides information about more than 45 000 fully annotated entities. Each entry in the database is manually annotated by experts and checked against the primary literature and other public data resources. Other external databases, such as Reactome (see Section 16.5.2) and BioModels (see Section 16.7.1), use stable and unique primary ChEBI identifiers to refer unambiguously to the chemicals as they appear in their biological context.

URL: www.ebi.ac.uk/chebi

Latest version: Release 129, July 2015

Availability: Free for all Free for academic Commercial

16.8.2

Guide to PHARMACOLOGY

The Guide to PHARMACOLOGY is an open-access resource on pharmacological, chemical, genetic, functional, and pathophysiological data on the targets of approved and experimental drugs [29]. It provides peer-reviewed information on key properties of drugs and their targets. The data provided by Guide to PHARMACOLOGY are extensively linked to other public databases, such as ChEMBL, DrugBank, Ensembl, PubChem, UniProt, and PubMed. As of March 2015, it integrates information of more than 7500 ligands, 2700 targets, and 12 800 curated binding constants.

URL: www.guidetopharmacology.org

Latest version: Version 2015.1, March 2015

Availability: Free for all Free for academic^{a)} Commercial

a) Guide to PHARMACOLOGY data are licensed under the Creative Commons License.

16.9

Transcription Factor Databases

16.9.1

JASPAR

JASPAR is a database for transcription factor binding sites from multicellular eukaryotes, provided as frequency matrices [30]. The database can be accessed interactively via a web interface, programmatically via a BioPython package (see Chapter 17) or Web API, and it can also be downloaded as flat files. JASPAR consists of a collection of different databases, of which the most important is JASPAR CORE. This is a curated, nonredundant set of profiles from published articles that currently contains 593 entries. Using the web interface, the different subsections of JASPAR CORE (vertebrates, nematodes, insects, plants, and fungi) can be browsed interactively or searched according to ID, name, species, class, and type. Curiously, the “Quick Search” button on the main page accepts text for the species search (e.g., “*Homo sapiens*”), while the search inside the different CORE sections only accepts taxonomic species IDs (e.g., 9606 for “*Homo sapiens*”). If a single profile is selected, the frequency matrix is provided as text and sequence logo and links to other databases, such as PubMed or PDB, are also available. If several profiles are selected, they can be clustered hierarchically using the STAMP tool (<http://www.benoslab.pitt.edu/stamp>) or a user-supplied sequence can be compared against the selected profiles to predict possible binding sites inside the sequence.

URL: jaspardev.genereg.net

Latest version: Version 5, November 2013

Availability: Free for all Free for academic Commercial

16.9.2

TRED

TRED (Transcriptional Regulatory Element Database) contains genome-wide information about transcriptional regulatory elements for human, mouse, and rat [31,32]. Promoter sequences were extracted from databases such as GenBank, EPD, and DBTSS in an automated process resulting in 58 229 promoter sequences for humans, 50 764 promoter sequences for mice, and 30 386 promoter sequences for rats. After specifying the organism and chromosome number, TRED can be browsed according to genes, promoters, or binding sites. For a specific promoter, the sequence can be displayed together with information from where

it was obtained (e.g., GenBank entry) and a quality score indicating how reliable the information is. TRED also contains curated information about 36 cancer-related transcription factor families (BCL, BRCA, MYB, etc.) that can be used to search for target genes in the three available species. However, in contrast to JASPAR, TRED does not provide transcription factor-specific frequency or weight matrices.

URL: rulai.cshl.edu/TRED

Latest version: Unknown

Availability: Free for all Free for academic Commercial

16.9.3 Transcription Factor Encyclopedia

The Transcription Factor Encyclopedia is a web-based repository of mini-review articles on transcription factors in human, mouse, and rat [33]. The mission of the Transcription Factor Encyclopedia is to facilitate the curation, evaluation, and dissemination of transcription factor data and to provide a platform for the research community working on transcription factor-related aspects. As of July 2015, the web-based compendium provides more than 800 transcription factor articles. Each article starts with an overview of the described transcription factor, followed by detailed information on its structure, its binding sites and targets, known isoforms, information on interactions with other molecules related to its activity, further details on genetics and expression of the transcription factor, and links to related ontologies and publications. To assess the completeness of the description, an article completion score is computed for every transcription factor article.

URL: www.cisreg.ca/tfe

Latest version: Unknown

Availability: Free for all Free for academic Commercial

16.10 Microarray and Sequencing Databases

In molecular biology, high-throughput data analysis is getting more and more popular. This led to the development of data repositories making this kind of data available to the scientific community in a standardized way. The two most popular data repositories dealing with the integration of high-throughput data coming primarily from microarray studies and next-generation sequence analysis are Gene Expression Omnibus and ArrayExpress.

16.10.1

Gene Expression Omnibus

The Gene Expression Omnibus repository is maintained by the National Center for Biotechnology Information and acts as a global repository for high-throughput functional genomic data [34,35]. Data deposited in GEO represent original research provided by the scientific community. In particular, it is well known as a public archive for microarray and next-generation sequencing (NGS) data. The website supports archiving raw data, processed data, and metadata that are indexed and cross-linked. As of July 2015, the repository covers more than 3800 data sets providing data on more than 1.5 million samples. GEO offers functions to search, browse, and download data. Specific links and functions also help to discover further information about the data set of interest; for example, links refer to similar studies or information about pathways helping to characterize a gene or a list of genes, and cluster diagrams give a first overview of the results of a gene expression study. Besides acting as a data repository, the GEO website also provides tools to query, analyze, and visualize the data. For instance, the integrated clustering tool provides several clustering algorithms and allows us to zoom into the heat map diagram of the hierarchical clustering. A more advanced tool is the GEO2R web application that allows, for example, individually designed expression analysis of a data set providing top-ranked differentially expressed genes with, for example, *p*-values and log fold changes, and profile views for specific genes across the samples.

URL: www.ncbi.nlm.nih.gov/geo/

Latest version: 3848 data sets, July 2015

Availability: Free for all Free for academic Commercial

16.10.2

ArrayExpress

Another repository of array-based studies and sequencing data is ArrayExpress, which acts as an archive for functional genomic data and is maintained by the European Bioinformatics Institute [36]. It covers data from over 7000 sequencing and 42 000 array-based studies. In particular, the submission of sequencing data is significantly growing over the past few years. Similarly to GEO, ArrayExpress promotes data compliance to the Minimum Information About a Microarray Experiment (MIAME) or Minimum Information About Sequencing Experiment (MINSEQE) standard. In addition to the data uploaded directly to ArrayExpress, also data from GEO are imported to provide users with a single access point for

functional genomic data. All data from ArrayExpress are available for download in the standardized format MAGE-TAB facilitating the linking to analysis environment such as Bioconductor [37].

URL: www.ebi.ac.uk/arrayexpress

Latest version: Updated daily

Availability: Free for all Free for academic Commercial

References

- 1 Bader, G.D., Cary, M.P., and Sander, C. (2006) PathGuide: a pathway resource list. *Nucleic Acids Res.*, 34 (Database issue), D504–D506.
- 2 Milo, R., Jorgensen, P., Moran, U. *et al.* (2010) BioNumbers – the database of key numbers in molecular and cell biology. *Nucleic Acids Res.*, 38 (Database issue), D750–D753.
- 3 Pruitt, K.D., Brown, G.R., Hiatt, S.M. *et al.* (2014) RefSeq: an update on mammalian reference sequences. *Nucleic Acids Res.*, 42 (Database issue), D756–D763.
- 4 Hubbard, T., Barker, D., Birney, E. *et al.* (2002) The Ensembl genome database project. *Nucleic Acids Res.*, 30 (1), 38–41.
- 5 Cunningham, F., Amode, M.R., Barrell, D. *et al.* (2015) Ensembl 2015. *Nucleic Acids Res.*, 43 (D1), D662–D669.
- 6 UniProt Consortium (2015) UniProt: a hub for protein information. *Nucleic Acids Res.*, 43 (Database issue), D204–D212.
- 7 Berman, H.M., Westbrook, J., Feng, Z. *et al.* (2000) The Protein Data Bank. *Nucleic Acids Res.*, 28 (1), 235–242.
- 8 Mi, H., Muruganujan, A., and Thomas, P.D. (2013) PANTHER in 2013: modeling the evolution of gene function, and other gene attributes, in the context of phylogenetic trees. *Nucleic Acids Res.*, 41 (Database issue), D377–D386.
- 9 Mitchell, A., Chang, H.-Y., Daugherty, L. *et al.* (2015) The InterPro protein families database: the classification resource after 15 years. *Nucleic Acids Res.*, 43 (Database issue), D213–D221.
- 10 Hoffmann, R. and Valencia, A. (2004) A gene network for navigating the literature. *Nat. Genet.*, 36 (7), 664.
- 11 Fernández, J.M., Hoffmann, R., and Valencia, A. (2007) iHOP web services. *Nucleic Acids Res.*, 35 (web server issue), W21–W26.
- 12 Ashburner, M., Ball, C.A., Blake, J.A. *et al.* (2000) Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, 25 (1), 25–29.
- 13 Gene Ontology Consortium (2015) Gene Ontology Consortium: going forward. *Nucleic Acids Res.*, 43 (Database issue), D1049–D1056.
- 14 Kanehisa, M., Goto, S., Kawashima, S. *et al.* (2004) The KEGG resource for deciphering the genome. *Nucleic Acids Res.*, 32 (Database issue), D277–D280.
- 15 Goto, S., Okuno, Y., Hattori, M. *et al.* (2002) LIGAND: database of chemical compounds and reactions in biological pathways. *Nucleic Acids Res.*, 30 (1), 402–404.
- 16 Croft, D., Mundo, A.F., Haw, R. *et al.* (2014) The Reactome pathway knowledgebase. *Nucleic Acids Res.*, 42 (Database issue), D472–D477.
- 17 Milacic, M., Haw, R., Rothfels, K. *et al.* (2012) Annotating cancer variants and anti-cancer therapeutics in Reactome. *Cancers*, 4 (4), 1180–1211.
- 18 Kamburov, A., Wierling, C., Lehrach, H., and Herwig, R. (2009) ConsensusPathDB – a database for integrating human functional interaction networks. *Nucleic Acids Res.*, 37 (Database issue), D623–D628.
- 19 Kamburov, A., Pentchev, K., Galicka, H. *et al.* (2011) ConsensusPathDB: toward a more complete picture of cell biology. *Nucleic Acids Res.*, 39 (Database issue), D712–D717.
- 20 Kamburov, A., Stelzl, U., Lehrach, H., and Herwig, R. (2013) The ConsensusPathDB interaction database: 2013 update. *Nucleic Acids Res.*, 41 (Database issue), D793–D800.
- 21 Kelder, T., van Iersel, M.P., Hanspers, K. *et al.* (2012) WikiPathways: building research communities on biological pathways. *Nucleic Acids Res.*, 40 (Database issue), D1301–D1307.
- 22 Chang, A., Schomburg, I., Placzek, S. *et al.* (2015) BRENDA in 2015: exciting developments in its 25th year of existence. *Nucleic Acids Res.*, 43 (Database issue), D439–D446.
- 23 Wittig, U., Kania, R., Golebiewski, M. *et al.* (2012) SABIO-RK – database for biochemical reaction kinetics. *Nucleic Acids Res.*, 40 (Database issue), D790–D796.
- 24 Le Novère, N., Bornstein, B., Broicher, A. *et al.* (2006) BioModels Database: a free, centralized database of curated, published, quantitative kinetic models of biochemical and cellular systems. *Nucleic Acids Res.*, 34 (Database issue), D689–D691.
- 25 Li, C., Donizelli, M., Rodriguez, N. *et al.* (2010) BioModels Database: an enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Syst. Biol.*, 4, 92.
- 26 Chelliah, V., Juty, N., Ajmera, I. *et al.* (2015) BioModels: ten-year anniversary. *Nucleic Acids Res.*, 43 (Database issue), D542–D548.
- 27 Olivier, B.G. and Snoep, J.L. (2004) Web-based kinetic modelling using JWS Online. *Bioinformatics*, 20 (13), 2143–2144.
- 28 Hastings, J., de Matos, P., Dekker, A. *et al.* (2013) The ChEBI reference database and ontology for biologically relevant chemistry: enhancements for 2013. *Nucleic Acids Res.*, 41 (Database issue), D456–D463.
- 29 Pawson, A.J., Sharman, J.L., Benson, H.E. *et al.* (2014) The IUPHAR/BPS Guide to PHARMACOLOGY: an expert-driven knowledgebase of drug targets and their ligands. *Nucleic Acids Res.*, 42 (Database issue), D1098–D1106.
- 30 Mathelier, A., Zhao, X., Zhang, A.W. *et al.* (2014) JASPAR 2014: an extensively expanded and updated open-access database of transcription factor binding profiles. *Nucleic Acids Res.*, 42 (D1), D142–D147.
- 31 Zhao, F., Xuan, Z., Liu, L., and Zhang, M.Q. (2005) TRED: a Transcriptional Regulatory Element Database and a platform for *in silico* gene regulation studies. *Nucleic Acids Res.*, 33 (Database issue), D103–D107.
- 32 Jiang, C., Xuan, Z., Zhao, F., and Zhang, M.Q. (2007) TRED: a Transcriptional Regulatory Element Database, new entries and other development. *Nucleic Acids Res.*, 35 (Database issue), D137–D140.
- 33 Yusuf, D., Butland, S.L., Swanson, M.I. *et al.* (2012) The transcription factor encyclopedia. *Genome Biol.*, 13 (3), R24.
- 34 Edgar, R., Domrachev, M., and Lash, A.E. (2002) Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res.*, 30 (1), 207–210.
- 35 Barrett, T., Wilhite, S.E., Ledoux, P. *et al.* (2013) NCBI GEO: archive for functional genomics data sets – update. *Nucleic Acids Res.*, 41 (Database issue), D991–D995.
- 36 Kolesnikov, N., Hastings, E., Keays, M. *et al.* (2015) ArrayExpress update – simplifying data submissions. *Nucleic Acids Res.*, 43 (D1), D1113–D1116.
- 37 Gentleman, R.C., Carey, V.J., Bates, D.M. *et al.* (2004) Bioconductor: open software development for computational biology and bioinformatics. *Genome Biol.*, 5 (10), R80.

Software Tools for Modeling

17

Summary

Databases and high-throughput experiments are a rich source of data for modeling in systems biology. Many different tools in the form of programming languages and software packages are required and available to process and visualize these large quantities of data. The tools can roughly be divided into general-purpose and specialized programs. General-purpose tools such as Mathematica, Matlab, or R are enormously powerful packages for the numerical, symbolical, and visual analyses of arbitrary mathematical problems. However, their generality can also be a limitation, since they have a very steep learning curve, requiring considerable effort to get started.

