

DEVELOPMENT AND APPLICATION OF WHOLE-CELL COMPUTATIONAL
MODELS FOR SCIENCE AND ENGINEERING

A DISSERTATION
SUBMITTED TO THE PROGRAM IN BIOPHYSICS
AND THE COMMITTEE ON GRADUATE STUDIES
OF STANFORD UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Jonathan Ross Karr
January 2014

© 2014 by Jonathan Ross Karr. All Rights Reserved.
Re-distributed by Stanford University under license with the author.



This work is licensed under a Creative Commons Attribution
3.0 United States License.

<http://creativecommons.org/licenses/by/3.0/us/>

This dissertation is online at: <http://purl.stanford.edu/nt763gr5676>

Includes supplemental files:

1. Table S1: Experimentally measured growth rates of wild-type M. genitalium and 12 single-gene disruption strains (tableS1.xls)
2. Table S2: Phenotypes of in silico wild-type and single-gene disruption strains (tableS2.xls)
3. Table S3: Whole-cell model parameters (tableS3.xlsx)
4. Movie S1: Life cycle of one in silico cell (movieS1.mp4)

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Markus Covert, Primary Adviser

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

Kerwyn Huang

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation for the degree of Doctor of Philosophy.

James Ferrell, Jr.

Approved for the Stanford University Committee on Graduate Studies.

Patricia J. Gumpert, Vice Provost Graduate Education

This signature page was generated electronically upon submission of this dissertation in electronic format. An original signed hard copy of the signature page is on file in University Archives.

Abstract

A central challenge in biology is to understand how phenotype arises from genotype. Despite decades of research which have produced vast amounts of biological data, a complete, predictive understanding of biological behavior remains elusive. Computational techniques are critically needed to assemble this wealth of data into a unified theory.

We have developed an integrative approach to computational modeling which enables comprehensive predictive models. We used this approach to construct the first “whole-cell” model. The model predicts the life cycle dynamics of the Gram-positive bacterium *Mycoplasma genitalium* from the level of individual molecules and their interactions, including its metabolism, transcription, translation, and replication. We validated the model by broadly comparing its predictions to a wide range of experimental data.

We have demonstrated that the model can guide biological discovery including determining how the *M. genitalium* metabolic network can regulate the cell cycle, enumerating the modes of cellular death, and determining metabolic kinetic parameters. We have also demonstrated how whole-cell models can guide rational biological design.

In addition, we have developed several software tools to facilitate whole-cell modeling, including databases for organizing training data and storing model predictions, and software for visually analyzing model predictions. Together, these technologies will accelerate bioengineering and medicine by enabling rapid *in silico* experimentation, facilitating experimental design and interpretation, and guiding rational biological design.

Preface

The work presented here consists of several studies focused on the development and application of “whole-cell” computational models of cellular physiology. Throughout my graduate career I had the great pleasure of working with numerous world-class collaborators. These studies would not have been possible without their invaluable contributions.

Chapter 2 — Reproduced with permission from Karr JR, Sanghvi JC, Macklin DN, Arora A & Covert MW. WholeCellKB: model organism databases for comprehensive whole-cell models. *Nucleic Acids Res* 41, D787–92 (2013); copyright 2013 Jonathan Karr, Jayodita Sanghvi, Derek Macklin, Abhishek Arora, and Markus Covert. This chapter describes a software application for organizing, browsing, and searching model organism databases for whole-cell models, as well as a model organism database of the Gram-positive bacterium *Mycoplasma genitalium*. I conceived and implemented the software and assembled much of the data contained within the *M. genitalium* database.

Chapter 3 and Appendix B — Reproduced with permission from Covert MW, Xiao N, Chen TJ & Karr JR. Integrated flux balance analysis model of *Escherichia coli*. *Bioinformatics* 24, 2044–50 (2008); copyright 2008 Markus Covert, Nan Xiao, Tiffany Chen, and Jonathan Karr. This chapter describes our early efforts to integrate multiple mathematical formalisms into a single predictive model. Markus conceived the study. Markus, Nan, Tiffany, and I worked together to implement the model and perform and analyze the *in silico* experiments.

Chapter 4 and Appendix C — Reproduced with permission from Karr JR*, Sanghvi JC*, Macklin DN, Gutschow MV, Jacobs JM, Bolival B, Assad-Garcia N, Glass JI & Covert MW. A whole-cell computational model predicts phenotype from genotype. *Cell* 150, 389–401 (2012); copyright 2012 Elsevier Inc. This chapter describes the first comprehensive computational model of any living organism. Markus, Jayodita, and I conceived the model. Jayodita and I developed the model with help from Jared. I performed the simulations. Jayodita, Derek and I analyzed the model’s predictions. I developed much of the software underlying the model. Jayodita led the validation experiments with help

from Miriam, Ben, and I and materials from Nacyra and John. Markus supervised all aspects of the project.

Chapter 5 — Reproduced with permission from Lee R*, Karr JR* & Covert MW. Whole-CellViz: data visualization for whole-cell models. *BMC Bioinformatics* 14, 253 (2013); copyright 2013 Ruby Lee, Jonathan Karr, and Markus Covert. This chapter describes a web-based software application for visually analyzing whole-cell model predictions. The project was conceived in collaboration with Ruby and Markus, and implemented in collaboration with Ruby.

Chapter 6 — This chapter describes several ongoing projects as of this writing, including collaborations with Veronica Llorens, Maria Lluch-Senar, Luis Serrano, and Markus Covert to construct a comprehensive model of *Mycoplasma pneumoniae*; with Pablo Meyer and Gustavo Stolovitzky to crowdsource the development of improved methods for estimating whole-cell model parameters; and with Nolan Phillips, Yingwei Wang, and Markus Covert to develop a database to store predicted whole-cell model trajectories.

Appendix A — Reproduced with permission from Karr JR, Guturu HG, Chen E, Blair S, Irish JM, Kotecha N & Covert MW. NetworkAnalyzer: Intracellular pathway animation. (In preparation); unpublished work copyright 2013 Jonathan Karr, Harendra Guturu, Ed Chen, Stu Blair, Jonathan Irish, Nikesh Kotecha, and Markus Covert. This chapter describes a web-based software application for visualizing high-throughput biological data in the context of signaling diagrams. This project was conceived through collaboration with Markus and Jonathan, and implemented in collaboration with Harendra, Ed, Stu, and Nikesh.

Acknowledgements

Graduate school has been an extraordinary experience. I've had the great fortune of helping build a promising new technology with far-reaching applications to biological science, bioengineering, and medicine. Throughout this experience I've learned an enormous amount and grown tremendously as a scientist and as a person. My experience would not have been possible without my numerous world-class colleagues and collaborators who worked alongside me, constantly challenging and inspiring me.

First, I would like to thank my thesis advisor Markus Covert for guiding me throughout graduate school, both professionally and personally. Markus is the model scientist and mentor I aspire to be. He always puts his students first, constantly guiding them toward scientifically fruitful paths and challenging them to greatness. By example, Markus has inspired me to be intensely creative, open-minded to new ideas and methods, and eager to tackle ambitious projects. I am truly grateful for the amazing graduate experience Markus has given me. I hope to give my students the same level of support and guidance Markus gave me.

I am immensely appreciative of my numerous colleagues who've supported me over the past seven years. In particular, I would like to thank Jayodita Sanghvi for embarking on a extremely challenging and ambitious project with me to pioneer a new area of computational systems biology. I would like to thank Jared Jacobs and Derek Macklin for their countless brainstorming sessions on whole-cell modeling and software engineering; the rest of the Covert Lab whole-cell modeling subgroup – Elsa Birch and Nick Ruggero – for our thoughtful discussions; my Covert Lab experimental collaborators – Ben Bolival, Silvia Carrasco, Miriam Gutschow, and Sergi Regot – for teaching me everything I know about experimental cell biology; the rest of the Covert Lab, past and present – Mialy DeFelice, Miriam Gutschow, Jake Hughey, Keara Lane, Tim Lee, and Nate Maynard – for making Clark W150 a fun and stimulating scientific home; Elissa Denny for shaping what has become the Covert Lab; Kimberly Chin, Jocelyn Hollings, Kathleen Guan, Lorie Langdon, and Thi van Anh Thach for smoothing out all of the administrative bumps on my path through graduate school; and Bill Weis and Vijay Pande for leading the biophysics graduate program which gave me complete flexibility to pursue my own interests while at Stanford.

I would also like to thank my numerous collaborators at Stanford and beyond: Harendra Guturu, Ed Chen, Nikesh Kotecha, Jonathan Irish, Stu Blair, Chad Rosenberg, and Tiffany Chen for collaborating with me on my first project in graduate school to develop pathway visualization software for flow cytometry; Tiffany Chen and Nan Xiao for jump-starting the integrated flux-balance analysis project which paved the way for whole-cell modeling; Nacyra Assad-Garcia and John Glass for your experimental expertise and sharing your *Mycoplasma genitalium* deletion strain library; Ruby Lee for spearheading the development of WholeCellViz; Abhishek Arora for inspiring me to improve WholeCellKB and teaching me about relational database design; Oliver Purcell, Bonny Jain, and Tim Lu for giving me an opportunity to explore the application of whole-cell modeling to synthetic biology; John Mason for leading the antibacterial modeling project; Veronica Llorens, Maria Lluch-Senar, Marie Jeanne Trussart, and Luis Serrano for challenging me to increase the accuracy of whole-cell models and for sharing your amazing wealth of experimental data; Pablo Meyer, Gustavo Stolovitzky, Thea Norman, Mike Kellen, Brian Bot, Bruce Hoff, Jay Hodgson, Brandon Allgood, Simon Wilkinson, and Christian Basile for collaborating with me to create a DREAM challenge on whole-cell model parameter estimation; and Nolan Phillips and Yingwei Wang for taking on the challenge of developing a database suitable for storing and querying whole-cell model predictions.

I would also like to thank my thesis committee and other faculty mentors. I would like to thank Ben Barres for creating Stanford's unique translational medicine master's program and giving me an avenue to pursue an education in translational medicine alongside computational systems biology research; Euan Ashley for introducing me to genomic medicine and graciously allowing me to observe Stanford Hospital's coronary care unit and inherited cardiovascular disease clinic; Kerwyn Huang for your thoughtful suggestions throughout graduate school; James Ferrell for my first experimental cell biology experiences; and Miriam Goodman for her passion for teaching and education, and for giving me a voice to shape the future of Stanford graduate education through the Bioscience advisory committee.

Lastly, I am deeply grateful for my amazing family and friends for your unqualified support. My best friend and fiancéé, Laurie Burns, stood by me through every hurdle in graduate school and celebrated every success with me. She constantly inspires me to be a better friend and person. My parents, Susan and Steven Karr, have always encouraged me to dream big and follow my passions. I would also like to thank my friends for reminding me that there's more to life beyond the lab; thank you for the amazing ski trips, bike rides, hikes, concerts, dinners, and more.

Contents

Abstract	iv
Preface	v
Acknowledgements	vii
1 Introduction	1
2 WholeCellKB: databases for whole-cell models	4
2.1 Introduction	4
2.2 Content	5
2.3 Curation	5
2.4 Comparison to existing resources	8
2.5 Data input	9
2.6 Data access	10
2.7 Developer API	10
2.8 Implementation	10
2.9 Summary and future directions.....	11
3 Integrated flux-balance analysis	13
3.1 Introduction	14
3.2 Methods	15
3.3 Results.....	20
3.4 Discussion	24
4 A Whole-cell model predicts phenotype	27
4.1 Introduction	27
4.2 Results.....	30
4.3 Discussion	45
4.4 Experimental Procedures	46

5 WholeCellViz: data visualization for whole-cell models	50
5.1 Background	50
5.2 Implementation	52
5.3 Results and discussion	54
5.4 Conclusions.....	60
5.5 Availability and requirements	61
6 Toward more accurate whole-cell models	62
6.1 A whole-cell model of <i>Mycoplasma pneumoniae</i>	62
6.2 Whole-cell model parameter estimation methods	63
6.3 WholeCellDB: database for whole-cell model predictions.....	64
7 Conclusion	66
A NetworkAnalyzer: intracellular pathway animation	69
A.1 Introduction	69
A.2 Features.....	70
A.3 Implementation	71
A.4 Discussion	71
B Integrated flux-balance analysis – Supplement	73
C A whole-cell model predicts phenotype – Supplement	78
C.1 Computational Methods.....	78
C.2 Cellular State Methods.....	88
C.3 Cellular Process Methods.....	123
C.4 Experimental Procedures	247
C.5 Computational Implementation	250
D List of supplementary materials	268
D.1 Supplemental tables	268
D.2 Supplementary movies	281

List of Tables

2.1 WholeCellKB-MG size	7
2.2 WholeCellKB-MG parameters	7
5.1 WholeCellViz visualizations	56
B.1 Model parameters	75
B.2 Enzyme subunit composition.....	76
B.3 Genetic perturbation phenotypes	77
C.1 Major sources of the <i>M. genitalium</i> reconstruction	85
C.2 Chromosome mathematical representation	91
C.3 Connections between the Protein Complex state class and other processes in the cell ...	109
C.4 Connections between the ProteinMonomer state class and other processes in the cell ...	112
C.5 Connections between ProteinMonomer state and other processes in the cell	117
C.6 Fixed parameters used in the Chromosome Condensation process class	129
C.7 Fixed parameters used in the Chromosome Segregation process class	132
C.8 State classes connected to the Chromosome Segregation process class	133
C.9 Fixed parameters used in the Cytokinesis process class	135
C.10 State classes connected to the Cytokinesis process class	137
C.11 Enzymes and complexes used in the DNA Supercoiling process class	147
C.12 Fixed parameters used in the DNA Supercoiling process class	148
C.13 State classes connected to the DNA Supercoiling process class.....	152
C.14 Fixed parameters used in the FtsZPolymerization orocess class	157
C.15 Enzymes and complexes used in the Replication process class.....	196
C.16 Fixed parameters used in the Replication process class	197
C.17 State classes connected to the Replication process class	199
C.18 Fixed parameters used in the Replication Initiation process class	204
C.19 State classes connected to the Replication Initiation process class	207
C.20 Fixed parameters used in the RNA Decay process class.....	214
C.21 State classes connected to the RNA Decay process class	215
C.22 Enzymes and complexes used in the Transcription process class	226
C.23 Fixed parameters used in the Transcription process class	231
C.24 State classes connected to the Transcription process class	233
C.25 Enzymes and complexes used in the Translation process class.....	239

C.26	Fixed parameters used in the <code>Translation</code> process class	239
C.27	State classes connected to the <code>Translation</code> process class	241
C.28	Enzymes and complexes used in the <code>tRNA Aminoacylation</code> process class	245
C.29	Fixed parameters used in the <code>tRNA Aminoacylation</code> process class	245
C.30	<code>Chromosome</code> class public methods	257
C.31	Chromosome computational representation	258
C.32	<code>ChromosomeProcessAspect</code> class public methods	258

List of Figures

1.1	Model integrates 28 submodels of diverse cellular processes	2
2.1	WholeCellKB-MG integrates diverse data sources into a single database	6
2.2	Molecular properties represented by WholeCellKB	8
2.3	Comparison of WholeCellKB-MG with existing biological databases	11
3.1	Model schematic	16
3.2	Simulation algorithm schematic	17
3.3	Predicted and observed glucose/lactose diauxic growth dynamics	21
3.4	Predicted glucose/lactose diauxic growth reaction fluxes	22
3.5	Predicted single-gene deletion phenotypes	23
4.1	Model integrates 28 submodels of diverse cellular processes	28
4.2	Model was trained with heterogeneous data and reproduces independent data	31
4.3	Model was trained using gene expression data and reproduces metabolic data	33
4.4	The model highlights the central physiological role of dna-protein interactions	35
4.5	The model predictions regarding regulation of the cell-cycle duration	37
4.6	Model provides a global analysis of the use and allocation of energy	40
4.7	Model identifies molecular pathologies underlying gene disruption phenotypes	41
4.8	Quasiessential and essential single-gene disruption strains exhibit reduced growth	42
4.9	Characterization of deletion strains identifies novel gene functions and kinetics	43
5.1	Cell cycle dynamics view	53
5.2	Additional WholeCellViz visualizations	55
5.3	Replication dynamics view	57
5.4	Population variance view	59
A.1	NetworkAnalyzer visualizes high-dimensional data in their pathway context	70
B.1	Predicted and observed glucose/glucose-6-phosphate diauxic growth dynamics	74
B.2	Predicted glucose/glucose-6-phosphate diauxic growth reaction fluxes	74
C.1	Model representation of cell geometry	98
C.2	Representation of volume of septum region	98
C.3	Protein monomer forms diagrammed in the context of the maturity pipeline	110

C.4	RNA forms diagrammed in the context of RNA maturation	116
C.5	SMC complex occupation of the DNA	129
C.6	Algorithm to bind, bend, and dissociate FtsZ filaments to pinch cell	136
C.7	Regions of varying superhelical density on the replicating chromosome	150
C.8	Enzyme activity profiles	151
C.9	Supercoiling transcriptional regulation	152
C.10	Metabolite perspective of the flux-balance analysis metabolic model	166
C.11	Schematic of DNA replication	196
C.12	DnaA DNA-binding state cooperativity	206
C.13	DnaA DNA-binding total cooperativity	207
C.14	Non-coding RNA cleavages	220
C.15	Hierarchical assembly of the <i>M. genitalium</i> terminal organelle	224
C.16	Comparison of consecutive dilutions	248
C.17	Whole-cell model architecture	250

List of Algorithms

C.1	Whole-cell dynamic simulation algorithm	79
C.2	Initial conditions identification algorithm	82
C.3	Cell state initialization procedure	83
C.4	Ribosome and Polypeptide state initialization	115
C.5	DNA damage simulation	141
C.6	DNA repair simulation	146
C.7	Host-parasite interaction simulation	161
C.8	Macromolecular complexation simulation	163
C.9	Metabolism FBA simulation	169
C.10	Protein activation simulation	172
C.11	Protein decay simulation	176
C.12	Protein refolding simulation	176
C.13	Protein unfolding simulation	177
C.14	Macromolecular complex degradation simulation	177
C.15	Aborted polypeptide degradation simulation	177
C.16	Protein monomer degradation simulation	178
C.17	Protein folding simulation	180
C.18	Protein modification simulation	184
C.19	Protein processing (I) simulation	187
C.20	Protein processing (II) simulation	190
C.21	Protein translocation simulation	195
C.22	Ribosome assembly simulation	213
C.23	RNA modification simulation	218
C.24	RNA processing simulation	223
C.25	Terminal organelle assembly simulation	225
C.26	Transcriptional regulation simulation	237

Chapter 1

Introduction

The “Holy Grail” of biology is to understand and predict how biological behavior emerges from the molecular level. Scientists widely believe that phenotypes arise from interactions among thousands of genetic and non-genetic factors. However, the details of this process are unknown. For example, which genes control the cellular growth rate? What combination of gene expression levels maximizes the cellular growth rate? Why do growth rates vary across individual cells?