Therefore, many specialized tools have been developed that are very restricted in their application range, but are much easier to use. Typical areas of specialization are the construction of biochemical reaction networks, the analysis of reaction networks (stability, flux analysis, and metabolic control theory) including parameter fitting, or the simulation of stochastic reactions.

A problem that arose with the multitude of tools was the lack of model compatibility. A model developed with one program had often to be reimplemented for use with a different tool. This important problem is now tackled with the development of model exchange languages. The Systems Biology Markup Language (SBML) is rapidly developing into a *de facto* standard with currently more than 280 tools supporting it (http://sbml.org/SBML_Software_Guide/SBML_Software_Matrix).

The databases described in Chapter 16 are huge repositories for the biological data that have been gathered by various techniques. The information in the databases represents raw material for most types of modeling efforts. Modeling tools help us to formulate theoretical ideas and hypothesis and to extract information relevant to these hypotheses from the raw material stored in the databases. Chapter 5 provided a first overview of the most popular modeling tools and data formats. However, there are

- 17.1 [13C-Flux2](#)
- 17.2 [Antimony](#)
- 17.3 [Berkeley Madonna](#)
- 17.4 [BIOCHAM](#)
- 17.5 [BioNetGen](#)
- 17.6 [Biopython](#)
- 17.7 [BioTapestry](#)
- 17.8 [BioUML](#)
- 17.9 [CellDesigner](#)
- 17.10 [CellNetAnalyzer](#)
- 17.11 [Copasi](#)
- 17.12 [CPN Tools](#)
- 17.13 [Cytoscape](#)
- 17.14 [E-Cell](#)
- 17.15 [EvA2](#)
- 17.16 [FEniCS Project](#)
- 17.17 [Genetic Network Analyzer \(GNA\)](#)
- 17.18 [Jarnac](#)
- 17.19 [JDesigner](#)
- 17.20 [JSim](#)
- 17.21 [KNIME](#)
- 17.22 [libSBML](#)
- 17.23 [MASON](#)
- 17.24 [Mathematica](#)
- 17.25 [MathSBML](#)
- 17.26 [Matlab](#)
- 17.27 [MesoRD](#)
- 17.28 [Octave](#)
- 17.29 [Omix Visualization](#)
- 17.30 [OpenCOR](#)
- 17.31 [Oscill8](#)

[17.32 PhysioDesigner](#)[17.33 PottersWheel](#)[17.34 PyBioS](#)[17.35 PySCeS](#)[17.36 R](#)[17.37 SAAM II](#)[17.38 SBMLEditor](#)[17.39 SemanticSBML](#)[17.40 SBML-PET-MPI](#)[17.41 SBMLsimulator](#)[17.42 SBMLSqueezer](#)[17.43 SBML Toolbox](#)[17.44 SBtoolbox2](#)[17.45 SBML Validator](#)[17.46 SensA](#)[17.47 SmartCell](#)[17.48 STELLA](#)[17.49 STEPS](#)[17.50 StochKit2](#)[17.51 SystemModeler](#)[17.52 Systems Biology Workbench](#)[17.53 Taverna](#)[17.54 VANTED](#)[17.55 Virtual Cell \(VCell\)](#)[17.56 xCellerator](#)[17.57 XPPAUT](#)[Exercises](#)[References](#)

many more. To provide the reader with a rough overview of the plethora of other tools that exist and that could not be discussed, we have included here a compendium that is based on reviews and surveys from the literature [1,2], as well as various Internet sources. The short description also provides information about the availability, native operating system, and the time when the last update was released (as of May 2015). Tools that have not been updated since 2010 are not included in this list, since that is a strong indication that the active development and maintenance has stopped. Although the compendium contains more than 50 tools, we realize that this cannot be an exhaustive list since the development and usage of software tools is a very dynamic area.

Furthermore, because of the large number of programs, we don't claim to be experts for all the programs described here. The description is thus either based on our own experience or paraphrased from the information given at the respective website. We therefore apologize in advance for personal bias and ignorance that entered into the construction of the compendium.

17.1 13C-Flux2

13C-FLUX2 is a high-performance simulator for ^{13}C -base metabolic flux analysis. All complex interactions between the cellular networks of genes, transcripts, proteins, and metabolites finally result in metabolic fluxes. However, *in vivo* fluxes are not directly observable and have to be inferred by data from labeling experiments and the use of mathematical models. ^{13}C -based metabolic flux analysis emerged as the state-of-the-art technique in the field of fluxomics. 13C-FLUX2 is a software suite of applications for the detailed quantification of intracellular steady-state fluxes.

URL: www.13cflux.net/13cflux2

Latest version: 13C-FLUX2, January 2013

<i>Operating system:</i>	Windows <input type="checkbox"/>	Mac OS <input type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input type="checkbox"/>	Free for academic <input checked="" type="checkbox"/>	Commercial: <input checked="" type="checkbox"/>

17.2 Antimony

Antimony is a text-based model definition language originally based on Jarnac, and extended to be fully modular. Antimony models can be converted to and from SBML, flattening the modularity in the process. Antimony was designed as a successor to Jarnac's model definition language, with some new features that mesh with newer elements of SBML. A programming library, libAntimony, exists in tandem with the language to allow computer translation of Antimony-formatted models into SBML and other formats.

URL: antimony.sourceforge.net/

Latest version: Antimony 2.8, October 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.3

Berkeley Madonna

Berkeley Madonna is a fast, general-purpose differential and difference equations solver. It is also possible to perform discrete simulations using conveyors, ovens, and queues. Furthermore, (bio)chemical reactions can be entered in a special shorthand syntax (e.g., $A + 2B \leftrightarrow C + D$) and Berkeley Madonna can automatically find parameter values that minimize the deviation between the model output and a given data set (i.e., parameter fitting). Developed on the Berkeley campus under the sponsorship of NSF and NIH, it is currently used by academic and commercial institutions for constructing mathematical models for research and teaching. The number crunching engine of Berkeley Madonna was originally written in C, and later extended with the Flowchart graphical interface written in Java. Currently, a version of Berkeley Madonna, called JMadonna, is in development. It will have the user interface written in Java, while retaining the simulation engine in C for speed. Also, a Linux version of JMadonna is planned. Student licenses are available at reduced prices as well as a demo version with restricted features (e.g., no saving).

URL: www.berkeleymadonna.com

Latest version: Berkeley Madonna 8.3, 2010

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input type="checkbox"/>
<i>Availability:</i>	Free for all <input type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input checked="" type="checkbox"/>

17.4

BIOCHAM

BIOCHAM (Biochemical Abstract Machine) is a programming environment for modeling biochemical systems, performing simulations and querying the model in temporal logic. It has a simulator for Boolean, stochastic, and differential models that can be accessed via a GUI. It has its own rule-based modeling language, which is compatible with SBML. The stand-alone versions as well as a Web-based version come with various example models.

URL: contraintes.inria.fr/Biocham

Latest version: BIOCHAM 3.7, July 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.5

BioNetGen

BioNetGen is a tool for automatically generating mathematical models of biological systems from user-specified rules for molecular interactions. Rules are specified in the BioNetGen language (BNGL), which enables precise, visual, and extensible representation of molecular interactions. The language was designed with protein–protein interactions in mind. A user can explicitly indicate the parts of proteins involved in an interaction, the conditions upon which an interaction depends, the connectivity of proteins in a complex, and other aspects of protein–protein interactions. It is one of the few software tools available for generating physicochemical models of systems marked by combinatorial complexity. Apart from the stand-alone versions for the different operating systems, a Web-based version is also available.

URL: www.bionetgen.org

Latest version: BioNetGen 2.2.6, June 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.6

Biopython

Biopython is a Python package that provides a programmatic interface to a large number of bioinformatics databases and tools. Some functions (e.g., downloading and analyzing of nucleotide and protein sequences) can be performed by Biopython alone; for others (“blasting” sequences or calculating phylogenetic trees), the necessary programs have to be installed and Biopython just provides a convenient interface. Reading and writing of DNA, RNA, or protein sequences is, of course, a central feature of Biopython using the subpackages Bio.Seq, Bio.SeqRecord and Bio.SeqIO. Further routines are available to align (Bio.AlignIO) or blast (Bio.Blast.NCBIWWW) sequences. Proteins can be searched in SwissProt (Bio.ExPASy, Bio.SwissProt) and the PDB database for 3D structures (Bio.PDB). It is also possible to cluster sequences (Bio.Cluster) or to construct phylogenetic trees (Bio.Phylo). Finally, the Bio.Entrez module provides routines to search the NCBI sequence databases, but also literature records in PubMed. A detailed description of all the modules is available at <http://biopython.org/DIST/docs/tutorial/Tutorial.html>.

URL: biopython.org

Latest version: BioPython 1.66, October 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.7 BioTapestry

BioTapestry is an interactive tool for building, visualizing, and simulating genetic regulatory networks. BioTapestry is designed around the concept of a developmental network model, and is intended to deal with large-scale models with consistency and clarity. It is capable of representing systems that exhibit increasing complexity over time, such as the genetic regulatory network controlling endomesoderm development in sea urchin embryos. BioTapestry has the ability to create and handle sets of submodels, which is helpful for structuring large networks in a readable way. These models can be exported in SBML format for dynamic simulation in other tools.

URL: www.biotapecstry.org

Latest version: BioTapestry 7, September 2014

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.8 BioUML

BioUML is an open-source integrated Java platform that spans a comprehensive range of capabilities, including access to databases with experimental data, tools for formalized description of biological systems, structure, and functioning, and tools for their visualization, simulation, parameters fitting, and analyses. Due to scripts (R, JavaScript) and workflow support, it provides powerful possibilities for the analyses of high-throughput data. The plug-in-based architecture allows adding new functionality using plug-ins. The system consists of the BioUML server and the BioUML workbench, which can work in a stand-alone mode or in connection with the server. Models written in SBML, CellML, or BioPAX can be used.

URL: www.biouml.org

Latest version: BioUML 0.9.6, November 2013

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.9 CellDesigner

CellDesigner is a structured diagram editor for drawing gene regulatory and biochemical networks. Networks are drawn based on the Systems Biology Graphical Notation (SBGN) and stored using the Systems Biology Markup Language, a standard for representing models of biochemical and gene regulatory networks. CellDesigner can connect to various databases to download models, reaction parameters, and literature information. Models can also be simulated using an internal ODE solver or via a connection to the Copasi solver. Furthermore, models can also be analyzed via a link to the Systems Biology Workbench (SBW). A more in-depth description of CellDesigner is given in Chapter 5.

URL: celldesigner.org

Latest version: CellDesigner 4.4, July 2014

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.10 CellNetAnalyzer

CellNetAnalyzer is a package for Matlab and provides a comprehensive and user-friendly environment for structural and functional analyses of biochemical and cellular networks. CellNetAnalyzer facilitates the analysis of metabolic as well as signaling and regulatory networks solely on their network topology, that is, independent of kinetic mechanisms and parameters. The core concept of visualization and interactivity is realized by interactive network maps where the abstract network model is linked with network graphics. CellNetAnalyzer provides a powerful collection of tools and algorithms for structural network analysis, which can be started in a menu-controlled manner within the interactive network maps. API functionalities have been added to enable interested users to call algorithms of CellNetAnalyzer from external programs.

URL: <http://www.mpi-magdeburg.mpg.de/projects/cna/cna.html>

Latest version: CellNetAnalyzer 2015.1

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input type="checkbox"/>	Free for academic <input checked="" type="checkbox"/>	Commercial: <input type="checkbox"/>

17.11 Copasi

Copasi (Complex Pathway Simulator) is an application for the simulation and analysis of biochemical networks. It features stochastic and deterministic time course simulation, steady-state analysis, metabolic control analysis (MCA), parameter scans, optimization of arbitrary target functions, and parameter estimation. Copasi provides means to visualize data in customizable plots, histograms, and animations of network diagrams. Models are normally saved in Copasis own format, but import/export of SBML is supported up to Level 3 Version 1. A more detailed description of Copasi is given in Chapter 5.

URL: www.copasi.org

Latest version: Copasi 4.16 (Build 104), August 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.12 CPN Tools

CPN Tools is a tool for editing, simulating, and analyzing Colored Petri Nets as well as noncolored nets. The tool features incremental syntax checking and code generation while a net is being constructed. A fast simulator efficiently handles both untimed and timed nets. Full and partial state spaces can be generated and analyzed, and a standard state space report contains information such as boundedness properties and liveness properties. Models can be saved in the Petri Net Markup Language (PNML).

URL: cpn-tools.org

Latest version: CPN Tools 4.0.1, February 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input type="checkbox"/>	Linux <input type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.13 Cytoscape

Cytoscape is an open-source software platform for visualizing molecular interaction networks and biological pathways as well as integration of these networks with annotations, gene expression profiles, and other state data. Although Cytoscape was originally designed for biological research, now it is a general platform for complex network analysis and visualization. The Cytoscape core distribution provides a basic set of features for data integration, analysis, and visualization. Additional features are available as Apps (formerly called Plugins). Apps are available for network and molecular profiling analyses, new layouts, additional file format support, scripting, and connection with databases. They may be developed by anyone using the Cytoscape open API based on Java and App development is welcomed and encouraged. Most of the Apps are freely available from Cytoscape App Store.

URL: www.cytoscape.org

Latest version: Cytoscape 3.3.0, November 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.14 E-Cell

The E-Cell Project develops general technologies and theoretical support for computational biology with the grand aim to make precise whole-cell simulation at the molecular level possible. Apart from modeling methodologies, formalisms, and techniques, the project developed the E-Cell System, a software platform for modeling, simulation, and analysis of complex, heterogeneous, and multiscale systems like the cell. The source code for E-Cell is available at GitHub (github.com/).

URL: www.e-cell.org

Latest version: E-Cell4, March 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input type="checkbox"/>	Linux <input type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.15 EvA2

EvA2 (an Evolutionary Algorithms framework, revised version 2) is a comprehensive heuristic optimization

framework with emphasis on Evolutionary Algorithms implemented in Java. Eva2 integrates several derivation-free optimization methods, preferably population based, such as Evolution Strategies (ES), Genetic Algorithms (GA), Differential Evolution (DE), Particle Swarm Optimization (PSO), and classical techniques such as multistart Hill Climbing or Simulated Annealing. Besides typical single-objective problems, multimodal and multiobjective problems are handled directly by the Eva2 framework. Via the Java mechanism of Remote Method Invocation (RMI), the algorithms of Eva2 can be distributed over network nodes based on a client–server architecture. Eva2 aims at two groups of users: first, the end user who does not know much about the theory of evolutionary algorithms but wants to use it to solve an application problem; second, the scientific user who wants to investigate the performance of different optimization algorithms or wants to compare the effect of alternative or specialized evolutionary or heuristic operators. The latter usually knows more about evolutionary algorithms or heuristic optimization and is able to extend Eva2 by adding specific optimization strategies or solution representations. The software consists of a workbench GUI to construct and control the optimization problem and the number crunching core.

URL: <http://www.ra.cs.uni-tuebingen.de/software/JavaEva/>

Latest version: Eva 2.1.0, September 2013

Operating system:	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
Availability:	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.16 FEniCS Project

The FEniCS Project is a collection of free software tools with an extensive list of features for automated, efficient solution of partial differential equations by finite element methods. FEniCS has an extensive list of features for automated, efficient solution of differential equations, including automated solution of variational problems, automated error control and adaptivity, a comprehensive library of finite elements, high-performance linear algebra, and many more. FEniCS is organized as a collection of interoperable components that together form the FEniCS Project. These components include the problem-solving environment DOLFIN, the form compiler FFC, the finite element tabulator FIAT, the just-in-time compiler Instant, the code generation interface UFC, and the form language UFL. DOLFIN is a c++/Python library that functions as the main user interface of FEniCS. A large part of the functionality of FEniCS is implemented as part of

DOLFIN. It provides a problem-solving environment for models based on partial differential equations and implements core parts of the functionality of FEniCS, including data structures and algorithms for computational meshes and finite element assembly. To provide a simple and consistent user interface, DOLFIN wraps the functionality of other FEniCS components and external software, and handles the communication between these components.

URL: fenicsproject.org/

Latest version: FEniCS 1.6.0, July 2015

Operating system:	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
Availability:	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.17 Genetic Network Analyzer (GNA)

Genetic Network Analyzer (GNA) is a computer tool for the modeling and simulation of genetic regulatory networks. The aim of GNA is to assist biologists and bioinformaticians in constructing a model of a genetic regulatory network using knowledge about regulatory interactions in combination with gene expression data. Genetic Network Analyzer consists of a simulator of qualitative models of genetic regulatory networks in the form of piecewise linear differential equations. Instead of exact numerical values for the parameters, which are often not available for networks of biological interest, the user of GNA specifies inequality constraints. This information is sufficient to generate a state transition graph that describes the qualitative dynamics of the network. The simulator has been implemented in Java and has been applied to the analysis of various regulatory systems, such as the networks controlling the initiation of sporulation in *Bacillus subtilis* and the carbon starvation response in *Escherichia coli*.

URL: ibis.inrialpes.fr/article122.html

Latest version: GNA 8.5.0.1, December 2013

Operating system:	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
Availability:	Free for all <input type="checkbox"/>	Free for academic <input checked="" type="checkbox"/>	Commercial: <input type="checkbox"/>

17.18 Jarnac

Jarnac is the numerical solver that is part of the Systems Biology Workbench (SBW). There it is used with the network designer JDesigner, but it can also import SBML files produced by other tools. In addition, Jarnac can also

be used as a stand-alone tool with its own language for describing reaction networks. This simple control language is similar to the Basic language and supports many of the constructs one would expect, for loops, conditionals, while/do, and repeat/until. It supports several different data types, including integers, floats, Booleans, strings, vectors, matrices, and lists. Jarnac also supports user-defined functions and external modules. There is built-in computational support for dynamic simulation (using LSODA or CVODE integrator), steady-state analysis using the NLEQ solver, simple stability analysis (eigenvalues, using IMSL library), matrix arithmetic (all the main operators, including transpose, det, etc., using IMSL library), metabolic control analysis (all steady-state control coefficients and elasticities), metabolic structural analysis (null space and conservation relation analysis and others), and stochastic simulation (using standard Gillespie method).

URL: <http://sbw-app.org/jarnac/>

Latest version: Jarnac 3.32a, May 2013

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input type="checkbox"/>	Linux <input type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.19 JDesigner

Like Jarnac, JDesigner is a component of the Systems Biology Workbench (SBW). It is a visual design tool for biochemical networks and uses SBML as its native file format. SBW has to be installed before JDesigner can be installed. JDesigner runs under MS-Windows and allows users to draw a biochemical network and save it in SBML. JDesigner has an SBW interface that allows it to be called from other SBW compliant modules. In addition, JDesigner has the ability to use Jarnac as a simulation server (via SBW), thus allowing models to be run from within JDesigner. In this mode, JDesigner is both a network design tool and a simulator.

URL: <http://sbw-app.org/jdesigner/>

Latest version: JDesigner 2.4.7, May 2013

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input type="checkbox"/>	Linux <input type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.20 JSim

JSim is a Java-based simulation system for building quantitative numeric models and analyzing them with

respect to experimental reference data. JSim's primary focus is on physiology and biomedicine; however, its computational engine is quite general and applicable to a wide range of scientific domains. JSim models may intermix ODEs, PDEs, implicit equations, integrals, summations, discrete events, and procedural code as appropriate. JSim's model compiler can automatically insert conversion factors for compatible physical units as well as detect and reject unit unbalanced equations. JSim models are normally written using its own Mathematical Modeling Language (MML), but JSim also imports flawlessly models written in SBML or CellML and automatically converts them into MML code. Once loaded, models can be simulated and the results graphically displayed as XY, contour, or surface plots. Models can also be analyzed via a sensitivity analysis and automatic parameter optimization to fit a model to experimental data.

URL: www.physiome.org/jsim

Latest version: JSim 2.16, December 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input type="checkbox"/>	Free for academic <input checked="" type="checkbox"/>	Commercial: <input type="checkbox"/>

17.21 KNIME

KNIME (Konstanz Information Miner) is a user-friendly graphical workflow management system for the entire data analysis process, including data access, data transformation, initial investigation, predictive analytics, visualization, and reporting. The open integration platform provides over 1000 modules (nodes), including those of the KNIME community and its extensive partner network. The strength of KNIME is not so much in the area of kinetic modeling, but in the graphical creation of workflows for the analysis of data via clustering, classification, descriptive statistics, and visualization. The workbench itself resembles strongly the Eclipse IDE for program development and is centered around a large canvas on which icons (nodes) are dropped and connected to form the data workflow.

URL: www.knime.org/

Latest version: KNIME 3.1, December 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.22 libSBML

libSBML is a library designed to help you read, write, manipulate, translate, and validate SBML files and data streams. It is not an application itself (although it does come with many example programs), but rather a library you can embed in your own applications. libSBML is written in ISO C and C++, but as a library it may be used from many different programming languages such as C/C++, C#, Java, Python, Perl, Ruby, or Matlab. libSBML offers powerful features such as reading/writing compressed SBML files, support for SBML Level 3 packages, detecting overconstrained models, checking units, an API for SBML <annotation> content, and support for the three most popular XML parser libraries: Xerces, Expat, and libxml2.

URL: <http://sbml.org/Software/libSBML>

Latest version: libSBML 5.12.0, November 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.23 MASON

MASON is a fast Java library for the construction of multiagent simulations and designed to be the foundation for large custom-purpose Java simulations. MASON contains both a model library and an optional suite of visualization tools in two and three dimensions. Models are completely independent of visualization, which can be added, removed, or changed at any time. This separation makes it possible to run computationally demanding simulations on high-performance servers, while the visualization is performed later on a local computer. MASON can represent continuous, discrete, or hexagonal 2D, 3D, or Network data, and any combination of it. Provided visualization tools can display these environments in two or three dimension, by scaling, scrolling, or rotating them as needed.

URL: <http://cs.gmu.edu/~eclab/projects/mason/>

Latest version: MASON 19, June 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.24 Mathematica

Mathematica is a general-purpose tool for the calculation and visualization of any type of mathematical model. It is produced by Wolfram Research (www.wolfram.com) and exists for the operating systems Microsoft Windows, Macintosh, Linux, and several Unix variants. The Mathematica system consists of two components: the kernel that runs in the background performing the calculations and the graphical user interface (GUI) that communicates with the kernel. The GUI has the form of a so-called notebook that contains all the input, output, and graphics. Apart from its numerical calculation and graphics abilities, Mathematica is known for its capability to perform advanced symbolic calculations. Mathematica can be used either by interactively invoking the available functions or by making use of the built-in programming language to write larger routines and programs, which are also stored as or within notebooks. For many specialized topics, Mathematica packages (a special kind of notebook) are available that provide additional functionality. J/Link, .NET/Link, and MathLink, products that ship with Mathematica, enable the two-way communication with Java, .NET, or C/C++ code. This means that Mathematica can access external code written in one of these languages and that the Mathematica kernel can actually be called from other applications. The former is useful if an algorithm has already been implemented in one of these languages or to speed up time-critical calculations that would take too long if implemented in Mathematica itself. In the latter case, other programs can use the Mathematica kernel to perform high-level calculations or render graphic objects. Besides an excellent help-utility, there are also many sites on the Internet that provide additional help and resources. The site mathworld.wolfram.com contains a large repository of contributions from Mathematica user all over the world. If a function or algorithm does not exist in Mathematica, it is worthwhile to check this site before implementing it yourself. If questions and problems arise during the use of Mathematica, a valuable source of help is also the newsgroup: <http://news://comp.soft-sys.math.mathematica>.

URL: <http://www.wolfram.com/mathematica/>

Latest version: Mathematica 10.1, March 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input checked="" type="checkbox"/>

17.25 MathSBML

MathSBML is an open-source package for working with SBML models in Mathematica. It contains three functions that can be invoked directly: SBMLRead, SBMLNDSolve, and SBMLPlot. In addition, the MathSBML Model Builder consists of a suite of functions that can be used to build SBML Models manually. SBMLRead is the primary function provided by this package; it reads a model encoded in SBML into Mathematica, converts the model into a system of differential and possibly algebraic equations, and can generate a formatted listing of the model. SBMLNDSolve is used to solve the system of differential-algebraic equations produced by SBMLRead. SBMLPlot can be used to generate plots of the resulting solutions. Plots can also be generated directly with the Mathematica Plot command. MathSBML supports SBML Level 1 Version 2 as well as Level 2 Version 3.

URL: <http://sourceforge.net/projects/xlr8r/files/mathsbml/>
Latest version: MathSBML 1203-002-1102, March 2012

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.26 Matlab

Like Mathematica, Matlab is also a general-purpose computation package. In many respects, both products are very similar and it is up to the taste of the user which one to prefer. Matlab is available for the same platforms as Mathematica, has very strong numerical capabilities, and can also produce many different forms of graphics. It too has its own programming language and functions are stored in the so-called M-files. Toolboxes (special M-files) add additional functionality to the core Matlab distribution and like Mathematica, Matlab can be called by external programs to perform high-level computations. A repository exists for user-contributed files (<http://www.mathworks.com/matlabcentral/fileexchange> and <http://www.mathworks.net/MATLAB/toolboxes.html>) as well as a newsgroup (<http://news://comp.soft-sys.matlab>) for getting help. Although still slower than traditional programming languages like C/C++ or Java, Matlab code runs generally faster than Mathematica code.

URL: <http://www.mathworks.com/products/matlab>
Latest version: Matlab R2015b, 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input checked="" type="checkbox"/>

17.27 MesoRD

MesoRD (Mesoscopic Reaction Diffusion Simulator) is a tool for stochastic and deterministic simulation of chemical reactions and diffusion in three dimension, and planar 2D spaces. The description of the system that should be simulated is written in the SBML file format. In addition to the SBML file, MesoRD requires information about how the simulation should be executed, such as spatial discretization of the reaction volume, duration of the simulation, visualization, output options, and for deterministic simulations, also choice of integration method. These parameters are given through the MesoRD user interface. The output files from MesoRD are intended for external data analysis and visualization packages, for instance, the freely distributed MesoRD Matlab toolbox available from the MesoRD website. In both the deterministic and the stochastic modes of simulation, the reaction volume is discretized into a large number of small subvolumes and the state of the system is given by the number of molecules per subvolume. In the stochastic simulation, the number of molecules per subvolume is discrete. Furthermore, the reaction and diffusion events that change the number of molecules are probabilistic, in the sense that the next event in the system is sampled from a distribution function. In the deterministic simulation, the state is assumed to be a continuous variable and the change in the number of molecules per time unit is given by the average change as defined from the stochastic model. In addition to precompiled binaries for Windows, the C++ source code is also available.

URL: mesord.sourceforge.net/
Latest version: MesoRD 1.1, October 2012

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input type="checkbox"/>	Linux <input type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.28 Octave

GNU Octave is a high-level interpreted language, primarily intended for numerical computations. It provides

capabilities for the numerical solution of linear and nonlinear problems, and for performing other numerical experiments. It also provides extensive graphics capabilities for data visualization and manipulation. Octave is normally used through its interactive command line interface, but it can also be used to write noninteractive programs. The aim of Octave is that all code that runs in Matlab should also run in Octave. In practice, however, most Matlab programs, except for very simple ones, require modifications. To run under Windows, the Cygwin library (www.cygwin.com) has to be installed.

URL: <https://www.gnu.org/software/octave/>

Latest version: Octave 4.0.0, May 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.29 Omix Visualization

Omix® is a user-friendly and highly customizable editor and modeling tool for metabolic network diagrams, equipped with extensive data visualization features. Main application fields are the interactive mapping of multi-omics data in the direct context of network drawings, in particular in the fields of transcriptomics, metabolomics, and fluxomics. Omix features the drawing of high-quality network diagrams, interactive visualization of experimental data, analysis of time-dependent data by animations, modeling of ^{13}C isotope labeling experiments as input for 13C FLUX2, and static network analysis via elementary flux models, network dependencies, and flux balances.

URL: omix-visualization.com

Latest version: Omix 1.8.13, April 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input checked="" type="checkbox"/>

17.30 OpenCOR

OpenCOR is an open-source modeling environment that is supported on Windows, Linux, and OS X [3]. It relies on a modular approach, which means that all of its features come in the form of plug-ins. These plug-ins can be used to organize, edit, annotate, simulate, and analyze models encoded in the CellML (up to 1.1) format.

Together with JSim, OpenCOR is the tool of choice for working with CellML files.

URL: www.opencor.ws

Latest version: OpenCOR 0.4.1, May 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.31 Oscill8

Oscill8 is a suite of tools for analyzing large systems of ODEs, particularly with respect to understanding how the high-dimensional parameter space controls the dynamics of the system. It features time course integration, one- and two-parameter bifurcation diagrams, and bifurcation searches. It also exposes lower level numerical control parameters to the expert user, including a “raw” AUTO interface (see also XPPAUT).