Until recently, not enough data has been available about the molecules contained within cells to completely model a single organism. With the advent of high-throughput experimentation, researchers are now characterizing biological systems at an ever accelerating pace. Notably, Luis Serrano and his colleagues are rapidly characterizing the Gram-positive bacterium *Mycoplasma pneumoniae*, including its transcriptome²²⁵, proteome²⁰³, and metabolome⁴⁴⁵.

With these recent advances in biological measurement, the major remaining barrier to understanding how phenotypes arise is the lack of computational models that can represent heterogeneous cellular networks. Ordinary differential equation (ODE) models^{11,43,52,53,98,406} can represent many cellular networks, but require more data than is currently available. Boolean⁸⁸ and constraint-based modeling^{287,401} require fewer parameters. However their underlying assumptions are not consistent with all cellular networks. Building comprehensive cellular models entirely based on ODE, Boolean, or constraint-based modeling is therefore impractical. New methods that integrate heterogeneous data and mathematics into a single computational model are desperately needed.

Predicting how phenotype arises from genotype and the environment is central to biological science as well as bioengineering and medicine^{159,340}. Researchers critically need comprehensive computational models to integrate increasingly large and heterogeneous data sets. Bioengineers need mathematical models to more rationally and effectively invent biological systems capable of commodity chemical, biofuel, and drug production. Clinicians need predictive models to reason about each patient’s unique disease signature and design personalized prognoses and treatment programs. Consequently, predictive models that can account for the integrated function of every gene in a cell have the

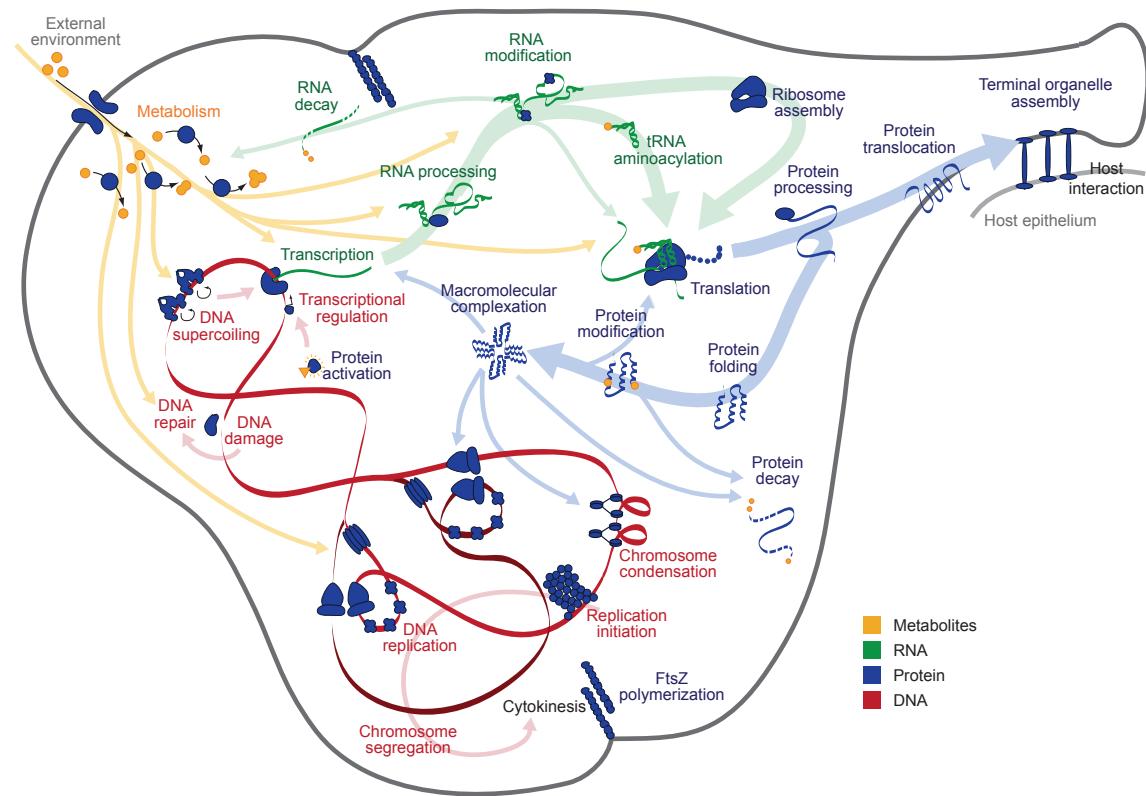


Figure 1.1. *M. genitalium* whole-cell model integrates 28 submodels of diverse cellular processes. Diagram schematically depicts the 28 submodels as colored words—grouped by category as metabolic (orange), RNA (green), protein (blue), and DNA (red)—in the context of a single *M. genitalium* cell with its characteristic flask-like shape. Submodels are connected through common metabolites, RNA, protein, and the chromosome, which are depicted as orange, green, blue, and red arrows, respectively.

potential to revolutionize biology and medicine⁸².

The primary challenges to predicting high-level biological behaviors from genotype are to integrate multiple mathematically distinct cellular networks into a single computational model, and to train models using noisy, heterogeneous data.

This thesis describes a novel integrative approach to cellular modeling in which cellular processes are independently modeled on short time-scales and integrated together at longer time scales to construct a comprehensive, unified model. The approach enables each process to be represented using the most appropriate mathematics and data, taking full advantage of all of the heterogeneous data and mathematics available in the scientific literature. For example, the approach allows metabolism to be modeled with flux-balance analysis and trained with chemical composition data at the same time

that transcription is modelled as a Markov process and trained using microarray data.

We used this approach to develop a model of the human pathogen and Gram-positive bacterium *Mycoplasma genitalium* (Figure 1.1). The *M. genitalium* model is the first “whole-cell” computational model to comprehensively explain the behavior of a single-cell from the molecular level. It describes the dynamics of every molecular species over the *M. genitalium* life cycle. It accounts for 28 cellular processes including metabolism, transcription, and translation; all annotated gene functions; and over 1,900 experimental measurements. It accurately predicts a wide range of observable cellular behaviors. The model also provides insights into many previously unobserved cellular behaviors, including in vivo rates of protein-DNA association, an inverse relationship between the durations of DNA replication initiation and replication, and previously undetected kinetic parameters and biological functions.

The first section of this thesis (Chapters 2–4) describes the new mathematical architecture and the development of the *M. genitalium* whole-cell model, including a new comprehensive database of *M. genitalium* molecular biology. Chapter 4 details the application of the *M. genitalium* model to basic biological discovery. Chapter 5 describes a new software tool for visually analyzing whole-cell model predictions in their biological context. Lastly, Chapter 6 introduces three ongoing projects to develop an even more comprehensive and accurate model of *M. pneumoniae*, to develop new tools for estimating whole-cell model parameters, and to develop an efficient database for storing and querying whole-cell model predictions. All of the software described in this thesis is available open-source under the MIT license at <http://simtk.org/home/iFBA> and <http://simtk.org/home/wholecell>.

Together these studies provide the first glimpse into whole-cell models and their broad applications to basic biological science, bioengineering, and medicine.

Chapter 2

WholeCellKB: model organism databases for whole-cell models

Abstract

Whole-cell models promise to greatly facilitate the analysis of complex biological behaviors. Whole-cell model development requires comprehensive model organism databases. WholeCellKB (<http://wholecellkb.stanford.edu>) is an open-source web-based software program for constructing model organism databases. WholeCellKB provides an extensive and fully customizable data model that fully describes individual species including the structure and function of each gene, protein, reaction, and pathway. We used WholeCellKB to create WholeCellKB-MG, a comprehensive database of the Gram-positive bacterium *Mycoplasma genitalium* using over 900 sources. WholeCellKB-MG is extensively cross-referenced to existing resources including BioCyc, KEGG, and UniProt. WholeCellKB-MG is freely accessible via a web-based user interface as well as via a RESTful web service.

2.1 Introduction

A primary challenge in computational biology is to predict how complex phenotypes such as growth and replication arise from networks of individual molecules. Whole-cell models promise to tackle this challenge by integrating heterogeneous molecular data into predictive computational models. This integration requires model organism databases which comprehensively provide readily-computable molecular data.

WholeCellKB is an open-source, web-based software program for developing comprehensive model organism databases for whole-cell models. As illustrated in Figure 2.1, WholeCellKB enables whole-cell modeling by organizing diverse molecular data from primary research articles, reviews, books,

and databases into a single database. The WholeCellKB data model supports detailed descriptions of individual species including their genes, operons, proteins, macromolecular complexes, molecular interactions, chemical reactions, and pathways. Importantly, WholeCellKB also facilitates extensive source documentation. We used WholeCellKB to develop WholeCellKB-MG, an extensive database of the pathogenic Gram-positive bacterium *Mycoplasma genitalium*.

Here we describe WholeCellKB-MG’s content, curation, and user interface, and implementation. We also compare WholeCellKB-MG to existing resources, highlighting WholeCellKB-MG’s greater scope and granularity. Lastly, we discuss our future plans for WholeCellKB.

2.2 Content

Our goal was to create a database comprehensive enough to enable a whole-cell model¹⁷⁵. As illustrated in Figure 2.2, WholeCellKB-MG broadly represents *M. genitalium* molecular biology including (i) its subcellular organization; (ii) its chromosome sequence; (iii) the location, length, direction, and essentiality of each gene; (iv) the organization and promoter of each transcription unit; (v) the expression and degradation rate of each RNA transcript; (vi) the specific folding and maturation pathway of each RNA and protein species including the localization, N-terminal cleavage, signal sequence, prosthetic groups, disulfide bonds, and chaperone interactions of each protein species; (vii) the sub-unit composition of each macromolecular complex; (viii) its genetic code; (ix) the binding sites and footprint of every DNA-binding protein; (x) the structure, charge, and hydrophobicity of every metabolite; (xi) the stoichiometry, catalysis, coenzymes, energetics, and kinetics of every chemical reaction; (xii) the regulatory role of each transcription factor; (xiii) its chemical composition; and (xiv) the composition of its laboratory growth medium. Table 2.1 summarizes WholeCellKB-MG’s size and content.

2.3 Curation

We curated WholeCellKB-MG in five steps based on more than 900 primary research articles, reviews, books, and databases. First, we curated the overall structure of *M. genitalium* including its

size, shape, subcellular organization, and chemical composition based on several experimental studies including Morowitz *et al.*, 1962²⁶¹. We also assembled the chemical composition of Mycoplasma laboratory growth medium based on analyses reported by Solabia³⁷³.

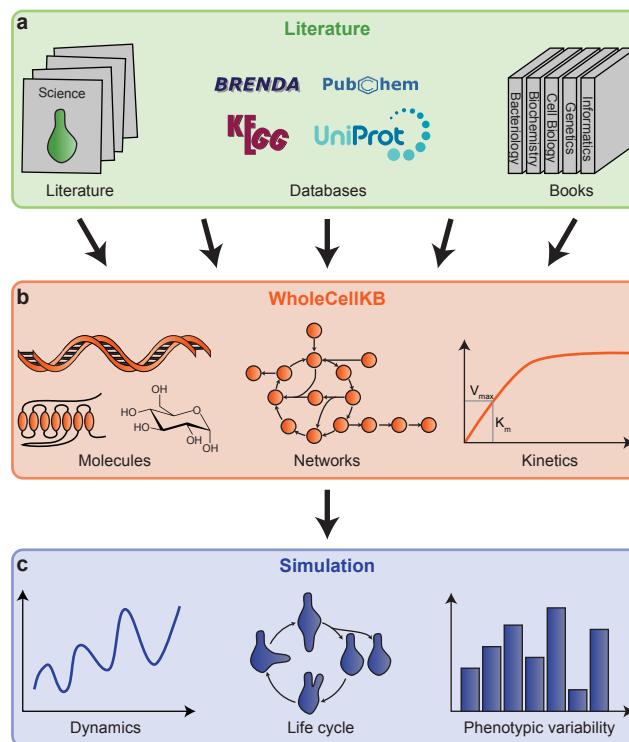


Figure 2.1. WholeCellKB-MG enables whole-cell modeling by integrating diverse data sources into a single database. (a) Currently, WholeCellKB-MG integrates more than 900 primary research articles, reviews, books, and databases. (b) WholeCellKB-MG comprehensively represents all aspects of molecular physiology including metabolomics, genomics, transcriptomics, and proteomics. (c) WholeCellKB-MG provides molecular data for whole-cell models.

Second, we curated the structure of the *M. genitalium* chromosome including its sequence, the location, length, and direction of each gene, and its transcription unit organization based on the Comprehensive Microbial Resource (CMR) annotation⁸⁷ and a recent study by Güell *et al.*, 2009¹⁴⁰. We reconstructed the location of each promoter and the expression, degradation rate, and essentiality of each gene product from four recent studies^{29,134,430,431}. We catalogued DNA-binding sites and transcriptional regulatory interactions from several sources including DBTBS³⁶².

Third, we assembled the structure of each RNA and protein gene product. We compiled the post-transcriptional processing and modification of each RNA transcript from several sources including

Table 2.1. WholeCellKB-MG size.

Entry type	No.
Cellular state	16
Chromosome feature	2305
Compartment	6
Gene	525
Metabolite	722
Pathway	17
Process	28
Protein complex	201
Protein monomer	482
Reaction	1857
Transcription unit	335
Transc. reg. interaction	30

Table 2.2. WholeCellKB-MG parameters.

Type	No.
Cell composition	73
Media composition	83
Reaction K_{eq}	225
Reaction K_m	483
Reaction V_{max}	434
RNA expression	525
RNA half-lives	525
Stimulus values	10
Transcriptional regulation	32
Activity	30
Affinity	2
Other	154

Peil, 2009²⁹⁸. We reconstructed the signal sequence, localization, chaperone-mediated folding, post-translational modification, disulfide bonds, subunit composition, and DNA footprint of each protein and macromolecular complex from a large number of primary research articles, computational models, and databases. We assembled the chemical regulation of each gene product from several sources including DrugBank¹⁹². We used ExPASy ProtParam¹⁰⁷ to calculate the pI, extinction coefficient, half-life, instability index, aliphatic index, and grand average of hydropathy of every protein species.

Fourth, we curated the specific chemical reactions catalyzed by each gene product starting from the CMR⁸⁷, GenBank²⁶, KEGG²²⁷, and UniProt⁷¹ genome annotations and the reconstructed RNA and protein maturation pathways. To maximize the scope of the database and to fill gaps in the genome annotation, we expanded each gene product's annotation based on primary research articles we identified by searching PubMed¹¹⁵ and Google Scholar (<http://scholar.google.com>). We consulted BioCyc¹⁸⁸, KEGG²²⁷, two flux-balance analysis (FBA) models of bacterial metabolism^{120,389}, and hundreds of additional primary research articles to curate the stoichiometry of each chemical reaction. We assembled the thermodynamics and kinetics of each chemical reaction from several databases including BRENDA²²⁹, SABIO-RK⁴³⁶, and UniProt⁷¹ and a FBA model¹²⁰.

Lastly, we compiled the *M. genitalium* metabolome. We included all metabolites involved in the reconstructed reactions, biomass or growth medium. We curated the empirical formula, structure, charge, and intracellular concentration of each metabolite from several databases including BioCyc¹⁸⁸, CyberCell³⁸⁷, and PubChem¹⁰⁶ and a comprehensive mass-spectrometry study²⁴. We used ChemAxon Marvin (<http://www.chemaxon.com/products/marvin>) to calculate the molecular weight, van der Waals volume, pI, \log_d and \log_p of each metabolite.

In order to create a comprehensive description of *M. genitalium* physiology, we based WholeCellKB-MG on studies of closely related organisms where studies of *M. genitalium* were unavailable. In cases where multiple observations were available, we based the reconstruction on the most closely related organism. We used bi-directional best BLAST³⁵⁶ to identify homologous genes. To provide model transparency we tracked the species, experimental conditions, and citation of each piece of evidence.