URL: oscill8.sourceforge.net/

Latest version: Oscill8 2.0.11, February 2011

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input type="checkbox"/>	Linux <input type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.32 PhysioDesigner

PhysioDesigner is an open platform that supports multi-level modeling of physiological systems in the field of integrated life sciences and systems biology, including physiology and neuroscience. Users can combine and build mathematical models of biological and physiological functions in PhysioDesigner. Users can also integrate morphometric data into a model, which is used, for example, to define a domain in which partial differential equations are solved. The models developed by PhysioDesigner are stored in PHML (Physiological Hierarchy Markup Language) format, which is an XML-based specification, to describe a wide variety of models of biological and physiological functions with a hierarchical structure. PhysioDesigner also allows the creation of SBML-PHML hybrid models, which is a novel way of creating multilevel physiological systems. Simulation of the models created by PhysioDesigner can be performed using the Flint simulator, which is developed concurrently with PhysioDesigner. Additionally, PhysioDesigner can export C++ and

Java source code, including numerical integration solvers. Thus, users can easily perform simulations by compiling them. Finite element simulation for partial differential equations can be done by exporting a model in FreeFem++ format and running the script on FreeFem++.

URL: physiodesigner.org/

Latest version: PhysioDesigner 1.3.1, July 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.33 PottersWheel

PottersWheel is a Matlab toolbox for the construction and analysis of kinetic mathematical models. A model can be entered as a system of ordinary differential equations (ODEs) or loaded from a SBML file. The model is displayed as reaction network, but the real strength of PottersWheel lies in its parameter fitting capabilities. The tool allows the simultaneous fitting of multiple data sets offering a range of different fitting algorithms. In addition to the parameter values, confidence intervals are also calculated and an identifiability analysis is performed. Since fitting problems can be computationally very demanding, the code can be run on a cluster of computers. PottersWheel can be controlled via scripts or a graphical user interface, the use of which is described by several video tutorials that are available on the website. The results are represented by different plots and can also be saved in a report (available as PDF, doc, or html).

URL: www.potterswheel.de/

Latest version: PottersWheel 4.0, 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input checked="" type="checkbox"/>

17.34 PyBioS

PyBioS is an integrated, Web-based software platform for the design, modeling, and simulation of cellular systems, in particular for human and mouse. Models in PyBioS can be constructed in a Web-based user interface that defines different objects for cellular components, such as genes, mRNAs, proteins, and metabolites as well as cellular compartments. PyBioS keeps

track of accession numbers of these entities, such as ENSEMBL gene, transcript or protein identifiers, or ChEBI and KEGG IDs. As each cellular component is defined by a reference entity in the underlying PyBioS database, it allows a modular design of models that can subsequently be merged into larger and more comprehensive models. This supports the easy reuse of models. Models can also be imported from external sources. An interface to the meta-pathway database ConsensusPathDB (see Chapter 16) that is integrating several public pathway resources allows the automatic import of reactions and pathways from databases such as KEGG and Reactome. Models can also be imported from [BioModels.org](http://biomodels.org) or directly as SBML. PyBioS comes along with several predefined kinetic laws that simplify the definition of new reactions. For quantitative simulation, ODE systems of the models are created by PyBioS on the fly. Simulation results can either be displayed in a diagram or they can be uploaded into the models network graph. This allows the animation of the changes of fluxes and concentrations by appropriate color codes of reaction and species nodes.

URL: pybios.molgen.mpg.de/

Latest version: PyBioS 3, November 2014

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input type="checkbox"/>	Free for academic <input checked="" type="checkbox"/>	Commercial: <input type="checkbox"/>

17.35 PySCeS

PySCeS (Python Simulator for Cellular Systems) is written in Python and provides a variety of tools for the simulation and analysis of cellular systems. The input is via a text-based model description language. Solvers for time course integration (LSODA, CVODE) and steady-state calculations (HYBRD, NLEQ2) exist. Various modules perform metabolic control analysis (i.e., elasticities, flux, and concentration control coefficients) and bifurcation analysis. The package also allows 1D and 2D parameter scans and their visualization via a flexible plotting interface (Matplotlib and/or Gnuplot). PySCeS can import and export SBML and is developed as open-source software. The use of a proper programming language gives PySCeS a great flexibility, but it should be clear that models are not as easy to construct as in GUI-based modeling tools.

URL: pysces.sourceforge.net/

Latest version: PySCeS 0.9.1, December 2014

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.36

R

R is a free programming environment that is especially designed and used to provide a wide variety of statistical tests and techniques. It consists of the R language, an integrated development environment (IDE), and a large set of (statistical) packages that can be downloaded from the Comprehensive R Archive Network (CRAN). Especially, two features make R interesting for biostatistical and bioinformatical analyses: first, the rich choice of publication quality plots that can be generated with R; second, the fact that new (bio)statistical analysis methods are often first available as R packages. Of special interest in this respect is the Bioconductor project (www.bioconductor.org), which consists of a collection of R packages for the analysis of high-throughput genomic data such as sequence, expression, and interaction data. Although R is a complete programming language, it is rarely used for “normal” programming tasks, since it is very slow compared to languages like Java or C++.

URL: www.r-project.org/

Latest version: R 3.2.3, December 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.37

SAAM II

SAAM II (Simulation Analysis and Modeling) is a modeling, simulation, and analysis software package that supports the development and statistical calibration of compartmental models in biological, metabolic, and pharmaceutical systems. Mathematically, these models translate into systems of ordinary linear or nonlinear differential equations. A flexible graphical user interface makes it easy for researchers with diverse backgrounds to use the software. The Epsilon Group, (TEG) acquired all rights of the SAAM II software packages from the University of Washington in early 2011 and added several features since then. During the modeling process, it is sometimes necessary to

understand the effect that data collection errors and collection times can have in identifying model parameters. To improve this process, experiments can be simulated using synthetic data with a variety of errors. Users also have the possibility to create sensitivity plots for a better understanding of relationships between the variables and compartments when there is uncertainty. Furthermore, it is also possible to investigate the robustness of models and select the best alternative model among competing alternatives. A variety of outputs can be automatically generated, including parameter values and their confidence intervals, statistical information, or plots (including experiment simulation, sensitivity plots, and batch processing). Support for SBML is in development, but not yet supported.

URL: <http://tegvirginia.com/solutions/saam-ii/>

Latest version: SAAM II, 2.3, November 2013

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input type="checkbox"/>
<i>Availability:</i>	Free for all <input type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input checked="" type="checkbox"/>

17.38

SBMLEditor

The Systems Biology Markup Language is a free and open interchange format for encoding computational models of biological processes. SBML aims to be a systems to systems format, and users are not expected to write SBML files manually. SBMLEditor is a simple, low-level editor for SBML files. It allows you to manipulate SBML elements in a controlled way while maintaining the validity of the final file. SBMLEditor also allows you to create and modify annotations, as defined in the SBML specifications. The editor supports up to SBML Level 3 Version 1 and checks the validity of the SBML code whenever it is saved.

URL: <https://www.ebi.ac.uk/compneur-srv/SBMLEditor.html>

Latest version: SBMLEditor 2.0-b1, June 2012

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.39

SemanticSBML

SemanticSBML is a suite of software tools for SBML models, supporting users in the annotation, alignment, and

merging of models. During model merging, the software detects inconsistencies between model elements and allows users to change the element alignment and to resolve conflicts. Using semantic annotations in SBML models, it also allows for a semantic clustering of models and for a ranked retrieval from models in the BioModels database at www.ebi.ac.uk/biomodels-main. SemanticSBML can be used online at www.semanticsbml.org. Free python code for programmers, including code for element comparison by semantic similarities, is freely available.

URL: www.semanticsbml.org

Latest version: SemanticSBML 2.0, September 2010

Operating system:	<input checked="" type="checkbox"/> Windows	<input checked="" type="checkbox"/> Mac OS	<input checked="" type="checkbox"/> Linux
Availability:	<input checked="" type="checkbox"/> Free for all	<input type="checkbox"/> Free for academic	<input type="checkbox"/> Commercial

17.40 SBML-PET-MPI

SBML-PET-MPI is a parallel parameter estimation tool for SBML-based models. The tool allows the user to perform parameter estimations by fitting multiple data sets. The tool performs an uncertainty and identifiability analysis of the estimated parameters using maximum likelihood and bootstrap methods. Models containing events are supported as well as those with constraints. The software uses the message parsing interface (MPI) protocol for parallelization and achieves good scalability with the number of available processors. To run under Windows, the Cygwin library (www.cygwin.com) has to be installed.

URL: <http://www.bioss.uni-freiburg.de/cms/sbml-pet-mpi.html>

Latest version: SBML-PET-MPI 1.2, September 2011

Operating system:	<input checked="" type="checkbox"/> Windows	<input checked="" type="checkbox"/> Mac OS	<input checked="" type="checkbox"/> Linux
Availability:	<input checked="" type="checkbox"/> Free for all	<input type="checkbox"/> Free for academic	<input type="checkbox"/> Commercial

17.41 SBMLsimulator

SBMLsimulator is a fast, accurate, and easily usable program for dynamic model simulation and heuristic parameter optimization of models encoded in the Systems Biology Markup Language. In order to ensure a high reliability of this software, it has been benchmarked against the entire SBML Test Suite and all models from the Biomodels.net

database. It includes a large collection of nature-inspired heuristic optimization procedures for efficient model calibration. SBMLsimulator provides an intuitive graphical user interface and several command-line options to be suitable for large-scale batch processing and model calibration. The simulation core library of SBMLsimulator can be obtained as a separate application programming library.

URL: <http://www.ra.cs.uni-tuebingen.de/software/SBMLsimulator/>

Latest version: SBMLsimulator 1.2.1, July 2014

Operating system:	<input checked="" type="checkbox"/> Windows	<input checked="" type="checkbox"/> Mac OS	<input checked="" type="checkbox"/> Linux
Availability:	<input checked="" type="checkbox"/> Free for all	<input type="checkbox"/> Free for academic	<input type="checkbox"/> Commercial

17.42 SBMLSqueezer

SBMLSqueezer generates kinetic equations for biochemical networks according to the context of each reaction. It can be used as a plug-in for CellDesigner, in which case it uses the information contained in the SBGN representation of the network components. Additionally, it can also be used in a stand-alone mode, in which SBMLSqueezer evaluates the Systems Biology Ontology (SBO) annotations to extract the relevant information. Finally, an online version of SBMLSqueezer is available that runs without the need to install any software on the local machine. The rate laws that can be produced by SBMLSqueezer include several types of generalized mass action, detailed and generalized enzyme kinetics, various types of Hill equations, convenience kinetics, and additive models for gene regulation. User-defined settings specify which equation to apply for any type of reaction and how to ensure unit consistency of the model. Equations can be created using contextual menus. All newly created parameters are equipped with the derived unit and annotated with SBO terms (if available) and meaningful textual names. MathML is inserted directly into the SBML file. A LaTeX and text export of the SBML model via the integrated tool SBML2LaTeX (<http://www.cogsys.cs.uni-tuebingen.de/software/SBML2LaTeX>) is also provided. Finally, a 90-pages user manual is also available on the website.

URL: <http://www.cogsys.cs.uni-tuebingen.de/software/SBMLSqueezer/>

Latest version: SBMLSqueezer 2.1, August 2015

Operating system:	<input checked="" type="checkbox"/> Windows	<input checked="" type="checkbox"/> Mac OS	<input checked="" type="checkbox"/> Linux
Availability:	<input checked="" type="checkbox"/> Free for all	<input type="checkbox"/> Free for academic	<input type="checkbox"/> Commercial

17.43 SBML Toolbox

SBMLToolbox is built on top of libSBML and provides a set of basic functions allowing SBML models to be used in both Matlab and Octave. SBMLToolbox provides functions for creating and validating models and for manipulating and simulating these models using ordinary differential equation solvers. SBMLToolbox works by translating SBML models to/from a Matlab structure called MATLAB_SBML. It provides facilities for manipulating this and its substructures within the Matlab or the Octave environment. The libSBML binding enables the import and export of these structures to and from SBML files. The toolbox is not intended to be a complete Systems Biology toolbox for Matlab, but rather a platform facilitating getting SBML in and out of Matlab and serving as a starting point from which users can develop their own functionality. The current version of SBMLToolbox supports all releases of SBML up through Level 3 Version 1.

URL: <http://sbml.org/Software/SBMLToolbox>

Latest version: SBML Toolbox 4.1.0, January 2012

Operating system:	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
Availability:	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.45 SBML Validator

The SBML Validator is an online resource that checks the syntax and internal consistency of SBML code that is uploaded as file, submitted as URL, or pasted directly into the webpage. The validation procedure is built on the libSBML 5.10.1 library and supports SBML code up to Level 3 Version 1. The maximum allowed file size is 32 MByte, which should be sufficient for most purposes. Several validation options, such as checking the MathML syntax and the consistency of identifiers or performing a static analysis to test if the model is overdetermined, can be selected to fine-tune the validation process. If a model gives problems with SBML processing tools, it is definitely a good idea to run it through the validator.

URL: <http://sbml.org/Facilities/Validator>

Latest version: Support up to SBML Level 3 Version 1

Operating system:	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
Availability:	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.44 SBtoolbox2

The Systems Biology Toolbox 2 (SBToolbox2) for Matlab offers systems biologists a powerful, open, and user-extensible environment in which to build models of biological systems. The toolbox features a wide variety of specialized analysis tools and Matlab adds to that with a large number of built-in functions and a high-level scripting language, allowing the user to quickly and efficiently add new functionality. Models can be defined using a simple syntax based on either differential equations or biochemical reaction equations. In addition, the toolbox also supports the import and export of SBML models. Via the toolbox, models can be simulated deterministically or stochastically and can also be analyzed by a variety of methods to study steady states and their stability and perform metabolic control analysis, bifurcation analysis, parameter sensitivity analysis, or a stoichiometric analysis.

URL: www.sbttoolbox2.org/main.php?display=documentation&SBT&menu=overview

Latest version: SBToolbox2 1361, July 2014

17.46 SensA

Mathematical models (written in SBML) can contain many parameters and it is important to know how sensitive model properties (i.e., concentrations or fluxes) depend on specific parameter values. SensA is a Web-based application for this type of sensitivity analysis [4]. It is based on metabolic control analysis and computes local, global, and time-dependent properties of model components. Models can be uploaded in SBML format and results are represented graphically in diagrams and color-coded tables. Results can be stored online or downloaded as text and graphics. While other tools such as Copasi and JWS online also offer an MCA-based sensitivity analysis for the steady-state case, SensA includes the computation of time-dependent sensitivities.

URL: gofid.biologie.hu-berlin.de/

Latest version: Support up to SBML Level 2 Version 4 (with the exception of events, constraints, algebraic rules, time variable compartment sizes, and initial assignments).