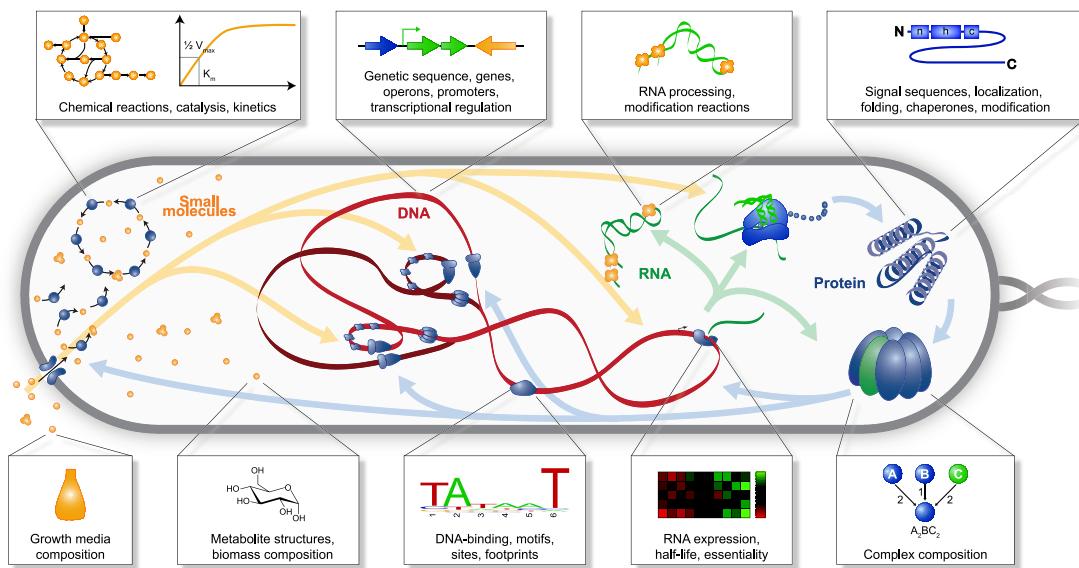


Figure 2.2. WholeCellKB aims to comprehensively describe cell physiology including the structure and dynamics of every metabolite, gene, RNA transcript, and protein. Boxes illustrate several molecular properties represented by WholeCellKB.

2.4 Comparison to existing resources

WholeCellKB represents the specific molecular interactions of individual species similar to previous databases such as BioCyc^{188,296} and BiGG¹⁶². In particular, WholeCellKB's data model, user interface, and species-specific content were heavily inspired by BioCyc.

Importantly, WholeCellKB-MG also has several major differences from existing resources. First, WholeCellKB-MG more broadly represents cell physiology. WholeCellKB-MG represents the molecular details of 28 cellular processes including well-studied processes such as metabolism as well as less well-understood processes such as DNA damage and repair and RNA and protein degradation. The online documentation at <http://wholecellkb.stanford.edu/about> provides further information

about the WholeCellKB-MG data model and how WholeCellKB-MG represents each cellular process. Figure 2.3 compares WholeCellKB-MG’s content to that of several existing databases.

Second, whole-cell modeling requires model organism databases which explicitly define the participants of each molecular interaction and chemical reaction. WholeCellKB-MG addresses this need by representing the specific molecules involved in every molecular interaction and by requiring structures for each molecule. For example, WholeCellKB-MG represents the specific RNA bases involved in every RNA methylation reaction, whereas existing resources lump RNA methylation interactions into a single generic reaction. WholeCellKB-MG represents every major cellular process including RNA processing and protein processing, modification, and translocation with similarly fine molecular resolution.

Third, where available WholeCellKB-MG contains not only structural, but also quantitative functional descriptions of each molecule and molecular interaction. For example, WholeCellKB-MG contains chemical reaction rate laws and kinetic parameters, RNA transcript expressions and half-lives, and cellular and growth medium chemical compositions. In total WholeCellKB-MG represents 1836 heterogeneous model parameters. Table 2.2 summarizes how WholeCellKB represents these heterogeneous parameters using several types of database entries.

2.5 Data input

WholeCellKB provides administrators with two editing interfaces: (1) a web form to edit single entries, and (2) an Excel-based interface to simultaneously edit multiple entries. We believe these two interfaces enable collaborative model organism database development.

In the beginning of our *M. genitalium* curation efforts we primarily used the batch interface to quickly import large amounts of data from other genome annotations. We continued to use the batch interface throughout the project to import high-throughput molecular data. Later in our *M. genitalium* curation efforts we primarily used the form interface to refine our annotation based on specific biochemical studies. Overall, we found that WholeCellKB improved the quality of our annotation and in particular encouraged us to thoroughly annotate the original source of each datum.

Data submitted to WholeCellKB was extensively validated to ensure consistency and correctness. For example, WholeCellKB checked that each chemical formula was valid, that each reaction was mass-balanced, and that every molecule and kinetic parameter was defined in each reaction rate law.

WholeCellKB provided hints on how to correct invalid data such as the atom imbalance of invalid reactions.

2.6 Data access

WholeCellKB-MG is freely accessible via a simple and intuitive web-based interface at <http://wholecellkb.stanford.edu>. This web-based interface allows users to quickly browse, search, and export the database. It also allows administrators to add, edit, and delete entries. Importantly, the interface is extensively commented and hyperlinked, allowing users to easily find the primary source of each datum.

WholeCellKB-MG is also accessible via a RESTful interface. This interface provides the content of every HTML page in JSON and XML formats. We are currently using this interface to develop software for visualizing whole-cell simulations.

2.7 Developer API

WholeCellKB was designed to enable modelers to develop model organism databases for whole-cell models, including designing custom data models and user interfaces. WholeCellKB provides a framework for viewing, searching, exporting, and editing database entries which developer's can combine with custom data models and HTML templates. This allows developers to build custom model organism databases with minimal effort and without any knowledge of database design. Furthermore, because WholeCellKB is open-source and implemented with Python, modelers can easily display scientific calculations alongside curated data in the user interface. The online documentation provides further instructions on how to customize WholeCellKB.

2.8 Implementation

WholeCellKB was implemented in Python using the Django (<http://www.djangoproject.com>) web framework and stored using the relational database MySQL (<http://www.mysql.com>). Full text search was implemented using Haystack (<http://haystacksearch.org>) and Xapian (<http://xapian.org>). Excel, JSON, and XML export were implemented using OpenPyXL (<http://bitbucket.org/ericgazoni/openpyxl>), simplejson (<http://pypi.python.org/pypi/simplejson>) and xml.dom (<http://docs.python.org/library/xml.dom.html>). WholeCellKB runs on the Apache (<http://www.apache.org>)

web server using the mod_wsgi (<http://code.google.com/p/modwsgi>) module. All of the software used to implement WholeCellKB is available open-source.

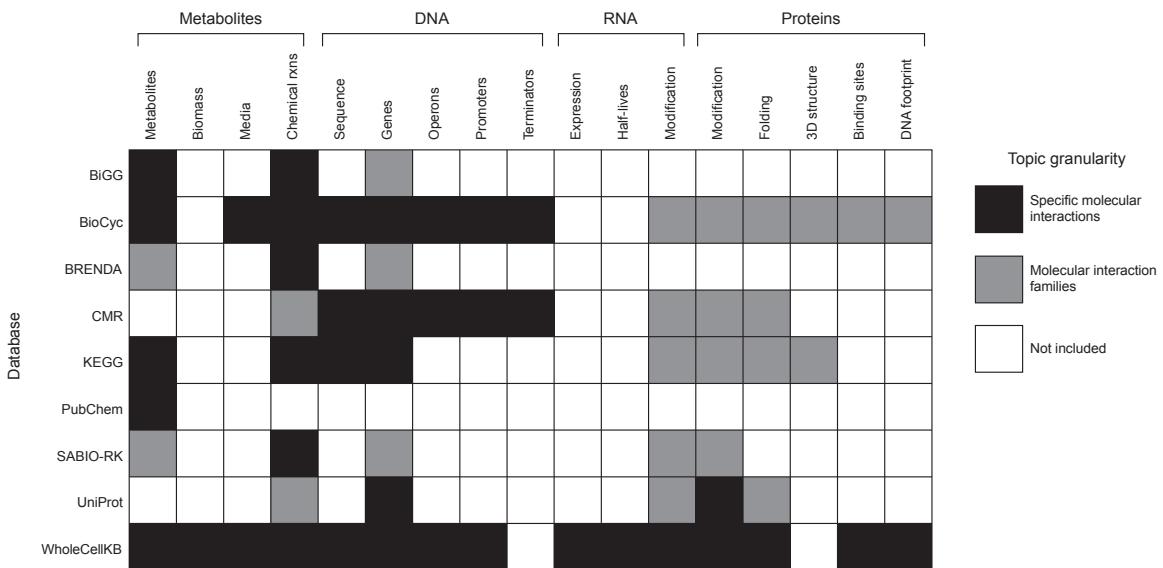


Figure 2.3. Detailed comparison of the content of WholeCellKB-MG and several existing biological databases. In addition to containing detailed descriptions of genetics, metabolism, and transcriptional regulation comparable to existing resources such as BiGG¹⁶², BioCyc¹⁸⁸, and CMR⁸⁷, WholeCellKB-MG has detailed representations of RNA degradation, RNA and protein maturation, and protein translocation. Black boxes indicate physiology represented with fine granularity including the specific molecules involved in each specific interaction (eg. specific metabolites involved in each metabolic reaction). Gray boxes indicate coarsely represented physiology, for example lumping families of similar reactions such as RNA methylation into a single database entry rather than representing the specific RNA bases involved in each individual reaction. White boxes indicate unrepresented physiology.

2.9 Summary and future directions

WholeCellKB-MG is an extensive database of *M. genitalium* designed to facilitate whole-cell modeling. Currently we are continuing to curate the database as well as starting to create equally comprehensive databases of other model microorganisms. Beyond facilitating realistic whole-cell models, we believe these databases are useful platforms for experimental and computational biologists.

We created WholeCellKB-MG using WholeCellKB, an open-source, web-based software program which enables modelers to quickly develop model organism databases for whole-cell modeling.

Beyond continuing to curate model organisms, we also plan to continue to strengthen the WholeCellKB software. We plan to add additional tools for importing databases curated with other tools such as PathwayTools²⁹⁶, storing the detailed history of each database entry, and comparing model organism databases, as well as expanding the search functionality of the RESTful API. As the whole-cell modeling community grows, in the future we also plan to enable open-editing similar to Wikipedia. Finally, we are currently using WholeCellKB’s RESTful API to develop tools for visualizing whole-cell simulations.

We hope that other researchers will use WholeCellKB to develop model organism databases and whole-cell models. We believe that WholeCellKB will not only speed up database curation and whole-cell model development, but also encourage best annotation practices. Ultimately we hope that WholeCellKB in combination with whole-cell models will accelerate biological discovery and bioengineering.

Chapter 3

Integrating metabolic, regulatory and signaling models

Abstract

Motivation: The effort to build a whole-cell model requires the development of new modeling approaches, and in particular the integration of models for different types of processes, each of which may be best described using different representation. Regulatory flux-balance analysis has been useful for large-scale analysis of metabolic networks and the associated transcriptional regulation, and of current interest is the integration of this approach with detailed kinetic models based on ordinary differential equations.

Results: We developed an approach to modeling the dynamic behavior of metabolic, regulatory and signaling networks by combining flux-balance analysis with regulatory boolean logic, and ordinary differential equations. We use this approach (called integrated flux-balance analysis, or iFBA) to create an integrated model of *Escherichia coli* which combines a flux-balance based, central carbon metabolic and transcriptional regulatory model with an ODE-based, detailed model of carbohydrate uptake control. We compare the predicted *E. coli* wild-type and single gene perturbation phenotypes for diauxic growth on glucose/lactose and glucose/glucose-6-phosphate with that of the individual models. We find that iFBA encapsulates the dynamics of 3 internal metabolites and 3 transporters inadequately predicted by rFBA. Furthermore, we find that iFBA predicts different and more accurate phenotypes than the ODE model for 85 of 334 single gene perturbation simulations, as well as the wild type simulations. We conclude that iFBA is a significant improvement over the individual rFBA and ODE models.

Availability: All MATLAB files used in this study will be available at <http://www.simtk.org/home/ifba/>.

3.1 Introduction

Can we build a model that accounts for all of the gene products in a cell? Certainly the effort to build a whole-cell model will depend on the development of new modeling approaches, and in particular the integration of models for different types of processes, each of which may be best described using different representation. Moreover, such an effort will likely identify novel and important cross-talk between different networks.

One approach that has been particularly successful in enabling large-scale modeling of carbon and energy metabolism is called flux-balance analysis (FBA). FBA has now been used to model metabolism in a host of microbial species, and has been expanded for a variety of applications (reviewed in²⁷⁶). Two extensions of FBA of interest here are the use of multiple FBA steps to simulate growth dynamics^{3,319,339}, and the incorporation of transcriptional regulatory network models^{76,391,392}. These expansions enabled us to integrate a regulatory network including 104 regulatory proteins with an existing model of 906 gene products involved in *E. coli* metabolism⁷⁵. We found that this integrated model (called regulatory flux balance analysis, or rFBA), significantly increased our ability to predict knockout strain phenotypes in a variety of environmental conditions (10,800 correct predictions out of 13,750 cases total⁷⁵). We also demonstrated the power of a model-driven approach to discovery, identifying over one hundred putative components and interactions in the *E. coli* metabolic and regulatory networks. Several of these have recently been verified experimentally¹⁷².

A major advantage of rFBA – requiring few kinetic parameters – could be a weakness in situations where the kinetic parameters have been determined and capture information not already contained in rFBA. For example, *E. coli* catabolite repression and its consequences on glycolysis has been modeled in great kinetic detail¹⁷⁷. It therefore seemed useful to create a framework which has rFBA’s ability to capture not only the metabolic pathways, but also the transcriptional regulation of an entire system, and the kinetic model’s greater level of detail. Other groups have integrated FBA with additional information, such as lin/log kinetics (¹⁷⁹) and coarse-grain time scale information¹⁷³. Furthermore, Yugi and colleagues showed that integrating metabolic flux analysis with more detailed kinetic descriptions reduces the amount of training data required to add additional reactions and metabolites to dynamic models¹⁸⁰.

Here we report the development of the iFBA framework, and the application of this framework to combining existing rFBA and kinetic models of *E. coli* central metabolism. Beyond the application

to *E. coli*, our approach differs significantly from the studies listed above in that we (1) integrate flux balance analysis of a metabolic network both with a boolean transcriptional regulatory network as well as a set of ODEs, and (2) incorporate two independently created models related to the same system and integrate them with minimal changes to either model. We see the resulting framework as an essential stepping-stone to development of a whole-cell model, enabling the integration of a wide variety of models of cellular processes.

We evaluated the integrated model by comparing its predicted time courses for wild-type and single gene perturbation *E. coli* diauxic growth on glucose/lactose and glucose/glucose-6-phosphate with that of the individual rFBA and ODE models and experimental data¹⁷⁷. We find that the integrated model is a significant improvement over the individual rFBA and ODE-based models, generating simulations which are more globally accurate and informative than the ODE-based model, and more accurate in their details than the rFBA model alone.

3.2 Methods

To create the integrated model, we combined a kinetic model of *E. coli* phosphotransferase (PTS) catabolite repression developed by Kremling and colleagues (Figure 3.1A², with an rFBA model of the same system (Figure 3.1C²⁶⁹. The rFBA model was expanded from that described in Covert et al., 2002²⁶⁹ to describe glucose-6-phosphate uptake by UhpT. The modified rFBA model describes the uptake and production of 11 carbohydrates, glycolysis, the pentose phosphate pathway, the TCA cycle, and the production of intermediate energy stores using biomass, 77 metabolites, 87 enzymes, and 16 transcription factors that regulate 46 out of 113 metabolic reactions. The Kremling at et. 2007 ODE model describes the uptake of 3 carbohydrates and glycolysis using biomass, 6 metabolites, and 4 proteins – 3 transporters and two phosphorylation states of the protein EIIA^{Crr}, a component of the phosphotransferase system as well as 16 metabolic and transport reactions². The integrated model describes biomass, 77 metabolites, 151 genes, and 113 reactions.

As illustrated in Figure 3.1B, we integrated the rFBA and ODE models by identifying values to pass from either model to the other. First, we identified the complete set of metabolites and fluxes common to both models (common metabolites shown as black circles, all ODE reactions are in this case, although not necessarily in every case, common to both models). The variables passed from the ODE model included fluxes which were not directly subject of global effects. These included enzyme fluxes v_{pts} , v_{lacY} , v_{uhpT} , and v_{pykAF} , as well as changes in metabolite concentrations, which we call

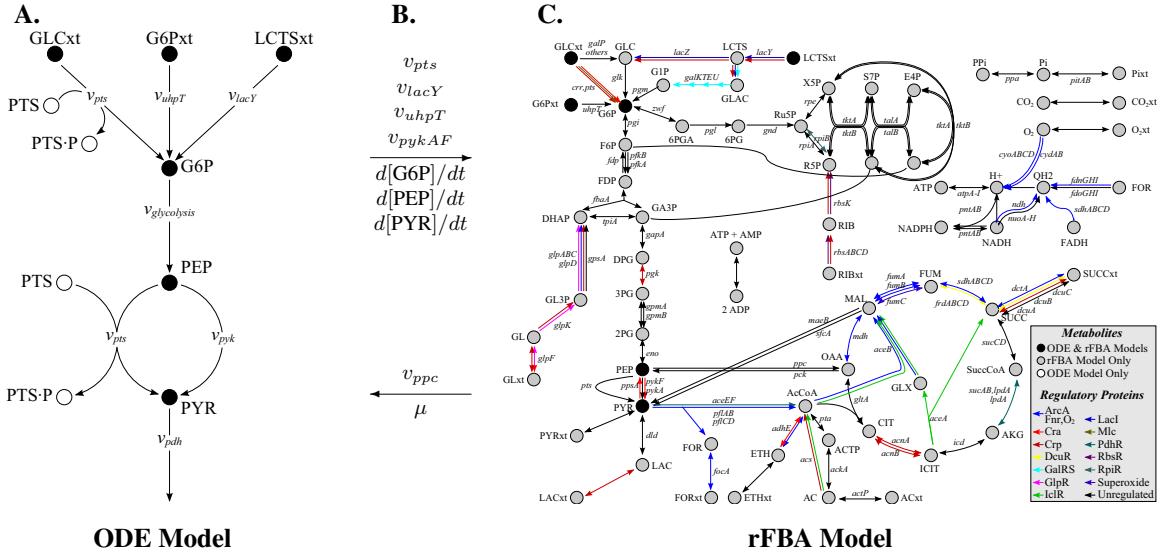


Figure 3.1. Diagram of the metabolic, regulatory and signaling networks used to build the iFBA model. (A) A schematic of the complete ODE model². (B) A list of the variables passed between the models as part of iFBA. Arrows indicate whether a value is being passed from the ODE model to rFBA or vice versa, and metabolites common to both the rFBA and ODE models are colored black. (C) A schematic of the complete rFBA model²⁶⁹. Regulated fluxes are indicated by arrow color according to the key (lower right).

“metabolite pooling fluxes” ($d[G6P]/dt$, $d[PEP]/dt$ and $d[PYR]/dt$). The variables passed from the rFBA model include the growth flux (μ) and the flux through phosphoenolpyruvate carboxylase (v_{ppc}). The growth flux was passed from rFBA because the ODE calculation of growth depends only on substrate uptake and neglects important global growth requirements. v_{ppc} was determined by rFBA because it was not included in the ODE model but can have an important effect on phosphoenolpyruvate concentration.

3.2.1 iFBA simulation algorithm

The following paragraphs describe each step of the iFBA simulation algorithm illustrated in Figure 3.2. Briefly, starting from initial conditions or those calculated in the previous time step, we first numerically integrated the ODE model and computed the regulatory constraints using the boolean regulatory model. Next, we constrained the primal of the FBA linear programming problem using the ODE and regulatory models, updated the right-hand-side of the FBA linear programming problem according to the pooling fluxes calculated by the ODE model, and solved for the FBA fluxes. Finally, we updated the biomass and external metabolite concentrations for use in subsequent time

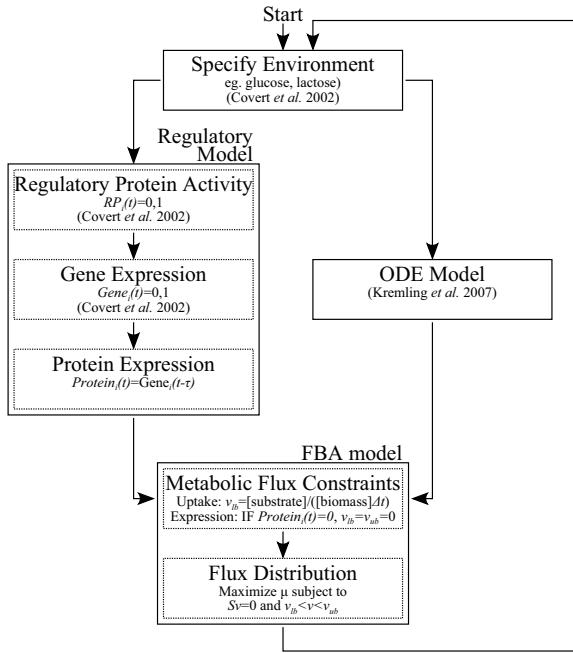


Figure 3.2. Schematic of the simulation algorithm.

steps. The length of each time step was chosen to be large enough that the FBA assumption that the concentrations of internal metabolites are time invariant holds, and yet small enough for the ODE model to calculate the system dynamics without accumulating numerical error. Although we used a time step of 3 min, we empirically found at any time step in the range 30 sec and 5 min gave the same results.

Specify initial environment – Initial conditions for the biomass, enzymes, and metabolites in the ODE model and the corresponding rFBA biomass and metabolites, where applicable, were obtained from Kremling et al., 2009², and are listed in Table B.1. Initial conditions for the 16 regulatory proteins were determined by the regulatory model under the additional assumption that the bacteria were in steady state with the external environment prior to the start of the simulation.

Calculate regulatory protein activity, gene and protein expression – Transcriptional regulation imposes time dependent constraints on the metabolic network. The activity of each regulatory protein, as well as expression of regulated genes and proteins was described using the Covert et al., 2001 boolean regulatory model with time delay⁷⁶, except that in cases where activity and expression were encapsulated by the ODE model – Crp, galEKMPT, lacYZ, pgk, ptsG, and pykF – the ODE

model-determined values superseded the boolean regulatory values.

Solve ordinary differential equations – At each time step we used the MATLAB ode15s function to numerically integrate the ODE model using the growth rate and *ppc* flux computed by the FBA model at the previous time step. Next we calculated the ODE rates at the end of the time step to later constrain the FBA linear programming problem.

Determine metabolic flux constraints and metabolite pooling fluxes – There are several types of metabolic flux constraints in iFBA: (1) irreversibility constraints, where the lower bound of the reaction is set to zero for reactions which can only proceed in the forward direction; (2) environmental constraints, where the maximum flux through an exchange reaction is limited by the amount of substrate in the culture medium; (3) transport constraints, which are represented as a maximum substrate uptake or by-product secretion rate; (4) regulatory constraints, where the flux through an enzyme is restricted by the expression of the corresponding protein(s); and ODE matching constraints, where fluxes passed by the ODE model are completely specified by the ODE model. Irreversibility constraints are determined from the literature²⁶⁹. Environmental constraints on the exchange fluxes, v_{ex} , were computed according to the scheme described by³. These constraints are then compared to the experimentally-derived transport constraints (listed in Table B.1), and the more restrictive constraints are used to bound the exchange reaction for the given time step. Regulatory constraints were derived from the expression profile of regulated proteins in the metabolic network over time. If the boolean rule indicated at some time t that protein i is expressed, then the corresponding reaction is not constrained and the metabolic flux distribution may calculate any value for that reaction, given the other non-regulatory constraints that also control the system. However, if the rule indicated that protein i is not expressed at time t , then the corresponding reaction flux was constrained to zero,

$$v_i(t) = 0. \quad (3.1)$$

Finally, ODE matching constraints included any flux represented in both the ODE and FBA models – v_{pts} , v_{lacY} , v_{uhpT} , and v_{pykAF} , and were implemented by setting the upper and lower bounds of the FBA fluxes equal to the corresponding rate calculated by the ODE model.

To capture the internal metabolite concentrations in iFBA, we incorporated metabolite pooling fluxes. The normal FBA mass balance equations assume that the concentrations of internal metabolites are time invariant. However, the ODE model calculates time-variant metabolite concentration

profiles, in our case for glucose-6-phosphate, phosphoenolpyruvate, and pyruvate. Metabolite pooling fluxes were implemented by setting the corresponding entries in right-hand-side of the FBA linear programming problem equal to the rates of change of their concentrations calculated by the ODE model – $d[G6P]/dt$, $d[PEP]/dt$ and $d[PYR]/dt$.

Calculate flux distribution – Fluxes were calculated by maximizing biomass production subject to the FBA mass balance equations using the open-source COIN-OR Linear Program Solver (CLP, freely available at <http://www.coin-or.org/>). The biomass mass balance equation was based on experimental data¹⁷⁰.

Calculate new environment – The growth rate and fluxes computed by the FBA model were next used to update the biomass and metabolite concentrations according to the scheme described by³,

$$[\text{biomass}](t + \Delta t) = \beta [\text{biomass}](t) e^{\mu \Delta t} \quad (3.2)$$

$$[\text{met}_i](t + \Delta t) = [\text{met}_i] + \frac{v_{ex}}{\mu} [\text{biomass}](t) (1 - e^{\mu \Delta t}), \quad (3.3)$$

where β is a growth rate scaling factor introduced to fit the experimental data obtained by¹⁷⁷ for *E. coli* diauxic growth on glucose/glucose-6-phosphate and glucose/lactose with the biomass equation experimentally determined by¹⁷⁰ for *E. coli* B/r growth on glucose minimal medium.

At the following time step the growth rate and flux through Ppc are used to correct the ODE phosphoenolpyruvate pooling flux to account for conversion to oxaloacetate by Ppc, and to calculate ODE rates and states.

3.2.2 Single gene perturbations

Single gene knockouts were implemented by setting the upper and lower bounds of the corresponding FBA flux(es) to zero, setting the values of the corresponding ODE kinetic parameter(s) to zero, and setting the expression of the corresponding transcription factor(s) to zero. For regulatory proteins, we could also simulate knock-in of a constitutively active transcription factor by setting the activity of the corresponding transcription factor(s) to one. The correspondences between ODE kinetic parameters and rFBA relationships are listed in Table B.2.

3.3 Results

We evaluated the integrated model by comparing the model's predictions for wild-type and single gene perturbation *E. coli* diauxic growth on glucose/lactose and glucose/glucose-6-phosphate with that of the individual rFBA and ODE models, and where available, experimental data^{177,178}.

3.3.1 Diauxic growth on glucose/lactose

Growth on glucose and lactose as carbon sources involves catabolite repression, leading to the preferential uptake of glucose and subsequent lactose uptake. Figure 3.3 shows the iFBA, rFBA and ODE wild type simulations together with experimental data. We found that although all three types of simulations were equally able to predict carbon source uptake and biomass production (Figure 3.3A,B), they were significantly different in other aspects. For example, acetate secretion was observed by Bettenbrock et al., 2006¹⁷⁷ under these environmental conditions. iFBA and rFBA were both able to account for acetate secretion under the given environmental conditions, but not the ODE model.

Additionally, the internal concentrations of phosphoenolpyruvate and pyruvate differed significantly between all iFBA and the individual models. Because the rFBA model assumes that the concentrations of internal metabolites are time invariant, it did not encapsulate any of the dynamics seen in the ODE and iFBA simulations (Figure 3.3C). The differences between the iFBA and ODE models in predicting internal concentrations reflected two fluxes which are included in the iFBA model but not considered in the ODE model. First, the iFBA growth flux is a drain on several key metabolites, including glucose-6-phosphate, glyceraldehyde-3-phosphate, 3-phosphoglycerate, phosphoenolpyruvate and pyruvate in glycolysis. Loss of these metabolites to biomass results in a small but significant reduction in phosphoenolpyruvate conversion to pyruvate. Second, the Kremling model assumes that pyruvate kinase and Pts are the dominant enzymes which utilize phosphoenolpyruvate, and that the phosphoenolpyruvate carboxylase flux is negligible². However, the iFBA model predicts that the Ppc flux is 5-15% of the total flux utilizing phosphoenolpyruvate (Figures 3.4, S2).

Modeling the transporter UhpT and PtsG concentrations led to similar results for the ODE and iFBA model, while the rFBA simulations exhibited step-like dynamics, due to the underlying boolean rules (Figure 3.3C). Similarly, rFBA did not encapsulate the complex behavior of EIIA^{Crr} because its behavior has been shown to be very complex and vary across carbon substrates¹⁷⁸. Consequently, its

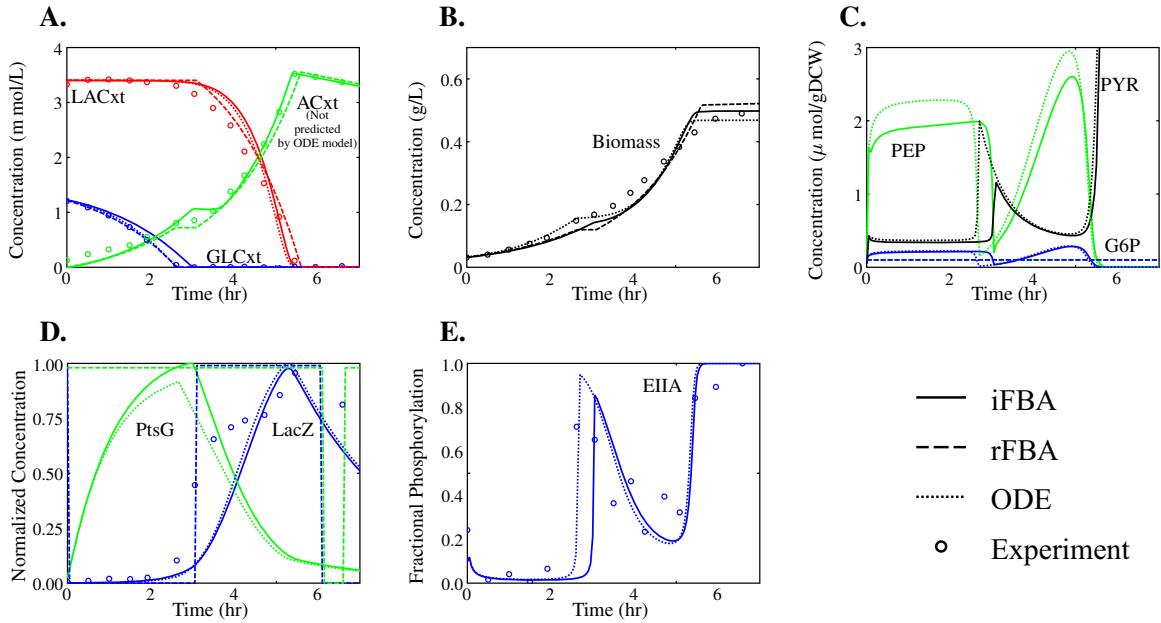


Figure 3.3. Growth of the iFBA (solid lines), ODE (dotted) and rFBA (dashed) wild type models in an aerobic environment with glucose and lactose as carbon sources, together with experimental data² where available (circles). Dynamic time profiles of external (A) acetate, glucose, lactose and (B) biomass concentrations; (C) internal pyruvate (PYR), phosphoenolpyruvate (PEP) and glucose-6-phosphate (G6P) concentrations; (D) key protein concentrations; and (E) degree of phosphorylation of regulatory protein EIa^{Crr}.

behavior is not fully described by the logic rules of the rFBA boolean regulatory model (Figure 3.3D).

The iFBA model includes over 100 additional genes and corresponding regulation or reactions, beyond what is included in the ODE model. The iFBA simulation therefore includes a large amount of additional data, such as changes in global gene expression and flux distributions. For example, the metabolic flux distributions at one hour and five hours are shown in Figure 3.4. At one hour, when bacteria are consuming glucose there is significant flux from internal glucose-6-phosphate through the pentose phosphate pathway. However, at five hours, when bacteria are consuming lactose, the flux from glucose-6-phosphate has shifted toward glycolysis. Additionally, at five hours the lactose-related transcription factors GalE, GalM, GalK, and GalT are now expressed, PtsG is now suppressed, and bacteria secrete ethanol in addition to acetate. Changes in gene expression were calculated using iFBA and rFBA as shown, and involve induction of the proteins required to utilize lactose as a carbon source.

3.3.2 Diauxic growth on glucose/glucose-6-phosphate

E. coli uptake of glucose and glucose-6-phosphate is concurrent, with some repression of the glucose transporter. Figure B.1 shows the iFBA, rFBA and ODE wild type simulations together with experimental data. We similarly observed that all three models describe the wild type experimental external glucose and glucose-6-phosphate, and biomass data equally well (Figure B.1A,B). Again the iFBA and rFBA models predicted similar rates of acetate secretion, and again the predicted concentrations of the internal metabolites glucose-6-phosphate, phosphoenolpyruvate and pyruvate, differed between the iFBA and ODE models (Figure B.1C) due to consideration of phosphoenolpyruvate conversion to oxaloacetate by Ppc and more detailed consideration of metabolite conversion to biomass. The concentration and activity profiles for transporters UhpT and PtsG, as well as EIIA were also similarly represented by the iFBA and ODE models, but not by rFBA (Figure B.1D,E) Finally, at the network level, in contrast to what we observed during glucose/lactose diauxie, we saw large shunting of the flux away from glucose-6-phosphate from glycolysis to the pentose phosphate pathway – 65% of the input flux was shifted from glycolysis to the pentose phosphate pathway between 1 and 6 hours (Figure B.2).

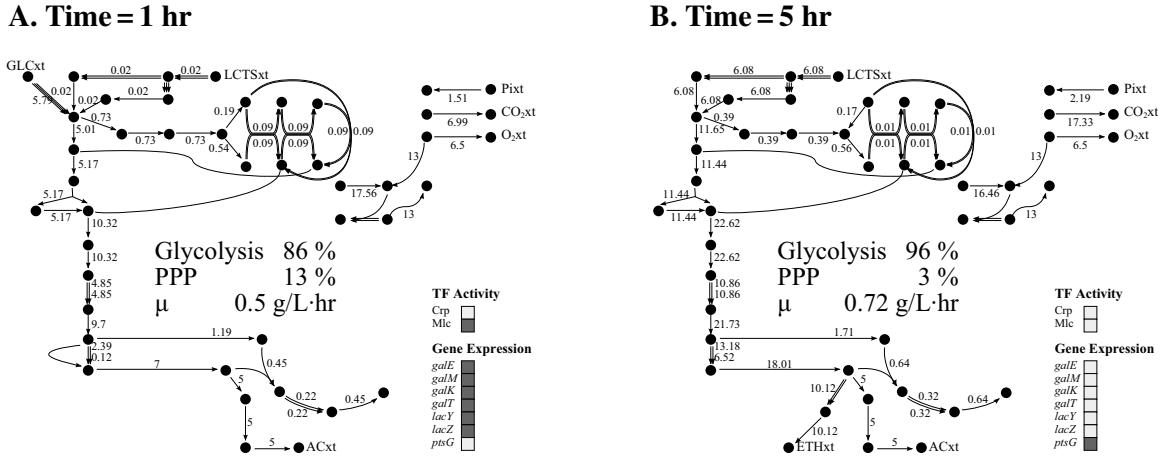


Figure 3.4. Flux distributions for iFBA simulation of glucose/lactose wild type diauxic growth, at (A) one hour and (B) five hours. Detailed labels for the network are shown in Figure 3.1, and all values are in m mol/gDCW/hr. Selected qualitative gene expression values calculated using the rFBA module of iFBA are also shown, where light gray denotes expression and dark gray denotes repression.