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.47 SmartCell

SmartCell has been developed to be a general framework for modeling and simulating diffusion–reaction networks in a whole-cell context. It supports localization and diffusion by using a mesoscopic stochastic reaction model. The SmartCell package handles any cell geometry, considers different cell compartments, allows localization of species, and supports DNA transcription and translation, membrane diffusion, and multistep reactions, as well as cell growth. Entities are represented by their copy number and location. In order to introduce spatial information, the geometry is divided into smaller volume elements, called voxel, where stochastic events take place. The use of a mesh allows it to consider diffusion as translocation across adjacent volume sites. The user-defined model is translated into an internal core model, where rates are converted into reaction probabilities per unit time. At this stage, reversible processes, diffusion, and complex formation are converted into an equivalent set of unidirectional elementary processes. Finally, each process is translated into as many individual events as volume elements in the region where the process is defined. The core model is subsequently used by the simulation engine itself. The simulation engine then uses different algorithms in order to have exact stochastic result (NREM, NSVM, Hybrid), exact deterministic result (ODE), or approximate stochastic result (Tau leap).

URL: <http://software.crg.es/smartzell/>

Latest version: SmartCell 4.3b3, March 2011

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.48 STELLA

STELLA is a commercial computer modeling package with a graphical user interface that allows users to construct dynamic models that simulate biological systems. Stocks and flows can be placed on the GUI to model

continuous processes or discrete events using so-called queues and ovens. Models are saved in a proprietary format. Furthermore, sliders, switches, and buttons can be placed on the model canvas to manipulate model parameters.

URL: <http://www.iseesystems.com/softwares/Education/StellaSoftware.aspx>

Latest version: STELLA 10.0.4, October 2013

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input type="checkbox"/>
<i>Availability:</i>	Free for all <input type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input checked="" type="checkbox"/>

17.49 STEPS

STEPS (Stochastic Engine for Pathway Simulation) is a package for exact stochastic simulation of reaction–diffusion systems in arbitrarily complex 3D geometries. The core simulation algorithm is an implementation of Gillespie’s SSA, extended to deal with diffusion of molecules over the elements of a 3D tetrahedral mesh. Since version 2.0, STEPS also supports accurate and efficient computation of local membrane potentials on tetrahedral meshes, with the addition of voltage-gated channels and currents. Tight integration between the reaction–diffusion calculations and the tetrahedral mesh potentials allows detailed coupling between molecular activity and local electrical excitability. The tool has been implemented as a set of Python modules, which means STEPS can be used via Python scripts to control all aspects of constructing the model, generating a mesh, controlling the simulation, and generating as well as analyzing output. The core computational routines are implemented as C/C++ extension modules for maximal speed.

URL: <http://steps.sourceforge.net/STEPS/default.php>

Latest version: STEPS 2.2.0, April 2014

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.50 StochKit2

StochKit2 is an efficient, extensible stochastic simulation framework developed in the C++ language. StochKit2 provides implementations of several (exact) stochastic

algorithms, including the direct method, optimized direct method, and composition-rejection method. These methods generate exact trajectories from the chemical master equation, but use modified underlying algorithms and data structures to achieve different performance and scaling properties. The direct method, which uses simple data structures, tends to be best for relatively small models, while for very large models, the composition-rejection method is the most efficient. The StochKit2 user interface is simple. When a user chooses to run an exact stochastic simulation, the software immediately analyzes the model and automatically chooses the appropriate algorithm. StochKit2 also provides an interface for running stochastic simulations using an adaptive tau-leaping method. The tau-leaping method sacrifices exactness in exchange for taking larger time steps. The software uses an XML-based file format that is similar to, but not identical with, SBML. However, a function is provided to convert StochKit models to and from SBML.

URL: <http://www.engineering.ucsb.edu/~cse/StochKit/>
Latest version: StochKit 2.0.11, February 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.51 SystemModeler

SystemModeler is a commercial product of Wolfram Research, the same company that produces Mathematica. The software provides a graphical user interface for creating models based on mechanical, electrical, or magnetic interactions. With the help of the BioChem library, it is also possible to create kinetic models of biochemical reactions. Reaction networks can be created in a drag-and-drop fashion as with CellDesigner and then solved to obtain time course solutions of the model. If desired, the models can be built in a SBML compliant way and exported as an SBML file. The concept of SystemModeler is quite similar to SimuLink for Matlab, but the software can also be used independent of Mathematica. But if Mathematica is installed, the package Wolfram SystemModeler Link (WSMLink) is automatically installed. This enables a tight connection between Mathematica and SystemModeler and allows controlling and analyzing a SystemModeler model from within Mathematica.

URL: <http://www.wolfram.com/system-modeler/>
Latest version: SystemModeler 4.2, December 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input checked="" type="checkbox"/>

17.52 Systems Biology Workbench

The Systems Biology Workbench (SBW) [5] is a software system that enables different tools to communicate with each other via a fast binary encoded message system. Thus, SBW-enabled tools can use services provided by other modules and in turn advertise their own specialized services. At the center of the system is the SBW broker that receives messages from one module and relays them to other modules. JDesigner and Jarnac are two modules that come with the SBW standard installation. JDesigner is a program for the graphical creation of reaction networks, and Jarnac is a tool for the numerical simulation of such networks (time course and steady state). Jarnac runs in the background and advertises its services to the SBW broker. JDesigner contacts the broker to find out which services are available and displays the found services in a special pull-down menu called SBW. A reaction model that has been created in JDesigner can now be sent to the simulation service of Jarnac. A dialog box opens to enter the necessary details for the simulation and then the broker calls the simulation service of Jarnac. After a time course simulation finishes, the result is transmitted back to JDesigner (via the broker) and can be displayed. The list of SBW-enabled programs contains programs specialized in the graphical creation of reaction networks (JDesigner and CellDesigner), simulation tools (Jarnac and TauLeapService), analysis and optimization tools (Metatool, Bifurcation, and Optimization), and utilities such as the Inspector module, which provides information about other modules.

URL: <http://sbw.kgi.edu/research/sbwintro.htm>
Latest version: SBW 2.9, February 2012

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.53 Taverna

Taverna is an open-source and domain-independent Workflow Management System similar to KNIME. Thus, it provides a suite of tools used to design and execute

scientific workflows around the analysis of data. The Taverna suite is written in Java and includes the Taverna Engine (used for enacting workflows) that powers both the Taverna Workbench (the desktop client application) and the Taverna Server, which allows remote execution of the created workflows. Taverna is also available as a Command Line Tool for a quick execution of workflows from a terminal. Finally, Taverna Online lets you edit and run Taverna workflows on the Web. Workflows are created via drag-and-drop method and the resulting workflow diagrams are then displayed using the external tool Graphviz.

URL: www.taverna.org.uk/

Latest version: Taverna 2.5, May 2014

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.54 VANTED

VANTED stands for *V*sualization and *A*nalysis of *N*etworks containing *E*xperimental *D*ata. This system makes it possible to load and edit graphs, which may represent biological pathways or functional hierarchies. It is possible to map experimental data sets onto the graph elements and visualize time series data or data of different genotypes or environmental conditions in the context of underlying biological processes. Built-in statistic functions allow a fast evaluation of the data (e.g., *t*-test or correlation analysis). The latest version also contains support for models in the SBML format.

URL: immersive-analytics.infotech.monash.edu/vanted/

Latest version: VANTED 2.6, November 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.55 Virtual Cell (VCell)

Users can build models with a Java interface to specify compartmental topology and geometry, biochemical reactions, and relevant interaction parameters. VCell can handle compartmental and spatial models, which can be simulated deterministically or stochastically. Models are normally stored in VCML (Virtual Cell Markup Language), but deterministic nonspatial (and some spatial)

models can also be imported and exported as SBML. Models are stored in a personal online repository, but they can also be saved on the local computer. Virtual Cell automatically converts the biological description into a corresponding mathematical system of ordinary (compartment model) and/or partial differential equations (2D and 3D spatial models). Mathematically experienced users may directly specify the complete mathematical description of the model, bypassing the schematic interface. Normally the equations are then sent to online servers of the University of Connecticut, where the model is simulated by appropriate solvers and the results are then transferred back for display. But if desired, the model can also be run at the local machine. Finally, VCell can also perform parameter estimations for which it internally uses the Copasi engine (for nonspatial deterministic models). Results can be displayed and analyzed online or downloaded to the user's computer in a variety of formats. A more in-depth description of VCell is given in Chapter 5.

URL: www.vcell.org/

Latest version: VCell 5.3, June 2015

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.56 xCellerator

xCellerator is a Mathematica® package designed to aid biological modeling via the automated conversion of chemical reactions into ODEs. These equations can then be solved via numerical integration and are displayed as time course diagrams. SBML is also supported via plugins and MathSBML. The current version generates some warnings with Mathematica version 9, but nevertheless works fine.

URL: www.cellerator.org

Latest version: xCellerator 0.91, November 2012

<i>Operating system:</i>	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
<i>Availability:</i>	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

17.57 XPPAUT

XPPAUT is a tool for solving differential equations, difference equations, delay equations, functional equations,

boundary value problems, and stochastic equations. XPPAUT contains the code for the popular bifurcation program AUTO. Thus, you can switch back and forth between XPPAUT and AUTO, using the values of one program in the other and vice versa.

URL: <http://www.math.pitt.edu/~bard/xpp/xpp.html>

Latest version: XPPAUT 7, December 2012

Operating system:	Windows <input checked="" type="checkbox"/>	Mac OS <input checked="" type="checkbox"/>	Linux <input checked="" type="checkbox"/>
Availability:	Free for all <input checked="" type="checkbox"/>	Free for academic <input type="checkbox"/>	Commercial: <input type="checkbox"/>

Exercises

- 1) When and why should a system be modeled stochastically instead of deterministically?
- 2) Which development in the last years is important for the exchange of models between different simulation tools?
- 3) What is the purpose of libSBML?
- 4) 17.4 Is it possible to develop models in (a) Mathematica or (b) Matlab that support SBML?
- 5) Use CellDesigner to model the irreversible reaction $S \rightarrow P$ using a Michaelis–Menten kinetics. Draw the diagram, specify the kinetics (for $K_m = 2 \text{ mmol}^{-1}$, $V_{\max} = 5 \text{ mmol (l*s)}^{-1}$, $S_{t=0} = 100$ molecules, and $P_{t=0} = 0$ molecules) and run a time course simulation.
- 6) Export the model as SBML and import it into Copasi. Run a time course simulation to see if it is identical to the one from CellDesigner.
- 7) Use the following three time/substrate concentration data points for model fitting: $P_1: 5 \text{ s}60 \text{ mmol}^{-1}$, $P_2: 10 \text{ s } 50 \text{ mmol}^{-1}$, $P_3: 15 \text{ s } 20 \text{ mmol}^{-1}$. What are the values of K_m and V_{\max} after fitting?
- 8) Import the CellDesigner SBML model into Virtual Cell and run a stochastic simulation. Do you see any differences to the deterministic solution?

References

- 1 Klipp, E., Liebermeister, W., Helbig, A., Kowald, A., and Schaber, J. (2007) Systems biology standards: the community speaks. *Nat. Biotechnol.*, 25 (4), 390–391.
- 2 Ghosh, S., Matsuoka, Y., Asai, Y., Hsin, K.Y., and Kitano, H. (2011) Software for systems biology: from tools to integrated platforms. *Nat. Rev. Genet.*, 12 (12), 821–832.
- 3 Garny, A. and Hunter, P.J. (2015) OpenCOR: a modular and interoperable approach to computational biology. *Front. Physiol.*, 6, 26.
- 4 Floettmann, M., Uhendorf, J., Scharp, T., Klipp, E., and Spiesser, T.W. (2014) SensA: web-based sensitivity analysis of SBML models. *Bioinformatics*, 30 (19), 2830–2831.
- 5 Hucka, M., Finney, A., Sauro, H.M., Bolouri, H., Doyle, J., and Kitano, H. (2002) The ERATO Systems Biology Workbench: enabling interaction and exchange between software tools for computational biology. *Pac. Symp. Biocomput.*, 450–461.

Index

a

action functional 244
activated complex 41
activation energy 41
activator–inhibitor model
– linear stability analysis 140, 141
actuarial aging rate 315
adaptation and exploration strategies 229
– fold-change detection 230
– metabolic shifts and anticipation 233
– sensing and random switching 231
– shannon information 232
– signaling pathways, information transmission in 230
– value of information 232
adaptation motif 157, 158
Advanced Search 451
affine transformations, of probability density 395
agarose gel electrophoresis of DNA restriction fragments 359
agent-based models 17, 133
agent-based systems 63
aging process 11, 314
– actuarial aging rate 315
– defined 314
– delay differential equations 323
– environmental risk 315
– evolution of 316
– intrinsic vulnerability 315
– mechanistic theories 315
– – graphical representation 315
– stochastic simulations 318
aging rate 315
alcohol dehydrogenase 452
algebraic equations 30
algebro-differential equation system 30
allosteric feedback 220
allosteric regulation 49
AmiGO 449
amino acids 27, 76, 289, 343, 344, 375, 450
anaphase-promoting complex (APC) 309
animal cell 347
ANOVA test 17, 402, 403
antagonistic pleiotropy theory 317
APC. *see* anaphase-promoting complex (APC)

archetypes 246

artificial selection 262
artificial siRNAs 371
autonomous agents 17

b

backup genes 219
backup pathways 219
BACs. *see* bacterial artificial chromosomes (BACs)
bacterial artificial chromosomes (BACs) 360
bacterial operons 160
bacterial promoter sequences 146
bacteriophages 358
balance equations 24, 25
Bayesian conditioning 94
Bayesian estimation 93
Bayesian model selection 103, 104
Bayesian networks 93
– for gene expression 145
Bayesian parameter estimation 92
Bayesian statistics 88
Bernoulli experiments 395
bifurcation analysis 210
binding constants 46, 48
binding of ligands to proteins 46
Binomial distribution 394, 395
biochemical models 87
– simplification of 105
biochemical reaction systems 40, 41, 127
– chemical master equation 128, 129
– deterministic kinetic mode 128
– networks 4
– – structure of 146
– Poisson distribution, with average value and variance 128
biological environments 23
biological function 165
biological hypotheses 4
biological membranes
– structure and function of 347, 348
biological modules 160
biological molecules 334
– forces important in 336–338
– major classes of 338
biological networks 145, 146