3.3.3 Single gene perturbation analysis

To further compare the iFBA, rFBA and ODE models we simulated diauxic growth of 167 *E. coli* single gene perturbations on glucose/glucose-6-phosphate and glucose/lactose – 151 knockouts of 135 enzymes and 16 transcription factors, and 16 cases where we forced each transcription factor to be constitutively active. We found that the iFBA model predicted different phenotypes than the ODE model for 41 and 44 of the mutants on glucose/glucose-6-phosphate and glucose/lactose, respectively. As illustrated in Figure 3.5, the genes corresponding to these mutants can be grouped into five classes – 6 TCA cycle genes, 24 intermediate energy storage genes, 3 carbohydrate transport genes, 7 glycolysis genes, and 5 transcriptional regulatory genes.

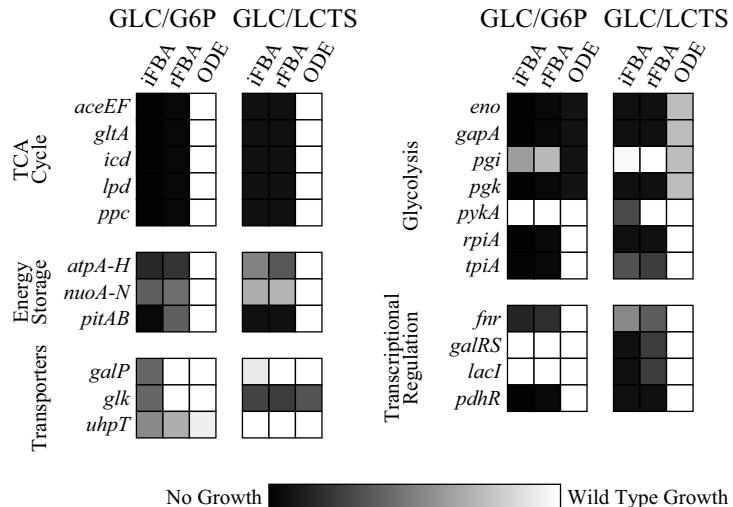


Figure 3.5. Gene perturbation analysis. The ratio of mutant to wild type biomass concentration at 8 hours is shown for all mutants where differences were observed between the iFBA, rFBA and ODE-based simulations. All perturbations are knockdowns except for the catabolite repression genes where we forced the corresponding transcription factor to be constitutively expressed. All simulation results are found in Table B.3.

For most of the 85 cases, the iFBA and rFBA models predicted the correct outcome of gene perturbation observed in various reports (reviewed in²⁶⁹), while the ODE model failed to predict the correct outcome in many of these cases. We investigated these differences in more detail. For the TCA cycle and intermediate energy storage genes, the ODE model was unable to predict the effects of gene deletion because it does not consider metabolic pathways beyond glycolysis and therefore does not include these genes. For transport and glycolysis, the ODE model includes the corresponding genes, but incorrectly predicts that these deletions will be non-lethal. This failure is because

the ODE model's equation for biomass is based only on transport of the extracellular metabolites, and not on the ability to produce biomass components. The *pgi* deletion represents an unusual case where the ODE model predicts a more negative impact on growth. We found that this difference is because the rFBA and iFBA models include the pentose phosphate pathway which is used as an alternate route from F6P to the TCA cycle when *pgi* is deleted. Forced constitutive activation of 4 transcription factors also led to repression of key genes whose absence had a negative impact on growth in the iFBA and rFBA models but were not included in the ODE model.

Finally, we found that the iFBA model predicted different phenotypes than the rFBA model for 2 mutants on glucose/glucose-6-phosphate – *galP*, and *glk*, and 1 mutant on glucose/lactose – *pykA*. These predictions highlight the advantage of the iFBA model over the rFBA model to include the subtle effects of the dynamics of the internal metabolites glucose-6-phosphate, phosphoenolpyruvate, and pyruvate.

3.4 Discussion

The iFBA model described above has several strong advantages over both the rFBA and ODE models. First, the kinetic description in the iFBA model contains a much greater level of detail than the rFBA approach to modeling regulatory activities and events. This would be critical in cases where genes have multiple stable expression states, as has been observed with lac operon regulation⁴⁴¹, as well as with the action of Crp and EIIA^{Crr} described here. Similarly, because of FBA's quasi steady-state assumption described above, the concentration of metabolites internal to the cell is not calculable without the kinetic model. This is also critical where regulatory protein activities depend on internal metabolite concentrations, which previously have been approximated by either external metabolite concentrations or combinations of metabolic fluxes²⁶⁹. Including the set of ODEs makes such crude approximations unnecessary. The importance of incorporating detailed kinetic information was also underscored by the *pykA*, *galP* and *glk* knockout simulations, where the iFBA model made significantly different predictions than the rFBA model due the effects of internal metabolite concentrations on the system.

A second advantage of iFBA over rFBA is that certain enzymes are expressed and active which would never be part of a strictly optimal growth scenario. This is because they are utilized not for their metabolic contribution to growth, but for other important functions such as signal transduction. As an example, the adenylate cyclase enzyme catalyzes the conversion of ATP to cyclic AMP (cAMP), a

key mediator of catabolite repression. cAMP is not required as part of the growth objective function in the FBA model, and therefore the adenylate cyclase flux wastes ATP and would never be used. In fact, adenylate cyclase is one of many reactions in reconstructed metabolic networks listed as “dead ends” in FBA because they lead to production of metabolites that would never be used as part of a growth-optimal solution¹⁷¹. However, the phosphorylation of EIIA^{Crr} demonstrated by iFBA would result in the utilization of this flux to generate cAMP¹⁷⁸. This raises the possibility that many of the FBA-determined “dead ends” are in fact “gateways” to other important cellular networks such as cell signaling.

We also found that the iFBA model has certain advantages over the ODE model. In particular, we observed that the iFBA model enabled us to see the global effects of dynamic changes in the Kremling model, because of its ability to calculate a flux distribution for an entire network with only few additional parameters. An important illustration of this is the experimentally-determined secretion of acetate under glucose/lactose diauxie, which is captured in the iFBA model but not the ODE model. Another example is the predicted shift of metabolic flux from the glycolysis to the pentose phosphate pathway as glucose-6-phosphate is depleted in glucose/glucose-6-phosphate diauxie, which could not be determined using the ODE model. In this case we also saw that the flux through phosphoenolpyruvate carboxlyase, which was assumed to be negligible in the ODE model, made up a significant percentage of the flux from phosphoenolpyruvate, resulting in a substantially lower predicted internal phosphoenolpyruvate concentration. This prediction correlates with the experimental observation that *ppc* knockout strains are unable to grow with glucose as the sole carbon source¹⁶⁵.

iFBA is also able to determine systemic properties such as the growth rate from integrated network behavior rather than from empirical correlation with substrate uptake or other parameters, and this growth rate has a significant impact on the behavior of the ODE model. This aspect of iFBA was highlighted most dramatically in the gene perturbation study where we found 85 cases in which the ODE model incorrectly and the iFBA model correctly predicted the experimentally observed result of gene perturbation²⁶⁹. We found that these cases fell into three categories: (1) ODE model predicts lethality because it is missing an alternate pathway, (2) ODE model predicts viability because it does not account for global demands on biomass production, and (3) ODE model fails to predict the correct phenotype because the function of the gene is not included in the model. The iFBA modeling framework therefore adds to the predictive power of ODE-based models, both in terms of scope and accuracy.

In summary, the great advantage of flux balance models over traditional sets of ordinary differential equations is that they allow for analysis of the entire metabolic and regulatory networks. The advantage of the ODE models is that they capture intracellular concentrations and short time scale dynamics, which are critical components of signal transduction. When applied at the large scale, we find that the approach described above has the potential to incorporate the advantages of both perspectives.

There are several ways to potentially improve the iFBA framework. First, this model is based on an objective which maximizes the growth rate, and it has been shown that other objectives may be more accurate predictors of phenotype, depending on the growth conditions^{10,320,352}. Additionally, there are multiple flux distributions which could provide an equivalent growth rate, and only one of these has been selected for the simulation. Incorporating these equivalent distributions could also lead to a richer description of phenotype³⁹², and possibly also account for the natural phenotypic variation between cells in a culture. Finally, although we decided to initially focus on central metabolism for the purposes of developing iFBA, with this proof-of-principle in hand our iFBA model could be improved by including the 755 additional genes described in our more comprehensive rFBA model of *E. coli*⁷⁵.

Can we build on this approach to eventually create a whole-cell model of *E. coli*? So far the largest flux-balance model of *E. coli* incorporates 1,260 open reading frames corresponding to metabolism¹²⁰, and as aforementioned, another includes an additional 104 genes corresponding to transcriptional regulation⁷⁵. Our current work suggests that these large-scale metabolic and regulatory network models may now be thought of as a scaffold with which any ODE-based or other model that has an interface with metabolism may be integrated. This integration would allow processes which have been characterized and modeled in isolation to be re-evaluated in the context of their global effects. As more ODE-based models are developed and integrated into frameworks like that described here, it may eventually be possible to capture a majority of the known biological processes which occur in *E. coli* or other organisms for which a large-scale metabolic-regulatory reconstruction is available.

Chapter 4

A Whole-cell computational model predicts phenotype from genotype

Abstract

Understanding how complex phenotypes arise from individual molecules and their interactions is a primary challenge in biology that computational approaches are poised to tackle. We report a whole-cell computational model of the life cycle of the human pathogen *Mycoplasma genitalium* that includes all of its molecular components and their interactions. An integrative approach to modeling that combines diverse mathematics enabled the simultaneous inclusion of fundamentally different cellular processes and experimental measurements. Our whole-cell model accounts for all annotated gene functions and was validated against a broad range of data. The model provides insights into many previously unobserved cellular behaviors, including *in vivo* rates of protein-DNA association and an inverse relationship between the durations of DNA replication initiation and replication. In addition, experimental analysis directed by model predictions identified previously undetected kinetic parameters and biological functions. We conclude that comprehensive whole-cell models can be used to facilitate biological discovery.

4.1 Introduction

Computer models that can account for the integrated function of every gene in a cell have the potential to revolutionize biology and medicine, as they increasingly contribute to how we understand, discover, and design biological systems⁸². Models of biological processes have been increasing in complexity and scope^{75,167,401}, but with efforts at increased inclusiveness of genes, parameters, and molecular functions come a number of challenges.

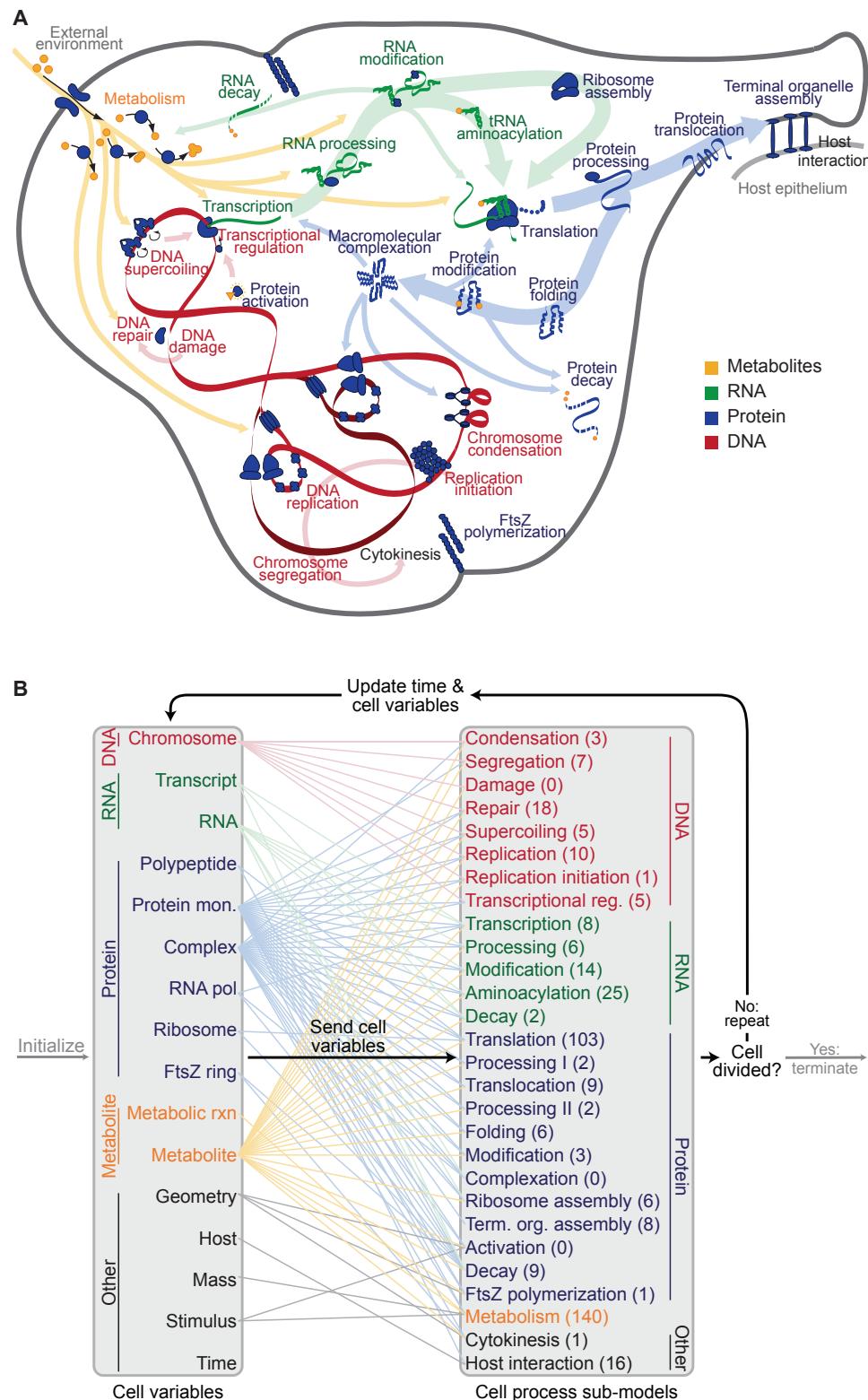


Figure 4.1. *M. genitalium* whole-cell model integrates 28 submodels of diverse cellular processes. (A) Diagram schematically depicts the 28 submodels as colored words—grouped by category as metabolic (orange), RNA (green), protein (blue), and DNA (red)—in the context of a single *M. genitalium* cell with its characteristic flask-like shape. Submodels are connected through common metabolites, RNA, protein, and the chromosome, which are depicted as orange, green, blue, and red arrows, respectively. (B) The model integrates cellular function submodels through 16 cell variables. First, simulations are randomly initialized to the beginning of the cell cycle (left gray arrow). Next, for each 1 s time step (dark black arrows), the submodels retrieve the current values of the cellular variables, calculate their contributions to the temporal evolution of the cell variables, and update the values of the cellular variables. This is repeated thousands of times during the course of each simulation. For clarity, cell functions and variables are grouped into five physiologic categories: DNA (red), RNA (green), protein (blue), metabolite (orange), and other (black). Colored lines between the variables and submodels indicate the cell variables predicted by each submodel. The number of genes associated with each submodel is indicated in parentheses. Finally, simulations are terminated upon cell division when the septum diameter equals zero (right gray arrow).

Two critical factors in particular have hindered the construction of comprehensive, “whole-cell” computational models. First, until recently, not enough has been known about the individual molecules and their interactions to completely model any one organism. The advent of genomics and other high-throughput measurement techniques has accelerated the characterization of some organisms to the extent that comprehensive modeling is now possible. For example, the mycoplasmas, a genus of bacteria with relatively small genomes that includes several pathogens, have recently been the subject of an exhaustive experimental effort by a European consortium to determine the transcriptome²²⁵, proteome²⁰³, and metabolome⁴⁴⁵ of these organisms.

The second limiting factor has been that no single computational method is sufficient to explain complex phenotypes in terms of molecular components and their interactions. The first approaches to modeling cellular physiology, based on ordinary differential equations (ODEs)^{11,43,52,53,98,406}, were limited by the difficulty in obtaining the necessary model parameters. Subsequently, alternative approaches were developed that require fewer parameters, including Boolean network modeling⁸⁸ and constraint-based modeling^{287,401}. However, the underlying assumptions of these methods do not apply to all cellular processes and conditions, and building a whole-cell model entirely based on either method is therefore impractical.

Here, we present a “whole-cell” model of the bacterium *Mycoplasma genitalium*, a human urogenital parasite whose genome contains 525 genes¹²². Our model attempts to: (1) describe the life cycle of a single cell from the level of individual molecules and their interactions; (2) account for the specific function of every annotated gene product; and (3) accurately predict a wide range of observable cellular behaviors.

4.2 Results

4.2.1 Whole-Cell Model Construction and Integration

Our approach to developing an integrative whole-cell model was to divide the total functionality of the cell into modules, model each independently of the others, and integrate these submodels together. We defined 28 modules (Figure 4.1A) and independently built, parameterized, and tested a submodel of each. Some biological processes have previously been studied quantitatively and in depth, whereas other processes are less well characterized or are hardly understood. Consequently, each module was modeled using the most appropriate mathematical representation. For example, metabolism was modeled using flux-balance analysis³⁸⁹, whereas RNA and protein degradation were modeled as Poisson processes.

A key challenge of the project was to integrate the 28 submodels into a unified model. Although we and others had previously developed methods to integrate ODEs with Boolean, probabilistic, and constraint-based submodels^{58,75–77}, the current effort involved so many different cellular functions and mathematical representations that a more general approach was needed. We began with the assumption that the submodels are approximately independent on short timescales (less than 1 s). Simulations are then performed by running through a loop in which the submodels are run independently at each time step but depend on the values of variables determined by the other submodels at the previous time step. Figure 4.1B summarizes the simulation algorithm and the relationships between the submodels and the cell variables. Appendix C provides a detailed description of the complete modeling process, including reconstruction and computational implementation.

4.2.2 Model Training and Parameter Reconciliation

Our model is based on a synthesis of over 900 publications and includes more than 1,900 experimentally observed parameters. Most of these parameters were implemented as originally reported. However, several other parameters were carefully reconciled; for example, the experimentally measured DNA content per cell^{155,260} represents less than one-third of the calculated mass of the mycoplasma chromosome. Appendix C details how we resolved this and several similar discrepancies among the experimentally observed parameters.

Once the model was implemented and all parameters were reconciled, we verified that the model

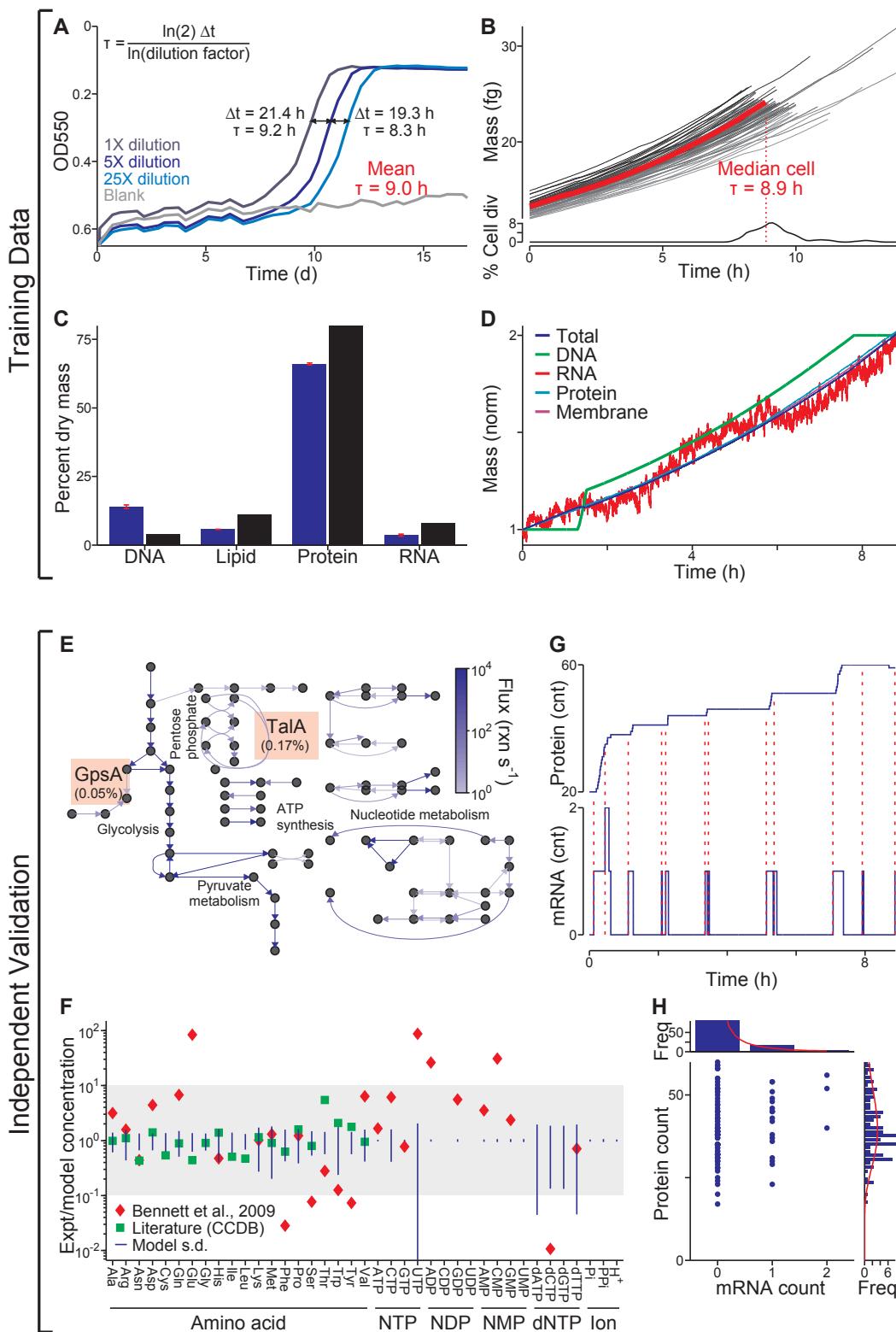


Figure 4.2. The model was trained with heterogeneous data and reproduces independent experimental data across multiple cellular functions and scales. **(A)** Growth of three cultures (dilutions indicated by shade of blue) and a blank control measured by OD550 of the pH indicator phenol red. The doubling time, t , was calculated using the equation at the top left from the additional time required by more dilute cultures to reach the same OD550 (black lines). **(B)** Predicted growth dynamics of one life cycle of a population of 64 in silico cells (randomly chosen from the total simulation set). Median cell is highlighted in red. Distribution of cell-cycle lengths is shown at bottom. **(C)** Comparison of the predicted and experimentally observed¹⁵⁵ cellular chemical compositions. Red bars indicate model SD; 155. did not report SD. **(D)** Temporal dynamics of the total cell mass and four cell mass fractions of a representative in silico cell. Mass fractions are normalized to their initial values. **(E)** Average predicted metabolic fluxes (see Figure 4.3B for metabolite and reaction labels). Arrow brightness indicates flux magnitude. The ratios of the GpsA and TalA fluxes to the Glk flux are indicated in orange boxes and are comparable to experimental data⁴⁴⁵. **(F)** Ratios of observed^{20,387} and average predicted concentrations of 39 metabolites. Blue bars indicate model SD. **(G)** Temporal dynamics of cytadherence high-molecular-weight protein 2 (HMW2, MG218) mRNA and protein expression of one in silico cell. Red dashed lines indicate the direct link between mRNA synthesis and subsequent bursts in protein synthesis. **(H)** HMW2 mRNA and protein copy number distribution of an unsynchronized population of 128 in silico cells. Histograms indicate the marginal distributions of the copy numbers of mRNA (top) and protein (right). Red lines indicate log-normal regressions of these marginal distributions. The absence of correlation between the copy numbers of mRNA and protein and the shapes of the marginal distributions is consistent with recent single-cell measurements³⁹⁵. See also Movie S1 and Tables S1 and S2.

recapitulates key features of our training data. We simulated 128 wild-type cells in a typical Mycoplasma culture environment, with each simulation predicting not only cellular properties such as the cell mass and growth rate but also molecular properties including the count, localization, and activity of each molecule (Movie S1 illustrates the life cycle of one in silico cell). We found that the model calculations were consistent with the observed doubling time (Figures 4.2A and 4.2B), cellular chemical composition (Figure 4.2C), replication of major cell mass fractions (Figure 4.2D), and gene expression ($R^2 = 0.68$; Figure 4.3A).

4.2.3 Model Validation against Independent Experimental Data

Next, we validated the model against a broad range of independent data sets that were not used to construct the model and which encompass multiple biological functions—metabolomics, transcriptomics, and proteomics—and scales from single cells to populations. In agreement with earlier reports⁴⁴⁵, the model predicts that the flux through glycolysis is >100-fold more than that through the pentose phosphate and lipid biosynthesis pathways (Figure 4.2E). Furthermore, the predicted metabolite concentrations are within an order of magnitude of concentrations measured in *Escherichia coli* for 100% of the metabolites in one compilation of data³⁸⁷ and for 70% in a more recent high-throughput study (Figure 4.2F)²⁰. Our model also predicts “burst-like” protein synthesis due to the local effect of intermittent messenger RNA (mRNA) expression and the global effect of

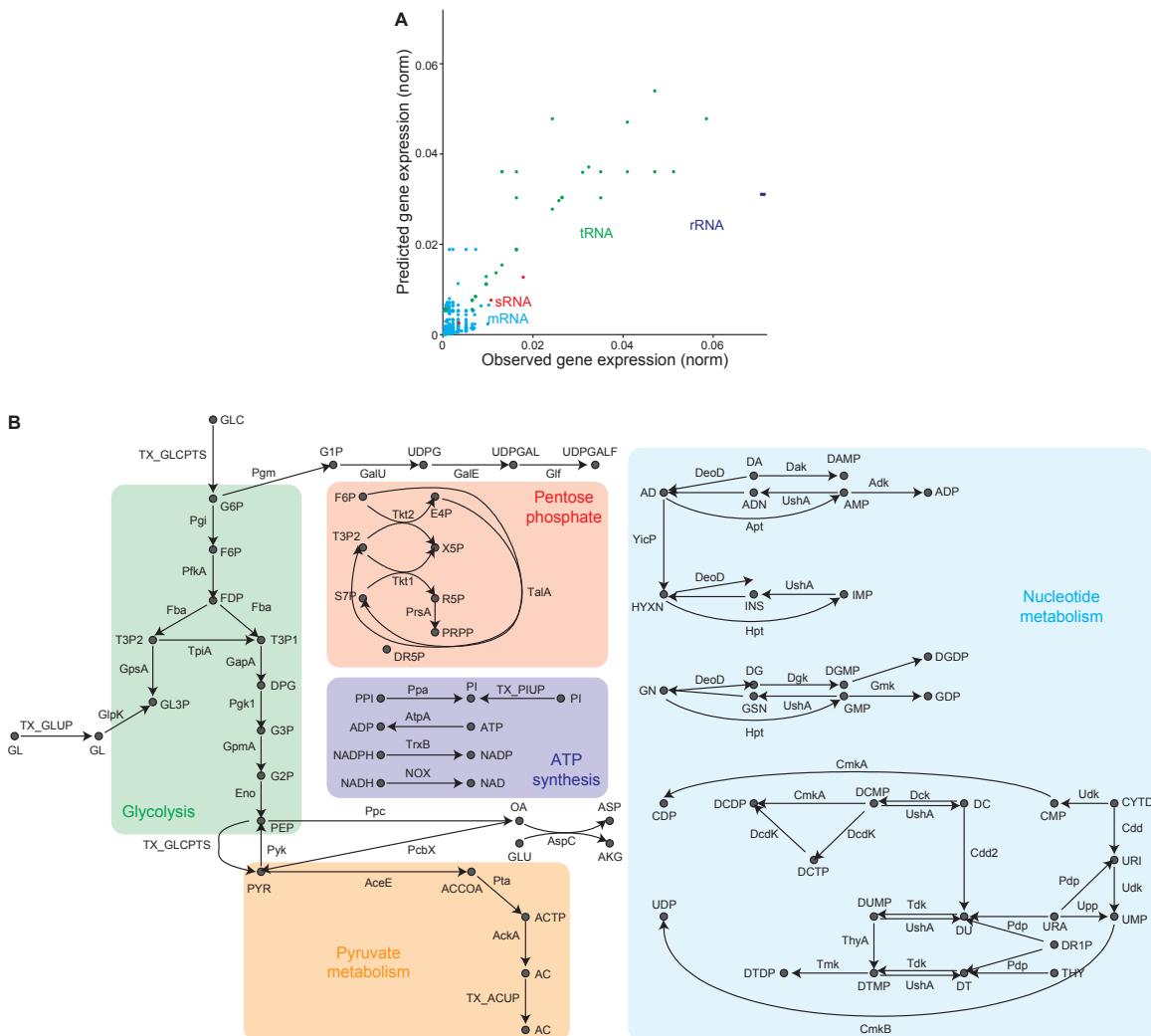


Figure 4.3. Model was trained using observed gene expression data, and it reproduces metabolic data. (A) Average predicted and experimentally observed ^{43}S RNA expression correlate strongly ($R^2 = 0.68$). Predicted and observed RNA expression are both normalized to the total RNA expression. (B) Labeled diagram of the *M. genitalium* metabolic network (Figure 4.2E).

stochastic protein degradation on the availability of free amino acids for translation, which is comparable to recent reports^{215,444} (Figure 4.2G). The mRNA and protein level distributions predicted by our model are also consistent with recently reported single-cell measurements (Figure 4.2H; compare to³⁹⁵). Taking all of these specific tests of the model’s predictions together, we concluded that our model recapitulates experimental data across multiple biological functions and scales.

4.2.4 Prediction of DNA-Binding Protein Interactions

Models are often used to predict molecular interactions that are difficult or prohibitive to investigate experimentally, and our model offers the opportunity to make such predictions in the context of the entire cell. Whereas previous studies have either focused on the genomic distribution of DNA-binding proteins³⁹³ or on the detailed diffusion dynamics of specific DNA-binding proteins³⁸, the whole-cell model can predict both the instantaneous protein chromosomal occupancy as well as the temporal dynamics and interactions of every DNA-binding protein at the genomic scale at single-cell resolution. Figure 4.4A illustrates the average predicted chromosomal protein occupancy as well as the predicted chromosomal occupancies for DNA and RNA polymerase and the replication initiator DnaA, which are three of the 30 DNA-binding proteins represented by our model. Consistent with a recent experimental study³⁹³, the predicted high-occupancy RNA polymerase regions correspond to highly transcribed ribosomal RNAs (rRNAs) and transfer RNAs (tRNAs). In contrast, the predicted DNA polymerase chromosomal occupancy is significantly lower and biased toward the terC (see below for further discussion).

The model further predicts that the chromosome is explored very rapidly, with 50% of the chromosome having been bound by at least one protein within the first 6 min of the cell cycle and 90% within the first 20 min (Figure 4.4B). RNA polymerase contributes the most to chromosomal exploration, binding 90% of the chromosome within the first 49 min of the cell cycle. On average, this results in expression of 90% of genes within the first 143 min (Figure 4.4C), with transcription lagging RNA polymerase exploration due to the significant contribution of nonspecific RNA polymerase-DNA interactions to RNA polymerase diffusion⁴⁴⁰.

The model also predicts protein-protein collisions on the chromosome. Previous researchers have studied the collisions of pairs of specific proteins³³⁷, but experimentally determining the collisions among all pairs of DNA-binding proteins at the genomic scale at single-cell resolution is currently infeasible. Our model predicts that over 30,000 collisions occur on average per cell cycle, leading

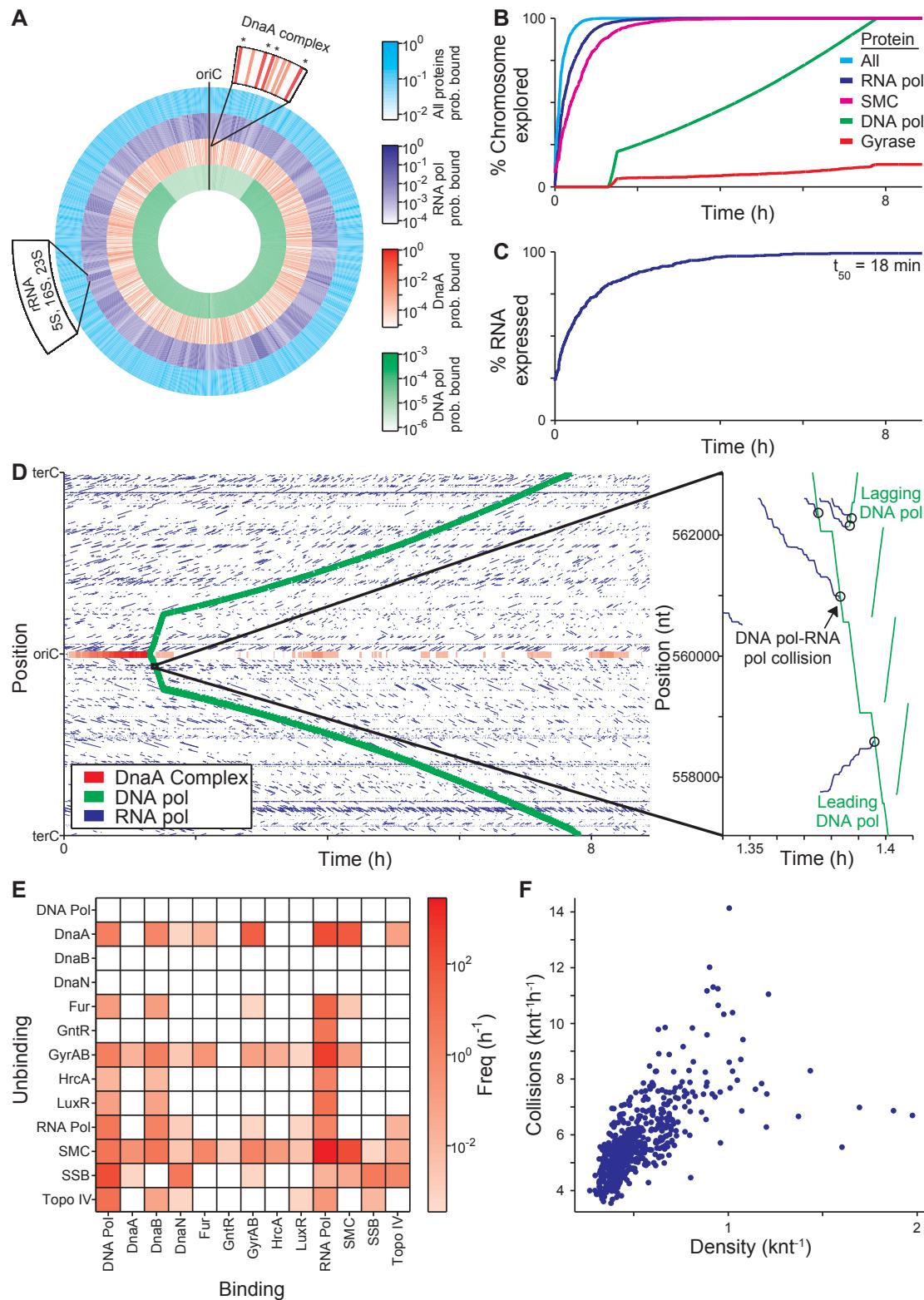


Figure 4.4. The model highlights the central physiological role of dna-protein interactions. (A) Average density of all DNA-bound proteins and of the replication initiation protein DnaA and DNA and RNA polymerase of a population of 128 *in silico* cells. Top magnification indicates the average density of DnaA at several sites near the oriC; DnaA forms a large multimeric complex at the sites indicated with asterisks, recruiting DNA polymerase to the oriC to initiate replication. Bottom left indicates the location of the highly expressed rRNA genes. (B and C) Percentage of the chromosome that is predicted to have been bound (B) and the number of genes that are predicted to have been expressed (C) as functions of time. SMC is an abbreviation for the name of the chromosome partition protein (MG298). (D) DNA-binding and dissociation dynamics of the oriC DnaA complex (red) and of RNA (blue) and DNA (green) polymerases for one *in silico* cell. The oriC DnaA complex recruits DNA polymerase to the oriC to initiate replication, which in turn dissolves the oriC DnaA complex. RNA polymerase traces (blue line segments) indicate individual transcription events. The height, length, and slope of each trace represent the transcript length, transcription duration, and transcript elongation rate, respectively. The inset highlights several predicted collisions between DNA and RNA polymerases that lead to the displacement of RNA polymerases and incomplete transcripts. (E) Predicted collision and displacement frequencies for pairs of DNA-binding proteins. (F) Correlation between DNA-binding protein density and frequency of collisions across the chromosome. Both (E) and (F) are based on 128 cell-cycle simulations.

to the displacement of 0.93 proteins per second. Figure 4.4D illustrates the binding dynamics of the same proteins depicted in Figure 4.4A over the course of the cell cycle for one representative simulation and highlights several protein-protein collisions. Further categorization of the predicted collisions by chromosomal location indicates that the frequency of protein-protein collisions correlates strongly with DNA-bound protein density across the genome (Figure 4.4F) and that the majority of collisions are caused by RNA polymerase (84%) and DNA polymerase (8%), most commonly resulting in the displacement of structural maintenance of chromosome (SMC) proteins (70%) or single-stranded binding proteins (6%) (Figure 4.4E and Table S2F).