- biological robustness properties 218
- biological systems 3
- biological thermodynamics 417
- biology 333
- biomarkers 9
- BioModels Database 81
 - annotations in systems biology models in 81
- BioNetGen language (BNGL) 459
- BioNumbers 76, 445
- BioPAX 72, 74, 78
- biophysics 15
- blotting techniques 362
- Boltzmann distribution 418, 421
- Boltzmann picture 418
- Boltzmann's gas constant 49
- Boolean models 63
 - of gene regulation 6
- Boolean networks 16, 122
 - advanced types of 123, 124
 - basic principles of 122
 - dynamics of, characterization 122
 - model 121
- Boolean rules (truth table) for systems
 - with one input 122
 - with two inputs 122
- Boolean transitions 122
- bootstrapping 88, 91
 - cross-validation 91, 92
 - used for estimating a mean value 91
- Box, George 5
- box plot 397
- Brownian motion, as a random process 407
- budding yeast cell cycle 311
 - schematic representation 312
 - self-oscillating network 313
- c**
- Caenorhabditis elegans* 10, 451
- calcium homeostasis 319
- canalization 218
- canonical ensemble 418, 421
- carbohydrates 339, 340
- Cartesian products 395
- β -catenin 302
- causal interactions 115, 116
- ^{13}C -based metabolic flux analysis 458
- CDKs. *see* cyclin-dependent kinases (CDKs)
- CDS. *see* coding sequences (CDS)
- cell cycle 307
 - budding yeast models 311
 - mitotic oscillator, minimal cascade model of 310
 - steps 309
- CellDesigner 66, 71, 472
- CellML 78
- cell populations, random switching 231
- cellular
 - composition 333
 - differentiation 19
 - machinery 3
 - organization 3
 - pathways 9
 - reprogramming with viral vectors 21
 - transcription networks 8
- central limit theorem 396
- centrifugation 364
- Chapman–Kolmogorov equation 407
- ChEBI database 79
- chemical bonds 336
- chemical equilibrium states 50
- chemical Langevin equation 130, 131
- chemical noise 130, 131
- chemical potentials 32, 422
 - difference 31
- chemical reaction systems 422
 - temperature and pressure as free variables 422
- chemotaxis 234
 - model 223
- ChIP. *see* chromatin immunoprecipitation (ChIP)
- ChIP-on-Chip method 372, 373
 - limitation 373
- ChIP-PET technique 372
- chromatin immunoprecipitation (ChIP) 153, 372
- CKI. *see* cyclin-dependent kinase inhibitors (CKI)
- classification methods 438, 439
 - boosting algorithms based on 441
 - clustering-based classification 441
 - k -nearest neighbor method 440
 - practical problems underlying classification of patients to 439
 - support vector machines (SVMs) 439, 440
 - unsupervised/supervised methods 439
- cloning vectors 359
- closed-loop linear control system 417
- clustered regular interspaced palindromic repeats (CRISPR) 371
- clustering algorithms 430
- clustering coefficient 149
- cluster validation 435
 - average compactness- (isolation-) values 435
 - external validation measures 435
 - internal validation measures 435
 - Silhouette index 435
 - visualization of cluster quality 436
- coding sequences (CDS) 448
- coefficient of relatedness 277
- coefficient of variation 397
- coefficients of control analysis 51
- column chromatography 364, 365
- COMBINE initiative 78
- combining rate laws, into models 113
- compartment models 30, 135, 136
- competitive exclusion principle 276
- competitive inhibition 45
- complementary DNA (cDNA) 361, 446

complex ODE systems 64
 compound/drug databases 452
 – ChEBI 453
 – Guide to PHARMACOLOGY 453
 comprehensive R archive network (CRAN) 468
 computational accuracy 64
 computational modeling 4, 5, 15, 16, 78
 – advantages of 5, 6
 – basic notions for 6
 computational systems biology 63
 computer-assisted modeling. *see* computational modeling
 computer chips 164
 computer simulations 4
 concentration control coefficient 52
 concentration response coefficient 52
 conditional probability 392
 connectivity 149, 150
 ConsensusPathDB 76, 467
 conservation relations 29, 30
 constant organization 264
 constants 6
 constitutive transcription 127
 constraint-based flux balance analysis 32
 constraint-based flux optimization 23, 30
 constraint-based methods 32
 – assumption of optimality 33
 continuous model 7
 continuous random processes 411
 – Fokker–Planck equation 411, 412
 – Langevin equations 411
 continuous time axis 16
 continuous values 16
 control coefficients 51–53
 convenience kinetics 49
 convolution rule, of normal distribution 395
 cooperative behavior 276
 – group selection 277
 – kin selection 277
 – reciprocity 277
 – spatial structure role 277
 cooperativity 47
 COPASI 66–69, 78
 – CopasiSE version 68
 – screenshot of COPASI’s GUI 69
 correlation analysis 9
 correlation of samples 398
 correlation plots and performance of correlation measures 399
 coupled systems 110
 – emergent behavior in 114, 115
 – modeling of 111
 – coupling of submodels 111, 112
 – hierarchical regulation analysis 112
 – modeling the system boundary 111
 – supply–demand analysis 112
 covariance matrix 394
 COX. *see* cytochrome-c oxidase (COX)

CRAN. *see* comprehensive R archive network (CRAN)
 CRISPR. *see* clustered regular interspaced palindromic repeats (CRISPR)
 CRISPR/Cas technique 371
 CyberCell Database 10
 cyclin-dependent kinase inhibitors (CKI) 309
 cyclin-dependent kinases (CDKs) 309
 cytochrome-c oxidase (COX) 319
 cytosol 349

d

DAG. *see* directed acyclic graph (DAG)
 1D and 2D protein gels 361, 362
 databases 445
 – compound/drug (*see* compound/drug databases)
 – enzyme reaction kinetics (*see* enzyme reaction kinetics databases)
 – general-purpose data resources 445
 – BioNumbers 446
 – PathGuide 445, 446
 – microarray/sequencing (*see* microarray/sequencing databases)
 – model collections
 – BioModels 452
 – JWS Online 452
 – National Center for Biotechnology Information 446
 – nucleotide sequence databases 446
 – EMBL Nucleotide Database 447
 – Ensembl 447
 – Entrez 447
 – European Nucleotide Archive (ENA) 447
 – GenBank/RefSeq/UniGene 446
 – ontology (*see* ontology databases)
 – pathway (*see* Pathway databases)
 – protein (*see* protein databases)
 – of protein modifications (RESID) 447
 – transcription factor (*see* transcription factor databases)
 data formats 63, 78, 457
 data for thermodynamic calculations 424
 data integration 8, 9, 72, 461
 data normalization 9
 data resources 75
 – general-purpose 445, 446
 defective mitochondria 316
 degree of heteroplasmy 319
 delay differential equations 323
 densities 393
 – estimation 397
 density function 202, 393, 395, 398, 415, 427
 deoxynucleotide triphosphates (dNTPs) 307
 deoxyribonucleic acid (DNA) 345
 dependence scheme
 – for model parameters 95
 – for rate constants and metabolic state 95
 dephosphorylation 292
 descriptive statistics 396
 deterministic models 7, 133

deterministic replicator equation 274
 DICER pathway for maturation 372
Dictyostelium discoideum 138
 differential equation
 – system 30
 differential equations 63
 differential evolution (DE) 462
 diffusion equation, solutions of 136, 137
 – cosine profile 137
 – Gaussian profile 137
 – stationary profile 137
 DIGE (difference gel electrophoresis) 362
 dimeric protein 47
 direct binding modular rate law 50
 directed acyclic graph (DAG) 449
 direct fitness advantage 229
 direct method 65
 discrete models 121, 122
 discrete random walk 406
 discrete time steps 16
 discrete value 16
 disease-relevant data 9
 disposable soma theory 317
 distribution functions 393
 DNA chips 367
 DNA Database of Japan (DDBJ) 447
 DNA libraries 359, 360, 361
 DNA ligase 359
 DNA methylation 19
 DNA microarrays 357
 DNA microinjection 370
 DNA polymerases 308, 309
 DNA–protein interactions 16
 DNA replication 308
 DNA sequences 3, 4
 DNA synthesis 315
 dNTPs. *see* deoxynucleotide triphosphates (dNTPs)
 double-strand break (DSB) 370
Drosophila melanogaster 138, 372
 3D structure of a protein 5
 dynamical behavior 15
 dynamical system 6
 dynamic behavior of feed-forward loops (FFLs) 159
 dynamic behavior of network 4
 dynamic equilibrium 15
 dynamic FBA 34
 dynamic fluctuations 132, 133
 dynamic model of feed-forward loops 158
 dynamic networks 16
 dynamic systems 161, 162, 386
 – describing with ordinary differential equation 386
 – notations 386
 – global stability of steady states 390
 – limit cycles 390, 391
 – linearization of autonomous systems 388
 – solution of linear ODE systems 388
 – stability of steady states 388, 389

e
 Eadie–Hofstee graphical representation 44
 EBI Ontology Lookup Service (OLS) 79
 ECTree browser 451
 elasticities 48, 49, 112, 115, 209, 213, 463
 elasticity coefficients 49, 51, 52
 elasticity sampling 213
 – under thermodynamic constraints 213
 electrochemical potentials 423
 electrophoresis 358
 elementary flux modes 27, 29
 embryonic development, robust pattern formation in 138
 – bicoid gradient in fly embryo 138, 139
 embryonic stem cells (ES cells) 370
 empirical distribution function 398
 endergonic reactions 41
 endoplasmic reticulum 350
 energy balance analysis 32
 Ensembl ContigView 447
 Ensembl Genome Browser 12
 enthalpy 32, 41
 entropy 189, 230, 247, 419, 420, 422, 426
 environmental risk 315
 enzymatic rate constants
 – from the Brenda database, distribution of 94
 – distributions of 94
 enzymatic reactions 8
 enzyme activity by effectors, regulation of 44
 enzyme-catalyzed reactions 4, 145, 422
 enzyme investments 250
 enzyme kinetics 43
 – parameters 94
 – standard 43
 enzyme mechanisms 42
 enzyme reaction kinetics databases
 – BRENDa 451, 452
 – SABIO-RK 452
 enzyme–substrate complex 42, 45
 epigenetic regulation 20
 epistasis 163
 – epistatic interactions 164
 The Epsilon Group, (TEG) 468
 equilibrium constant 421, 423
 – and energies 421
 equilibrium thermodynamics, in reaction systems 42
 Erdős–Rényi random graphs 147, 148
 error distribution 90
 ESS. *see* evolution, stable strategies (ESS)
 estimators 89
 ESTs. *see* expressed sequence tags (ESTs)
 eukaryotes 335
 eukaryotic cells 350
 Euler–Lotka equation 317
 European Bioinformatics Institute (EMBL-EBI) 446
 European Nucleotide Archive (ENA) 447
 evaluating system of ODEs 64
 evolution 241, 261

- of analogous traits 164
- biological macromolecules, selection equations for 263
- control of 243
- cost 247
- effort 248
- evolution strategies (ES) 462
- hypercycle model 267
- of modularity 164
- neutral theory of molecular evolution 270
- optimization 263
- Quasispecies model 265
- as search strategy 242
- spin glass model 269
- stable strategies (ESS) 276
- evolutionary game theory 271
 - cooperative behavior 276
 - evolutionarily stable strategies 275
 - game theory 273
 - metabolic yield and efficiency, compromises between 278
 - population dynamics, replicator equation for 274
 - rock–scissors–paper game, dynamical behavior 276
 - social interactions 272
- evolvability 229
- experimental techniques 357
- exploration strategies 234
 - chemotaxis 234
 - infotaxis 235
 - stress-induced mutagenesis 234
- exponential distribution 394
- expressed sequence tags (ESTs) 446
- expression of genes 351
- eXtensible Markup Language 72
- external metabolites 24
- extreme pathways 27

- f**
- FACS. *see* fluorescence-activated cell sorting (FACS)
- failure tolerance 218
- false discovery rate (FDR) 429
- fatty acids 341
- FBA. *see* flux balance analysis (FBA)
- FDR. *see* false discovery rate (FDR)
- feedback cycles 304
- feedback regulation 157
- feed-forward loop (FFL) 150, 158
 - functions of 159
- FFL. *see* feed-forward loop (FFL)
- fixed average energy 421
- fixed points 7
- fluorescence-activated cell sorting (FACS) 209
- flux balance analysis (FBA) 6, 8, 23, 29, 30
 - extensions of 33
 - minimization of metabolic adjustments 33
 - geometric interpretation of 31
- flux cone 27
- flux control 425
 - coefficient 52
- flux distributions 163
- flux–force relation, consequences of 425
- flux modes 27, 28, 162
- flux optimization 250
 - paradigm 32
 - applications and tests of 32
- flux ratio 104, 424, 425
- flux response coefficient 52
- flux sampling 212
- flux variability analysis 34
- FlyBase 449
- force-dependent modular rate law 50
- formats 72. *see also* data formats
- fourth-order Runge–Kutta algorithms 64
- FOXP1 gene 12
- free energy 32, 41, 189, 419, 422
- free energy differences
 - biochemical reactions 42
- FreeFem++ 467
- frequency response function 414
- functional groups, in biological molecules 338
- fundamental cellular structures 334

- g**
- GA. *see* genetic algorithms (GA)
- game theory 273
 - hawk–dove game 273
 - Nash equilibrium 274
 - payoff matrix 273
 - prisoner’s dilemma 273
 - repeated games 274
- Gauss algorithm 26, 29
- Gaussian distribution 95, 398
- Gaussian elimination algorithm 383
- Gaussian probability density 90
- Gauss–Markov random processes 415
- GEF. *see* guanine nucleotide exchange factor (GEF)
- gel electrophoresis 358
- gene cascades, temporal fluctuations 202
 - linear model with two genes 202
 - time correlations in protein levels, measurement 203
- gene-enrichment scores 437
- gene exchange and reuse 162
- gene expression 121, 171
 - dynamic models of 180
 - gene expression and regulation, basic model of 180
 - natural and synthetic gene regulatory networks 183
 - with Stochastic equations 186
 - in eukaryotic cells 352
 - fluctuations 196
 - gene cascades, temporal fluctuations 202
 - intrinsic and extrinsic variability 200
 - stochastic model of transcription and translation 197
 - functions 187
 - from equilibrium binding 188
 - inferring transcription factor activities 192