4.2.5 Identification of Metabolism as an Emergent Cell-Cycle Regulator

The model can also highlight interesting aspects of cell behavior. In reviewing our model simulations, we noticed variability in the cell-cycle duration (Figure 4.2B) and wanted to determine the source of that variability. The model representation of the *M. genitalium* cell cycle consists of three stages: replication initiation, replication itself, and cytokinesis. We found that there was relatively more cell-to-cell variation in the durations of the replication initiation (64.3%) and replication (38.5%) stages than in cytokinesis (4.4%) or the overall cell cycle (9.4%; Figure 4.5A). This data raised two questions: (1) what is the source of duration variability in the initiation and replication phases; and (2) why is the overall cell-cycle duration less varied than either of these phases?

With respect to the first question, replication initiation occurs as DnaA protein monomers bind or unbind stochastically and cooperatively to form a multimeric complex at the replication origin

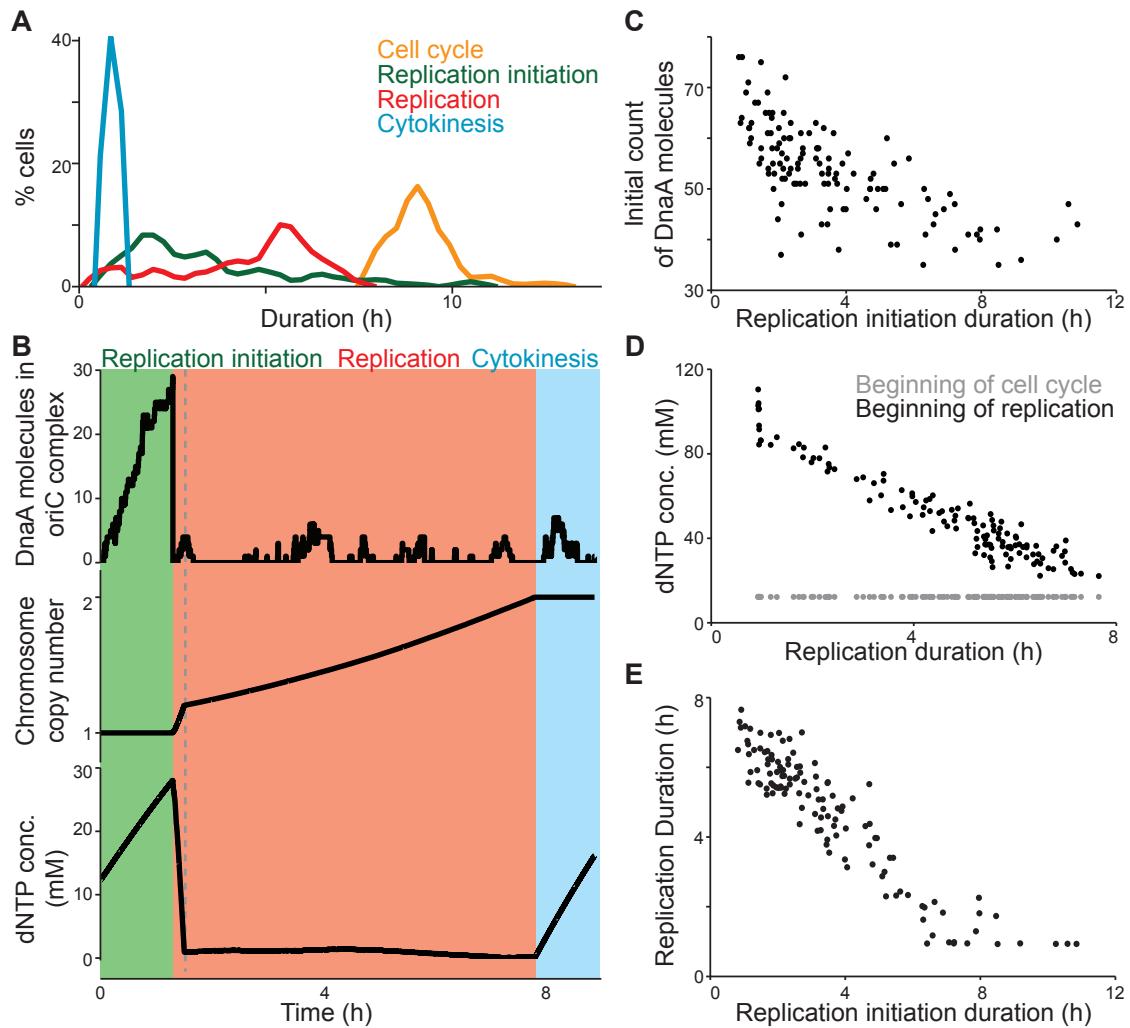


Figure 4.5. The model predictions regarding regulation of the cell-cycle duration. **(A)** Distributions of the duration of three cell-cycle phases, as well as that of the total cell-cycle length, across 128 simulations. **(B)** Dynamics of macromolecule abundance in a selected cell simulation. Top, the size of the DnaA complex assembling at the oriC (in monomers of DnaA); middle, the copy number of the chromosome; and bottom, the cytosolic dNTP concentration. The quantities of these macromolecules correlate strongly with the timing of key cell-cycle stages. **(C)** Correlation between the initial cellular DnaA content and the duration of the replication initiation cell-cycle stage across the same 128 in silico cells depicted in (A). **(D)** Correlation between the dNTP concentrations (both at the beginning of the cell cycle and at the beginning of replication) and the duration of replication across the same 128 in silico cells depicted in (A). **(E)** Correlation between the duration of replication initiation and replication across the same 128 in silico cells depicted in (A).

(Figure 4.5B, top)⁴³. When the complex is complete, DNA polymerase gains access to the origin, and the complex is displaced. We found a correlation ($R^2 = 0.49$) between the predicted duration of replication initiation and the initial number of free DnaA monomers (Figure 4.5C); however, the low correlation indicated that the duration depends on more than the initial conditions. In particular, we observed that the stochastic aspect of the transcription and translation submodels creates variability in the number of new DnaA monomers produced over time, as well as the DnaA-binding and -unbinding events themselves. This indicates that the variability in replication initiation duration depends not only on variability in initial conditions but also in the simulation itself.

As to the second question, because the replication submodel is substantially more deterministic than the initiation submodel, we expected to find a straightforward relationship between the progress of replication and the cell cycle. Instead, the model predicts that DNA replication proceeds at two distinct rates during the cell cycle. This is reflected in the motion and DNA-binding density of DNA polymerase (Figures 4.4A and 4.4D) and in the dynamics of DNA synthesis as compared to the synthesis of other macromolecules (Figure 4.5B, middle). Initially, replication proceeds quickly due to the free deoxyribonucleotide triphosphate (dNTP) content in the cell (Figure 4.5B, bottom). When DNA polymerase initially binds to the replication origin, dNTPs are abundant, and replication proceeds unimpeded. When the dNTP pool is exhausted, however, the rate of replication slows to the rate of dNTP synthesis. Accordingly, the duration of the replication phase in individual cells is more closely related to the free dNTP content at the start of replication than to the dNTP content at the start of the cell cycle (Figure 4.5D).

This change in the availability of dNTPs imposes a control on the cell-cycle duration. Specifically, the duration of the initiation and replication phases is inversely related to each other in single cells (Figure 4.5E), such that longer initiation times led to shorter replication times. This occurs because cells that require extra time to initiate replication also build up a large dNTP surplus, leading to faster replication. This interplay buffers against the high variability in the duration of replication initiation, giving rise to substantially less variability in the length of the cell cycle. The whole-cell model therefore presents a hypothesis of an emergent control of cell-cycle duration that is independent of genetic regulation.

4.2.6 Global Distribution of Energy

The model also provided an opportunity to develop a quantitative assessment of cellular energetics, which represents one of the most connected aspects of our model. To begin, we investigated the synthesis dynamics of the high-energy intermediates ATP, GTP, FAD(H₂), NAD(H), and NADP(H) and found that ATP and guanosine triphosphate (GTP) are synthesized at rates greater than 1,000-fold higher than the others (Figure 4.6A). Notably, the overall usage of ATP and GTP did not vary considerably in all but the very slowest of our simulations (Figure 4.6B), underscoring the role of metabolism in controlling the cell-cycle length. We then considered the processes that use ATP and GTP and found that usage is dominated by production of mRNA and protein (Figure 4.6C). We also found a large (44%) discrepancy between total energy usage and production (Figure 4.6D). Others have noted an uncoupling between catabolism and anabolism, attributing the difference to factors such as varying maintenance costs or energy spilling via futile cycles¹⁶⁶, and the model's prediction estimates the total energy cost of such uncoupling.

4.2.7 Determining the Molecular Pathologies of Single-Gene Disruption Phenotypes

Having considered these above-described model predictions for the wild-type *M. genitalium* strain, we next performed in silico genome perturbations to gain insight into the genetic requirements of cellular life. We performed multiple simulations of each of the 525 possible single-gene disruption strains (over 3,000 total simulations) and found that 284 genes are essential to sustain *M. genitalium* growth and division and that 117 are nonessential. The model accounts for previously observed gene essentiality with 79% accuracy ($p < 10^{-7}$ ¹⁶⁹; Figure 4.7A).

In cases in which the model prediction agrees with the experimental outcome with respect to gene essentiality, we found that a deeper examination of the simulation can generate insight into why the gene product is required by the system. We examined the capacities of the 525 simulated gene disruption strains to produce major biomass components (RNA, DNA, protein, and lipid) and to divide. As shown in Figure 4.7B, the nonviable strains were unable to adequately perform one or more of these major functions. The most debilitating disruptions involved metabolic genes and resulted in the inability to produce any of the major cell mass components. The next most debilitating gene disruptions impacted the synthesis of a specific cell mass component, such as RNA or protein. Interestingly, in these cases, the model predicted an initial phase of near-normal growth

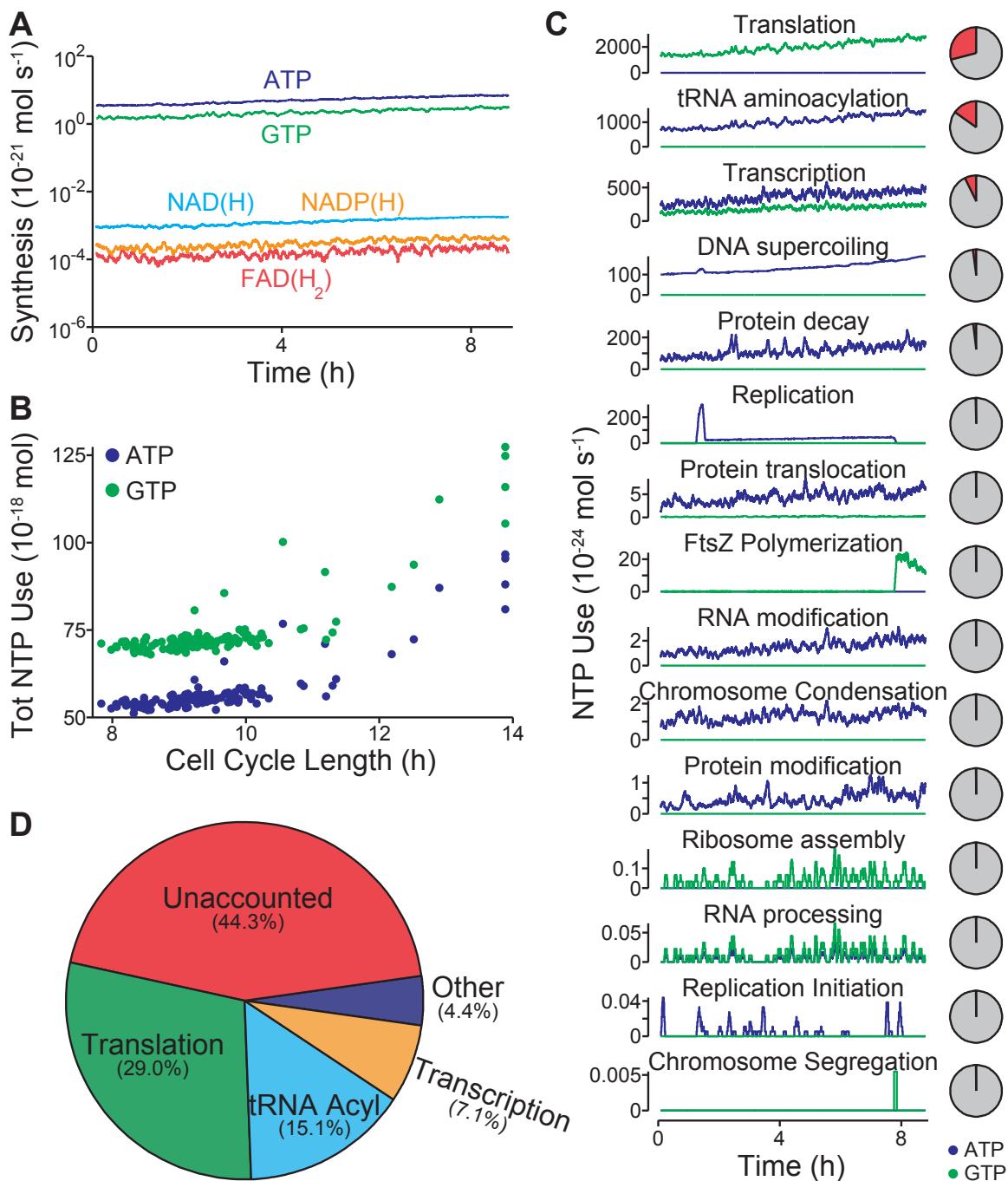


Figure 4.6. Model provides a global analysis of the use and allocation of energy. (A) Intracellular concentrations of the energy carriers ATP, GTP, FAD(H_2), NAD(H), and NADP(H) of one *in silico* cell. (B) Comparison of the cell-cycle length and total ATP and GTP usage of 128 *in silico* cells. (C) ATP (blue) and GTP (green) usage of 15 cellular processes throughout the life cycle of one *in silico* cell. The pie charts at right denote the percentage of ATP and GTP usage (red) as a fraction of total usage. (D) Average distribution of ATP and GTP usage among all modeled cellular processes in a population of 128 *in silico* cells. In total, the modeled processes account for only 44.3% of the amount of energy that has been experimentally observed to be produced during cellular growth.

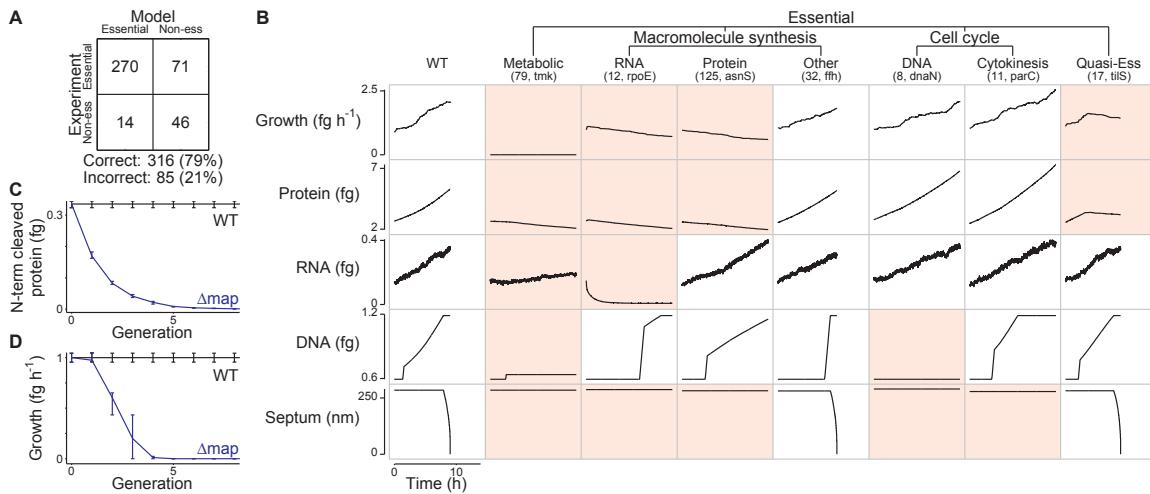


Figure 4.7. Model identifies common molecular pathologies underlying single-gene disruption phenotypes. (A) Comparison of predicted and observed¹⁶⁹ gene essentiality. Model predictions are based on at least five simulations of each single-gene disruption strain; see Appendix C for details. (B) Single-gene disruption strains were grouped into phenotypic classes (columns) according to their capacity to grow, synthesize protein, RNA, and DNA, and divide (indicated by septum length). Each column depicts the temporal dynamics of one representative in silico cell of each essential disruption strain class. Disruption strains of nonessential genes are not shown. Dynamics significantly different from wild-type are highlighted in red. The identity of the representative cell and the number of disruption strains in each category are indicated in parenthesis. (C and D) Degradation and dilution of N-terminal protein content (C) of methionine aminopeptidase (map, MG172) disrupted cells causes reduced growth (D). Blue and black lines indicate the map disruption and wild-type strains, respectively. Bars indicate SD. See also Figure 4.8 for the distribution of simulated growth rates.

followed by decreasing growth due to diminishing protein content. In some cases (Figure 4.7B, fifth column), the time required for the levels of specific proteins to fall to lethal levels was greater than one generation (Figures 4.7C and 4.7D). A third class of lethal gene disruptions impaired cell-cycle processes. For these, the model predicted normal growth rates and metabolism, but it also predicted incapacity to complete the cell cycle. The remaining lethal gene disruption strains grew so slowly compared to wild-type that they were considered nonviable (Figures 4.7B and 4.8). We conclude that the model can be used to classify cellular phenotypes by their underlying molecular interactions.