- lac Operon in *E. coli* 187
 - of lac promoter 191
 - mRNA and protein levels, correspondences between 196
 - network component analysis 194
 - promoter occupancy, thermodynamic models of 189
 - mechanisms 171
 - general promoter structure 173
 - microRNAs, post transcriptional regulation through 176
 - promoter elements, prediction and analysis of 174
 - omnibus (GEO) 447
 - regulation 171
 - regulation of 355
 - transcription factor-initiated 171
 - gene functions 160
 - gene loss 219
 - gene network coordinating 160
 - gene ontology (GO) 9, 79, 448
 - GeneOntology Consortium 436
 - general-purpose databases 75
 - gene regulation 15, 33, 93, 183, 187, 190, 194, 419, 469
 - genes code 3, 12, 35, 183
 - genetic algorithms (GA) 462
 - genetically modified mouse, serve as a model 5
 - genetic network fluctuations 199
 - genetics 334
 - code 353
 - programming 263
 - genetic sequence database (GenBank) 446
 - genetic tug-of-war (gTOW) method 252
 - genome editing 370
 - genome-scale networks 23
 - genome sizes of organisms 335
 - GEO. *see* gene expression, omnibus (GEO)
 - geometric mean 397
 - geometric random graphs 148
 - GFP gene 375
 - Gibbs free energies 31, 41, 422, 423, 424
 - change of 41
 - Gibbs picture 418
 - global model reduction 108–110
 - linearization of biochemical models 108, 109
 - linear relaxation modes 109
 - glucose-6-phosphate (G6P) 18
 - glycogen synthase kinase 3 (GSK3) 301
 - Goldbeter's minimal model 312
 - golgi complex 350
 - Gompertz–Makeham equation 314, 317
 - good models
 - defined 99
 - possible requirements for 99
 - GPCR. *see* G proteincoupled receptors (GPCR)
 - G proteincoupled receptors (GPCR) 296
 - G protein cycles 295
 - Gramian matrices 415, 416
 - graphical user interface (GUI) 464
 - green fluorescent protein 374, 375
 - group selection 277
 - growth and reproduction 333
 - GSK3. *see* glycogen synthase kinase 3 (GSK3)
 - gTOW. *see* genetic tug-of-war (gTOW) method
 - GTPase activity 297
 - guanine nucleotide exchange factor (GEF) 297
 - GUI. *see* graphical user interface (GUI)
- h***
- Haldane relationships 44, 95, 423
 - halobacteria 335
 - Hamming distance 123, 264
 - Hanes–Woolf graphical representation 44
 - Hankel singular values 416
 - Hawk–Dove game 273
 - Hayflick limit 316
 - Hessian matrices 254
 - heterogeneities 357
 - heterogeneous data sets 72
 - hierarchical clustering 431–433
 - parameters in 432
 - hierarchical methods 430
 - high osmolarity glycerol (HOG) 297
 - high-performance liquid chromatography (HPLC). 365
 - high-throughput methods 357, 358
 - sequencing method 373
 - high-yield fluxes 278
 - Hill equation 47
 - Hill function 20
 - HOG. *see* high osmolarity glycerol (HOG)
 - Holm's stepwise correction 429
 - Hox genes 12
 - Human Genome Project 8
 - human mortality 314
 - hybridization techniques 362
- i***
- identity matrix 29
 - IEF. *see* isoelectric focusing (IEF)
 - implicit methods 64
 - impulse input 414
 - impulse response function 414
 - induced pluripotent stem cells (iPS cells) 20
 - inequality constraints 246
 - inferring transcription factor 192
 - information encoded in stoichiometric matrix N 25
 - infotaxis 235
 - input-output relations 414
 - of signaling systems 153
 - in situ* hybridization 364
 - integral feedback 221
 - integrated metabolic, and regulatory network 35
 - intelligent database systems 9
 - Internet 10
 - interquartile range 397
 - intrinsic and extrinsic variability 200
 - calculation of 200

- measurement of 200
- intrinsic vulnerability 315
- invariant distribution 410
- inverse problems 91
- isoelectric focusing (IEF) 361

j

- Java simulations 464
- JDesigner 472
- Jensen's inequality 394
- JMadonna 459
- joint probability density 395
- JSim's model 463
- jump processes in continuous time 410
 - deriving master equation 410, 411

k

- Karhunen–Loève transform 416
- KEGG. *see* Kyoto encyclopedia of genes and genomes pathway (KEGG)
- kernel 26
 - matrix 26
- kinases transmit signals, by phosphorylating 82
- kinetic constants 41, 43, 52, 94
- kinetic modeling 7, 8, 39, 121, 421
 - of enzymatic reactions 39
- kinetic rate equation models 6
- Kinetic Simulation Algorithm Ontology (KiSAO) 78
- kinetics of a simple decay 40
- kin selection 277
- K-means algorithms 434, 435
- Knockout Mouse Project (KOMP) 12
- knockout mutations 23, 123
- Kyoto encyclopedia of genes and genomes pathway (KEGG) 75, 286, 450
 - database 76, 437

l

- lac Operon, *E. coli* 187
- lac permease activity 245
- lac promoter 191
- Lagrange multipliers 246
- Lambda 360
- large numbers, strong law 396
- law of mass action 40
- least-squares estimator 403
- least-squares method 89, 403
- Legendre transformation 422
- length and time scales in biology 4
- length scales 3
- libAntimony 458
- ligases 357
- likelihood function 89
- likelihood ratio test
- limit theorems 395
- linear algebra 381
- linear degradation 127

- linear dynamical systems 413
 - control of 412
- linear equations 49, 381–383
 - system 26
- linear filters 414
- linearization 44
- linear models 401, 402
- linear regression 17, 88, 89
- linear systems, systematic solution of 383, 384
- Lineweaver–Burk graphical representation 44
- link matrix 29
- lin-log kinetics 49
- lipids 340
- local sensitivity analysis 210
- log-normal distribution, density function 395
- low-yield fluxes 278
- lysosomes 351

m

- macromolecules 127
- macroscopic
 - behavior 127
 - model 133
 - kinetic 197
- Manhattan distance 430
- MAP kinase cascades 296
- MAPKs. *see* mitogen-activated protein kinases (MAPKs)
- marginal density 395
- Markov chains 410
- Markov processes 127, 409
- mass action kinetics 48, 49
- mass action law 42
- mass spectrometry (MS) 357, 369, 370
- master quasispecies distribution 266
- master sequence 265
- master species 265
- Mathematica 465
- mathematical description of biological systems 15
- mathematical graphs 147
- mathematical modeling 4, 8
 - of a biological system 16
- mathematical modeling language (MML) 463
- mathematical random processes 406
- mathematical robustness criteria 218
- Matlab 465, 470
- matrices 384
 - basic matrix operations 384, 385
 - basic notions 384
 - dimension and rank 385, 386
 - eigenvalues and eigenvectors of a square matrix 386
 - linear dependency 384
- matrix expressions for control coefficients 55–57
- matrix representation of coefficients 53
- maximal entropy, principle of 421
- maximum likelihood 89
 - estimation 92
- MCA. *see* metabolic control analysis (MCA)

- mean 396
- median 396
- median absolute deviation 397
- message parsing interface (MPI) protocol 469
- metabolic capacity 23
- metabolic control analysis (MCA) 50, 51, 461
- metabolic control theory, theorems of 53
 - connectivity theorems 54, 55
 - summation theorems 54
- metabolic efficiency 278
- metabolic maps 27
- metabolic modeling 19, 48
- metabolic networks 23, 24, 146, 150
 - *Escherichia coli* 146
- metabolic or regulatory network model
 - basic elements 23
- metabolic pathways 162
- represented by graphs 150
- metabolic shifts, and anticipation 233
- adaptation, indirect cues based 234
- metabolic shifts 233
- transient state, management of 233
- metabolic systems 285
- metabolic modeling, basic elements 286
- threonine synthesis pathway model 289
- upper glycolysis, toy model 286
- metabolic yield 278
- metabolism 163
- metabolites 4
- metabolite–transcript correlations 9
- Metropolis–Hastings algorithm 93
- Michaelis constants 43, 50, 94
- Michaelis–Menten equation 44
 - linearization 44
 - parameter estimation 44
 - for reversible reactions 44
- Michaelis–Menten kinetics 5, 42, 43, 311
 - different approaches for linearization of 44
 - general scheme of inhibition 45
 - types of inhibition for irreversible and reversible 46
- Michaelis–Mentenlike rate laws 95
- microarray 454
 - experiment 72
- microarray/sequencing databases
 - ArrayExpress 454, 455
 - Gene Expression Omnibus 454
- microcanonical ensemble 421
- microinjection 370
- microscopic stochastic model 198
- microstate 417
 - ensembles of 418
- microtubules 64
- minimal cascade model 310
- minimal fluxes, principle of 250
- minimal information, principle of 211
- minimization of metabolic adjustments (MoMA) 33
 - vs. FBA 33
- minimum information about a microarray experiment (MIAME) 72, 454
- minimum information about a proteomics experiment (MIAPE) 72
- minimum information about a simulation experiment (MIASE) 9
- minimum information about sequencing experiment (MINSEQE) 454
- Minimum Information for Biological and Biomedical Investigations (MIBBI) Consortium 78
- minimum information requested in the annotation of biochemical models (MIRIAM) 9
 - MIRIAM Registry 79
- missing values 430
- mitochondria 350
- mitochondrial damage study 318
 - delay differential equations 323
 - stochastic simulations 318
- mitochondrial DNA (mtDNA) 375
- mitogen-activated protein kinases (MAPKs) 298
 - cascades 82
 - mitophagy 323
- mitotic oscillator 310
- mixed inhibition 46
- MML. *see* mathematical modeling language (MML)
- model databases 77
 - BioModels 77
 - JWS Online 78
- modeling approaches, for biochemical systems 15–17
- modeling framework 16
- model organisms 9
 - *Caenorhabditis elegans* 11
 - *Drosophila melanogaster* 11, 12
 - *Escherichia coli* 9–11
 - *Mus musculus* 12
 - *Saccharomyces cerevisiae* 11
- models 5
 - adequateness 5
 - alignment 79
 - assignment 7
 - behavior 7
 - classification 7
 - combination 80–82
 - comparison 78
 - concepts 15
 - merging 83
 - parameterization 49
 - predictions 17, 88, 90, 91, 100, 103, 104, 372
 - purpose 5
 - reduction 104, 416
 - scope 6
 - selection 98
 - semantics 78
 - similarities 79
 - simplification 104
 - statements 6
 - of upper part of glycolysis 29

- validity 82, 83
- model selection, problem of 99
 - likelihood and overfitting 100, 101
 - methods for model selection 101
 - problem of overfitting 101
 - cross-validation 101
 - selection criteria 101
 - statistical tests 101
 - tests with artificial data 102
- modularity 160–163, 165
 - and biological function as conceptual abstractions 165
 - on levels of structure, dynamics, regulation, and genetics 161, 162
 - on various levels, exemplified by bacterial operons 162
- modular rate laws 49
- modular response analysis 113, 114
- molecular biology 3, 333, 334
 - of cell 336
- molecular dynamics 417
- molecule interactions 145
- moment-generating functions 412
- Monod model 48
- Monod–Wyman–Changeux model 48
- Mouse Atlas Project 12
- mouse genome database (MGD) 449
- mRNA 3
 - processing 353
- MS. *see* mass spectrometry (MS)
- multiple linear regression 403
- multiple testing 428, 429
- multivariate Gaussian distribution for logarithmic parameters 95
- multivariate statistics 9, 426
- mutation accumulation theory 317
- mutations 146, 262
 - clouds 266
- n**
- NANOG–OCT4–SOX2 network 125
- Nash equilibrium 274
- National Center for Biotechnology Information (NCBI) 446
- natural selection 262
- negative autoregulation 157
- negative decay rate 42
- negative feedback 156, 157
 - stabilization of protein levels by 165
- negative feedback loops 145
- network 8
- network-based models 16
- network component analysis 194
- network describing cell cycle dynamics
 - of *Saccharomyces cerevisiae* 126
- network motifs 150, 152
- network structures 145, 146, 151
 - groups of principles 151
 - analogous function and shaping for optimality 151
 - common origin or similar growth processes 151
 - – definition of the network 151
 - – material constraints 151
 - network picture revisited 152
- network with nodes 8
- neutral evolution 272
- neutrality 275
- neutral theory, mathematical models 270
- next-generation sequencing (NGS)
 - data 454
 - techniques 366, 367
- node distances 149
- noncatalyzed reaction 41
- noncompetitive inhibition 45
- nonequilibrium reactions 424
- nonlinear constraints 32
- nonnested models 102
- normal distribution 394
- normalization factor 51
- Northern blotting 363
- nuclear localization sequence (NLS) 355
- nuclear magnetic resonance (NMR) 448
- nucleic acids 345
- nucleus 349
- null hypotheses 151
- null space 29
- numerical integration 64
- numerical ODE solvers 64
- numerical optimization 245
- numerical parameter optimization 91
- o**
- Octave 470
- Oct4, Sox2, and Nanog (OSN) factors 20
- ODE. *see* ordinary differential equations (ODE)
- omics research 72
- OMIM (Online Mendelian Inheritance in Man) 447
- Ontology databases
 - gene ontology 449
 - Mouse Genome Database (MGD) 449
 - *Saccharomyces* Genome Database (SGD) 449
- optimal control 416
- optimal enzyme concentrations 255, 257
 - catalytic properties of single enzymes, optimization of 255
 - enzyme concentrations in a metabolic pathway, optimal distribution of 257
 - temporal transcription programs 259
- optimality 243
 - approaches in metabolic modeling 250
 - – enzyme fitness functions, measurements of 252
 - – enzyme levels, optimization of 251
 - – flux optimization 250
 - – mathematical concepts 245
 - – catalytic constants compromises 247
 - – cost–benefit models 245
 - – inequality constraints 246
 - – pareto optimality 246

- metabolic adaptation 253
- of enzyme activities 254
- optimal control profiles 254
- metabolic strategies 252
- fermentation 252
- product yield and enzyme cost, trade-off 253
- respiration 252
- metabolism modeling 248
- enzyme investments 250
- network structures 249
- short pathways preference 249
- thermodynamically feasible pathway 249
- yield efficiency 249
- optimization methods 97
 - genetic algorithms 98
 - global optimization 97
 - local optimization 97
 - sampling methods 98
- ordinary differential equation (ODE)
 - systems 16
- ordinary differential equations (ODE) 17, 24, 63, 124, 324, 467
 - based models 64
 - basic components of models 18
 - basic structure and properties of 63
 - illustrative examples of models 18
 - metabolic example 18, 19
 - regulatory network example 19–21
 - systems 25
 - for biochemical networks 17
 - for dynamics of reaction 42
- origin of life 334
- oscillatory input 414
- osmotic stress 314
- overrepresentation and enrichment analyses 436–438
- oxygen radicals 316

- p**
- pan-genome 10
- paradoxical regulation 159
- parameters 6
 - balancing 96
 - elasticity 52
 - estimation 44, 88
 - fluctuations 216
 - identifiability 90
- pareto optimality 246
- partial differential equations (PDEs) 63
- partial inhibition 46
- particle swarm optimization (PSO) 462
- partition function 419
- partitioning methods 430
- PathGuide 76, 445
- pathway crosstalk, with respect to the type-II diabetes mellitus candidate gene set 438
- pathway databases 76, 449
 - ConsensusPathDB 77, 451
 - KEGG 76, 450
 - Reactome 77, 450, 451
 - WikiPathways 77, 451
- pathway-related data and information 76
- pBR322 circular plasmid 360
- PCA. *see* principal component analysis (PCA)
- PDBe database 447
- Pearson's correlation coefficient 398, 399
- peptide linkage 344
- percentile 396
- per enzyme investment 250
- periodicity 7
- peroxisomes 351
- Petri Net Markup Language (PNML) 461
- Petri nets 124–127
 - model, simulation purposes 126
 - upper glycolysis represent as 126
- phenomenological thermodynamics 417
- phenotypical diversity, of organisms 335
- phenotypic switching 232
- PHML (Physiological Hierarchy Markup Language) format 466
- phospholipids 341
- phosphorelay systems 297
- phosphorylation 292
- phosphotransfer 292
- phylogenetic relations between some major groups of organism 335
- phylogenetic tree 4
- physical theories, chance in 405
 - deterministic chaos 406
 - perturbations by the environment 406
 - quantum-mechanical effects 406
 - underlying microscopic dynamics 406
- physiology 333
- PIR. *see* protein information resource (PIR)
- planning and designing experiments for case-control studies 426, 427
- Poincaré–Bendixson states 391
- Poisson distribution 17
- polyacrylamide gels 359
- polymerase chain reaction (PCR) 357, 365, 366
- polynomial regression model 402
- polypeptides 342
- polytene chromosomes 12
- population dynamics 274
- positive homotropic cooperativity 47
- posttranslational modifications 355
- power-law kinetics 49
- power-law modular rate law 49
- power of diagnostics 393
- practical nonidentifiability 90
- prediction of protein function 4
- preferential attachment model 149
- principal component analysis (PCA) 404, 405
- prisoner's dilemma 273
- probability distributions

- in discrete random walk 410
- for rate constants 94
- probability spaces 391, 392
- probability theory 392
- product experiments, and independence 395
- product formation 42
- product space 395
- prokaryotic and eukaryotic cells, comparison 334
- prokaryotic archaeabacteria 335
- promoter occupancy, thermodynamic models of 189
- promoter-operator concept 5
- protein chips 357, 367, 368
- protein databases
 - iHOP 449
 - InterPro 448, 449
 - PANTHER 448
 - Protein Data Bank 448
 - UniProt/Swiss-Prot/TrEMBL 448
- protein degradation 42
- protein information resource (PIR) 448
- protein investment, in different cell functions 161
- protein–protein interaction networks 146
- protein–protein interactions 16, 292
- proteins 127, 340
- proteins cross-linking 316
- protein sorting 355
- proteomic technologies 8
- PubMed 447
- Python modules 471

- q**
- qualitative model 7
- quality control 9
- quantitative proteomics data 9
- quasi-equilibrium 43, 107, 108
- quasispecies model 263
- quasi-steady-state 43, 107, 108

- r**
- random errors 90
- random fluctuations 217, 415
- random graphs 147
 - with predefined degree sequence 148
- random processes describing particle motion 409
- random variables 393
- Ras activation cycle 297
- Ras proteins 295
- Ras protooncogenes 297
- rate equations 45
 - deriving 43–45
- RBA. *see* resource balance analysis (RBA)
- reaction affinity 31
- reaction–diffusion
 - equation 137
 - models 121, 136
- reaction energetics 425
- reaction kinetics 39
- reaction–metabolite network 145
- reaction networks 25
- reaction pathways 30, 41
- reaction rate 42, 44
- reaction thermodynamics 40
- reactome 75
- real-world networks
 - scale-free degree distributions in 148
 - receptor–ligand interactions 293
 - reciprocal altruism 274
 - reciprocity 277
 - reduced and conditional distributions 407
 - reduction, of fast processes 105
 - relaxation time and other characteristic time scales 106, 107
 - time scale separation 105
 - reference sequence database (RefSeq) 446
 - regression 88
 - regularization 91
 - regulation edges and their steady-state response 156
 - regulation networks 23, 152
 - regulatory FBA 34
 - repeated games 274
 - replicator equation 274
 - resource balance analysis (RBA) 251
 - response coefficients 53
 - responsive switching 232
 - restriction endonucleases 357
 - restriction enzymes 358
 - recognize short stretches of DNA 358
 - reversible processes 7
 - ribonucleic acid (RNA) 345
 - RNA–DNA hybrid 353
 - RNA interference (RNAi) 11, 371
 - mechanism of 372
 - RNA polymerase 353
 - RNA primers 308
 - RNA-Seq (RNA-sequencing) 368
 - RNA synthesis 352
 - robustness mechanisms 217
 - by backup elements 219
 - in biochemical systems 218
 - against correlated expression changes 227
 - feedback control 219
 - limits of 228
 - role in evolution 228
 - – and modeling 228
 - scaling laws 224
 - by structure 222
 - – chemotaxis signaling pathway 223
 - – two-component system 222
 - – summation laws 227
 - – temperature compensation 228
 - – time scaling 227
 - ROC curve analysis 429, 430
 - rock–scissors–paper game, dynamical behavior 276

rule-based models 16, 17
 Runge–Kutta–Fehlberg method 64

s

Saccharomyces Genome Database (SGD) 11, 449
 sample size 426
 SBML files 464
 SBML model 74
 SBMLsimulator 469
 SBML Software Matrix 72
 SBOL (Synthetic Biology Open Language) 78
 scale-free networks 148, 149
 scaling laws 217

- allometric scaling 225
- geometric scaling 224
- power laws 224
- scaling relations, within cells 225

 SDS polyacrylamide gel electrophoresis (SDS-PAGE) 361, 362
 second law of thermodynamics 32, 419, 420
 SED-ML (Simulation Experiment Description Markup Language) 78
 selection criteria 102

- Akaike information criterion 102
- Bayesian information criterion 102
- calculated for 103

 selection equations 264
 selection processes 263
 selection threshold 265
 self-organization 261
 self-organizing maps (SOMs) 433, 434
 semantic annotations 79
 sensitivity analysis 210
 serine phosphorylation 295
 SGD. *see* *Saccharomyces* Genome Database (SGD)
 Shannon entropy 420
 sigmoid kinetics 48
 signaling cascade 26
 signaling molecules 152
 signaling networks 23
 signaling pathways 291

- crosstalk 306
- dynamic and regulatory features analysis 304
- intra and intercellular communication, function and structure of 292
- receptor–ligand interactions 293
- structural components 295
- G protein cycles 295
- MAP kinase cascades 296
- phosphorelay systems 295
- Ras proteins 295

 signaling systems process information 152
 signal-to-noise ratio 374
 simple linear regression 403
 simulation

- results of a VCell model 71
- techniques and tools 63
- tools 65

simultaneous binding modular rate law 49
 single-cell experiments 375
 single-cell methods 357
 single nucleotide polymorphisms (SNPs) 447
 single-stranded DNA (ssDNA) 358
 small-world networks 149, 150
 social interactions 272
 sodium dodecylsulfate (SDS) 361
 software tools

- Antimony 458
- Berkeley Madonna 459
- BIOCHAM (Biochemical Abstract Machine) 459
- BioNetGen 459
- Biopython 459
- BioTapestry 460
- BioUML 460
- CellDesigner 460
- CellNetAnalyzer 460
- ¹³C-FLUX2 458
- Copasi (Complex Pathway Simulator) 461
- CPN Tools 461
- Cytoscape 461
- E-Cell 461
- EvA2 (Evolutionary Algorithms framework, revised version 2) 461, 462
- FEniCS Project 462
- Genetic Network Analyzer (GNA) 462
- Jarnac 462, 463
- JDesigner 463
- JSim 463
- KNIME (Konstanz Information Miner) 463
- libSBML 464
- MASON 464
- Mathematica 464
- MathSBML 465
- Matlab 465
- MesoRD (Mesoscopic Reaction Diffusion Simulator) 465
- Octave 465, 466
- Omix visualization 466
- OpenCOR 466
- Oscill8 466
- PhysioDesigner 466, 467
- PottersWheel 467
- PyBioS 467
- PySCeS (Python Simulator for Cellular Systems) 467, 468
- R language 468
- SAAM II (Simulation Analysis and Modeling) 468
- SBMLEditor 468
- SBML-PET-MPI 469
- SBMLsimulator 469
- SBMLSqueezer 469
- SBMLToolbox 470
- SBML Validator 470
- SBToolbox2 (Systems Biology Toolbox 2) 470
- SemanticSBML 468, 469
- SensA 470, 471
- SmartCell 471
- STELLA 471

- STEPS (Stochastic Engine for Pathway Simulation) 471
- StochKit2 471, 472
- SystemModeler 472
- Systems Biology Workbench (SBW) 472
- Taverna 472, 473
- VANTED 473
- Virtual Cell (VCell) 473
- xCellerator 473
- XPPAUT 473, 474
- Southern blotting 363
- spatial models 133, 134
 - types of 134, 135
 - cellular automata 135
 - compartment models 135
 - reaction–diffusion systems 135
 - stochastic models 135
- spatial structure 278
- Spearman’s rank correlation 399
- spectral density matrix 415
- SPF. *see* S phase promoting factor (SPF)
- S phase promoting factor (SPF) 309
- splicing 353
- spontaneous pattern formation 139
 - Gierer–Meinhardt model 140
 - Turing instability 140
- S-systems approach 48
- standard deviation 397
- standard error
 - of the mean 426
 - of the ratio 426
- standards 9, 72
- state variables 7
- stationary 407
 - fluxes 31
 - metabolites 8
 - states 7
- statistical entropy 420
- statistical framework 400
 - error of first kind 400
 - error of second kind 400
- statistical models 16, 17
- statistical network analysis 8
- statistical relationships 145
- statistical Shannon information, signaling systems 153
- Statistics 391
 - for sample location 396
 - for sample variability 397
- steady states 7, 26
 - assumption 23
 - condition 27
 - fluxes 26, 51
- “stiff” differential equations 64
- stochastic modeling 6, 17, 127, 405
 - of biochemical reactions 127
 - of transcription and translation 197
 - genetic network fluctuations 199
 - macroscopic kinetic model 197
 - microscopic stochastic model 198
 - stochastic simulations 64, 129, 133, 318, 407
 - direct method 129
 - explicit τ -leaping method 129
 - first-order reaction 65
 - second-order reaction 65
 - stochastic and macroscopic rate constants 65
 - stochastic simulation and spatial models 130
 - StochKit2 user 472
 - stoichiometric coefficients 24
 - stoichiometric matrices 25, 26, 30, 214
 - stoichiometric models 121
 - stress-induced mutagenesis 234
 - structural analysis, of biochemical systems 23, 24
 - structural cell biology 345–347
 - structural nonidentifiability 90
 - structure diagram 70
 - substrate elasticity 52
 - substrate inhibition 46, 47
 - Sulfolobus acidocaldarius* 448
 - supermodels 8
 - support vector machines (SVMs) 439, 440
 - surface plasmon resonance (SPR) technique 376
 - sustainable modeling 78
 - SVMs. *see* support vector machines (SVMs)
 - Swiss Institute of Bioinformatics (SIB) 448
 - systematic single-gene knockout mutants 10
 - system equations 24
 - system response 414
 - Systems Biology Graphical Notation (SBGN) 74, 75, 460
 - activity flow diagrams 75
 - defining symbols 75
 - entity relationship diagrams 75
 - state transition diagrams 75
 - Systems Biology Markup Language (SBML) 9, 72
 - element similarities 79
 - semantics annotations in 79
 - systems biology models 4, 243
 - optimality in 243
 - teleological modeling approaches 244
 - Systems Biology Ontology (SBO) 469
 - Systems Biology Workbench (SBW) 460, 462, 463
 - system state 6

t

 - tacit assumption in pathway modeling 165
 - tags 72
 - TALE (transcription activator like effector) 371
 - TALENs (transcription activator like effector nucleases) 371
 - TATA-box 356
 - tau-leaping method 472
 - TaxTree search 451
 - TCA. *see* tricarboxylic acid (TCA) cycle
 - teleological modeling 244
 - telomere attrition 316
 - telomere shortening theory 315
 - temporal change of response coefficients 59
 - temporal evolution, of equation system 42
 - testing statistical hypotheses 399

tests for differential expression 427

– DNA arrays 427, 428

– next-generation sequencing 428

theorem of Glivenko–Cantelli 396

theoretical model 17

thermodynamics 15, 39, 41

– of chemical reactions 4

– constraints 31

– – on rate constants 94

– equilibrium and detailed balance 418

– kinetic rate laws 95

– laws, practical consequences for biochemical models 425

– – chemical potentials 425

– – constraints on model parameters 425

– – equilibrium states 425

– – partial fluxes 425

– – thermodynamic forces 425

– in systems biology 425

thermophilic bacteria 335

threonine 289

– synthesis pathway model 289

time-dependent flux response 60

time-dependent response coefficients 59

T lymphocytes 12

total protein concentration 257

transcription 4, 351

transcriptional feedback 220

transcriptional regulation networks 145, 164

transcription factor (TF) binding sites 357

transcription factor databases

– JASPAR 453

– Transcription Factor Encyclopedia 454

– TRED (transcriptional regulatory element database) 453, 454

transcription factors 153

– -initiated gene regulation 171

transcription networks 146, 153

– motifs 153

– network motifs in transcription network of *S. cerevisiae* 155

– positive and negative regulation 153

– potential regulation patterns with one, two, or three nodes 155

– regulation network of transcription factors in *E. coli* bacteria 154

– regulation structures and network motifs 155, 156

– transcriptional regulation of sugar utilization genes in *E. coli* bacteria 155

transcriptome 9

transcriptomics 8

transforming probability densities 395

transgenic animals 370

transition probabilities 407, 409

transition state theory 41

translation 4, 353–355

TrEMBL 447

trial-and-error process 262

tricarboxylic acid (TCA) cycle 286

α -trimmed mean 396

two sample location tests 400

– gamma function. 400

– unpaired Student's t-test 400

– Wilcoxon test 401

typical abstraction steps, in mathematical modeling 5

u

unbeatable 275

uncertainty analysis 211

– and principle of minimal information 211

uncompetitive inhibition 45

upper glycolysis, as realistic model 58

– flux and concentration control coefficients 58, 59

urn models 392

UV light fluoresces 359

v

validity criteria, for systems biology models 83

variability 210

– analysis 211

– and biochemical models 210

– – elasticity sampling 213

– – flux sampling 212

– – kinetic models, propagation of parameter variability 214

– – parameter fluctuations 216

– – uncertainty analysis 210

– propagation of 214

variables 6, 16, 30

variance 397

variational principle, for flux states 424

VCML (Virtual Cell Markup Language) 473

Virtual Cell (VCell) 70–72

visualization

– of sample characteristics by box plots 397

– techniques 9

w

Wegscheider conditions 423

Western blot 362, 363

Westfall and Young step-down correction 429

Wiener process 408

Wnt/ β -catenin signaling pathway 301

Wolfram SystemModeler Link (WSMLink) 472

x

xenografts 12

Xenopus laevis 334

XML-based native format 70

XML-compliant format 72

XML-like language style 9

y

yeast artificial chromosomes (YACs) 360

Yeast two-hybrid (Y2H) system 368, 369

z

zinc finger nucleases (ZFN) 370

zymomonas mobilis 253

WILEY END USER LICENSE AGREEMENT

Go to www.wiley.com/go/eula to access Wiley's ebook EULA.