4.2.8 Model-Driven Biological Discovery

Using computational modeling as a complement to an experimental program has previously been shown to facilitate biological discovery⁸². This is often accomplished by reconciling model predictions that are initially inconsistent with observations⁷⁵. To test the utility of the whole-cell model in this context, we experimentally measured the growth rates of 12 single-gene disruption strains—ten of

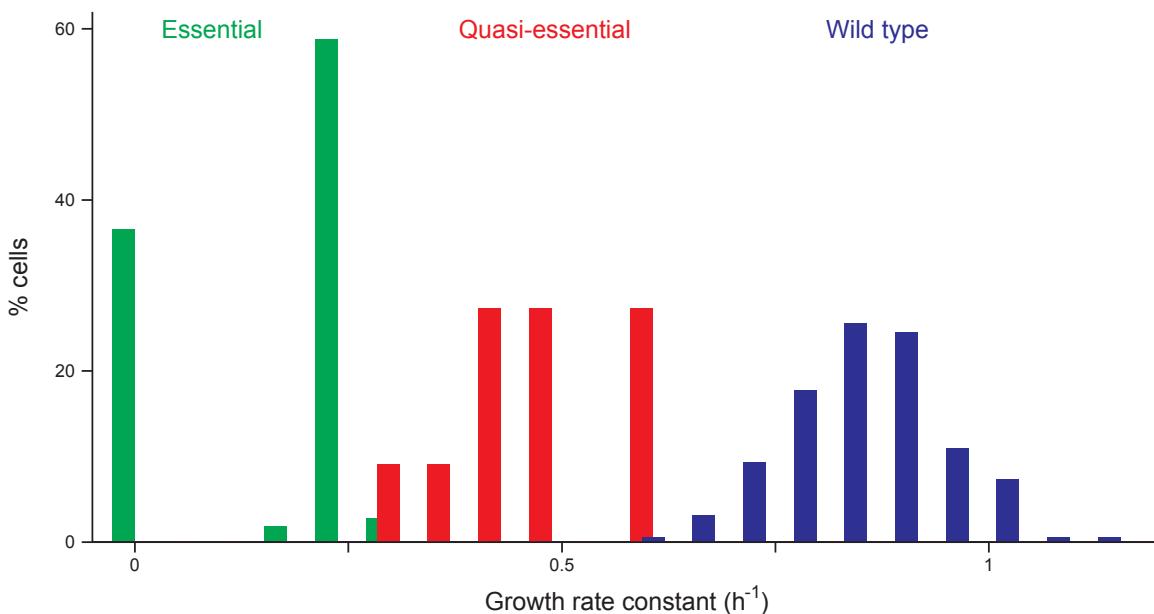
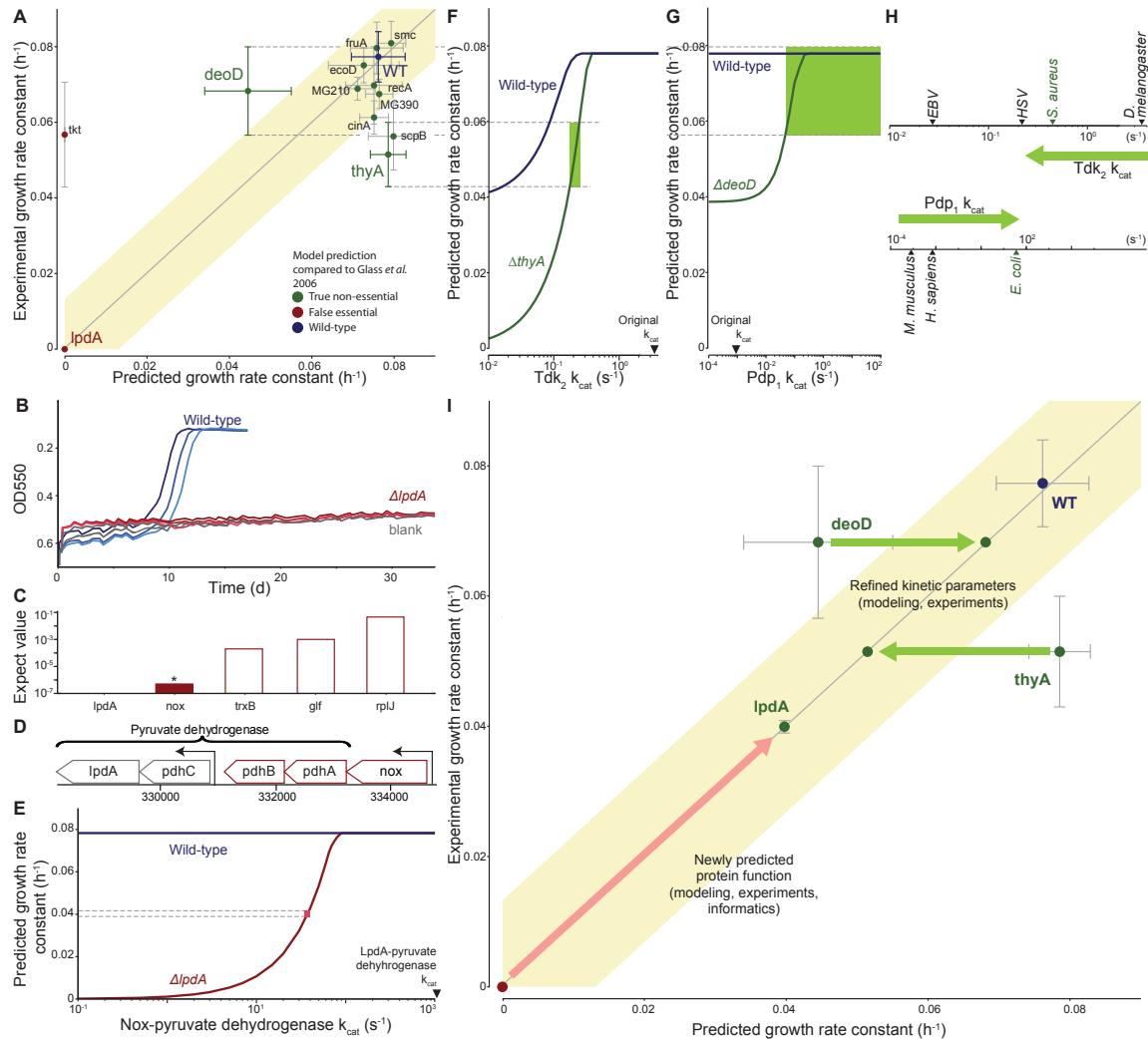


Figure 4.8. Quasiessential and essential single-gene disruption strains exhibit significantly reduced growth. In silico growth rate distribution of wild-type *M. genitalium* and of the quasi-essential and essential single-gene disruption strains. Genes were classified as quasi-essential or essential if the growth rate of the corresponding disruption strain was significantly less than that of wild-type *M. genitalium*. Genes were subclassified as essential if the growth rate of the corresponding disruption strain was zero or if the strain exhibited significantly reduced DNA, RNA, or protein production or did not undergo cell division; genes were subclassified as quasiessential if the corresponding disruption strain grew slowly, but otherwise exhibited normal macromolecular production and cell division. See Appendix C for further discussion of the classification of single-gene disruption strains.

which were correctly predicted to be viable and two of which were incorrectly predicted to be nonviable—for comparison to our model’s predictions (Figure 4.9A). We found that two-thirds of the predictions were consistent with the measured growth rates.

The most interesting of these comparisons concerned the *lpdA* disruption strain. The *lpdA* gene was originally determined to be nonessential¹⁶⁹. Consequently, we initially classified the model’s prediction as false (Figure 4.7A). However, we did not detect growth using our colorimetric assay (Figure 4.9B), which was a discrepancy that warranted further investigation. An alternative method to determine the doubling time yielded a value that was 40% lower than the wild-type (Table S1). Taken together, the data suggested that disrupting the *lpdA* gene had a severe but noncritical impact on cell growth.

In an effort to resolve the discrepancy between our model and the experimental measurements, we determined the molecular pathology of the *lpdA* disruption strain. The *lpdA* gene product is part



of the pyruvate dehydrogenase complex, which catalyzes the transfer of electrons to nicotinamide adenine dinucleotide (NAD) as a subset of the overall pyruvate dehydrogenase chemical reaction¹⁹³. The viability of the *lpdA* disruption strain suggests that this reaction could be catalyzed by another enzyme with a lower catalytic efficiency.

Because previous studies have shown that many *M. genitalium* genes are multifunctional^{74,314}, we searched the genome for candidates encoding an alternative NAD electron transfer pathway. We found that the Nox sequence was far more similar to the LpdA sequence than any other gene product in the genome, with 61% coverage, 25% identity, and an expectation value of less than 10^{-6} (Figure 4.9C). Furthermore, the *nox* gene product, NADH oxidase, has been shown to oxidize NAD³⁴⁴.

Figure 4.9. Quantitative characterization of selected gene disruption strains leads to identification of novel gene functions and kinetic parameters. (A) Comparison of measured and predicted growth rates for wild-type and 12 single-gene disrupted strains. Model predictions that fall within the shaded region were considered consistent with experimental observations; the region has a width of four times the SD of the wild-type strain growth measurement. Horizontal and vertical bars indicate predicted and observed SD. (B) Growth curves for the wild-type and *lpdA* gene disruption strains and blank, similar to Figure 4.2A. (C) Expectation values determined by performing a pBLAST search of the *M. genitalium* genome with the LpdA sequence as a query. The asterisk and colored bar indicate a significant match ($E < 10^{-6}$). (D) Detail of the *M. genitalium* genome. The pyruvate dehydrogenase complex genes are indicated by the top bracket, and transcription units identified in *M. pneumoniae*²²⁵ are indicated by arrows. The transcription unit including *nox* is highlighted in color. (E) Allowing Nox to partially replace LpdA in pyruvate dehydrogenase reconciles model predictions and experimental observations. The blue and red lines represent the predicted wild-type and $\Delta lpdA$ strain growth rates as a function of the Nox-pyruvate dehydrogenase k_{cat} . The pink box indicates the k_{cat} at which the model predictions are consistent with both the wild-type and $\Delta lpdA$ strain experimentally measured growth rates. (F and G) Diagnosing the discrepancy between predictions and experiment for the *thyA* (F) and *deoD* (G) gene disruption strains. Some of the functionalities of ThyA and DeoD can be replaced by the enzymes Tdk and Pdp, respectively. The predicted growth rates of the wild-type and gene disruption strains depend on the k_{cat} of these enzymes. The green region highlights the range of k_{cat} values that are consistent with the measured growth rates of both the wild-type and gene disruption strain. (H) Newly predicted k_{cat} values are similar to values that were measured in closely related organisms. Measured values of k_{cat} for Tdk (top) and Pdp (bottom) are shown; green arrow indicates the initial and revised k_{cat} values. The nearest *M. genitalium* relative is highlighted in green. (I) Model-based biological discovery. Comparison of model predictions to experimental measurements identified gene disruption strains of particular interest, including the *lpdA*, *deoD*, and *thyA* disruption strains. Further investigation—using a combination of experiments, modeling, and/or informatics—led to new and more consistent measurements and predictions. Most importantly, the higher consistency reflected novel insights into *M. genitalium* biology. The arrows (red for *lpdA*, green for *deoD* and *thyA*) indicate the shift from lower to higher consistency between model and experiment, and each arrow is annotated with the new biological insight and the supporting evidence in parentheses. The overall graph format is the same as in Figure 4.9A. Horizontal and vertical bars indicate predicted and observed SD.

Moreover, the *nox* locus falls in a suboperon that contains two other pyruvate dehydrogenase genes and has been shown to be coexpressed with *pdhA*²²⁵ (Figure 4.9D), strongly suggesting a functional relationship between the products of these two genes. Our model suggests that, to reproduce the observed growth rate in the absence of *lpdA*, the hypothetical Nox-dependent reaction would require a k_{cat} of $\tilde{50} \text{ s}^{-1}$ (Figure 4.9E), which represents only 5% of the maximum throughput of this enzyme. We therefore concluded that substrate promiscuity of Nox is likely to enable the *lpdA* disruption strain to survive.

Four gene disruption strains exhibited growth rates that were quantitatively different than those predicted by the model (Figure 4.9A); of these, we used the complete simulations for the *thyA* and *deoD* strains to determine the underlying pathology of the respective gene disruptions. The *thyA* gene product catalyzes thymidine monophosphate (dTMP) production and can be complemented by the *tdk* gene product. We therefore hypothesized that, by reducing the k_{cat} value for Tdk in

the model, we would see a reduction in the growth rate of the *tdk* disruption strain. Reducing the Tdk k_{cat} in the model did indeed reduce the predicted growth rate of the *thyA* strain, but it also affected the wild-type growth rate (Figure 4.9F). Only a small range of the k_{cat} values both reduced the *thyA* strain growth rate to the experimentally observed levels and was also consistent with the wild-type growth rate.

In a similar case, purine nucleoside phosphorylase (DeoD) catalyzes the conversion of deoxyadenosine to adenine and D-ribose-1-phosphate; these products can also be produced by the pdp gene product from deoxyuridine. We identified a Pdp k_{cat} range for which the wild-type and *deoD* gene disruption strains produce the same growth rate (Figure 4.9G).

Significantly, these newly predicted k_{cat} values are consistent with previously reported values. In the original model reconstruction, to least constrain the metabolic model, we conservatively set each of these k_{cats} to the least restrictive value found during the reconstruction process. For Tdk and Pdp, these values corresponded to distantly related organisms; however, the newly predicted k_{cat} values are consistent with reports from more closely related species (Figure 4.9H).

In each of these three cases (*lpdA*, *deoD*, and *thyA*), identifying a discrepancy between model predictions and experimental measurements led to further analysis, which resolved the discrepancy and also provided insight into *M. genitalium* biology (Figure 4.9I). These results support the assertion that large-scale modeling can be used to guide biological discovery^{40,146}.

4.3 Discussion

We have developed a comprehensive whole-cell model that accounts for all of the annotated gene functions identified in *M. genitalium* and explains a variety of emergent behaviors in terms of molecular interactions. Our model accurately recapitulates a broad set of experimental data, provides insight into several biological processes for which experimental assessment is not readily feasible, and enables the rapid identification of gene functions as well as specific cellular parameters.

In contemplating these results, we make two observations based on comparing this work in whole-cell modeling with earlier work in whole-genome sequencing. First, similar to the first reports of the human genome sequence, the model presented here is a “first draft,” and extensive effort is required before the model can be considered complete. Of course, much of this effort will be experimental (for example, further characterization of gene products), but the technical and modeling aspects of

this study will also have to be expanded, updated, and improved as new knowledge comes to light.

Second, in whole-genome sequencing as well as in whole-cell modeling, *M. genitalium* was a focus of initial studies, primarily because of its small genome size. The goal of our modeling efforts, as well as that of early sequencing projects, was to develop the technology in a reduced system before proceeding to more complex organisms. However, *M. genitalium* presents many challenges with regard to experimental tractability. Resistance to most antibiotics, the lack of a chemically defined medium, and a cell size that requires advanced microscopy techniques for visualization all greatly limit the range of experimental techniques available to study this organism. As a result, much of the data used to build and validate the model were obtained from other organisms. Therefore, although the results we report suggest several experiments that could yield important insight with respect to *M. genitalium* function, comprehensive validation of our approach will require modeling more experimentally tractable organisms such as *E. coli*.

We are optimistic that whole-cell models will accelerate biological discovery and bioengineering by facilitating experimental design and interpretation. Moreover, these findings, in combination with the recent de novo synthesis of the *M. genitalium* chromosome and successful genome transplantation of Mycoplasma genomes to produce a synthetic cell^{131,132,208,209}, raise the exciting possibility of using whole-cell models to enable computer-aided rational design of novel microorganisms. Finally, we anticipate that the construction of whole-cell models and the iterative testing of them against experimental information will enable the scientific community to assess how well we understand integrated cellular systems.

4.4 Experimental Procedures

4.4.1 Reconstruction

The whole-cell model was based on a detailed reconstruction of *M. genitalium* that was developed from over 900 primary sources, reviews, books, and databases. First, we reconstructed the organization of the chromosome, including the locations of each gene, transcription unit, promoter, and protein-binding site. Second, we functionally annotated each gene, beginning with the Comprehensive Microbial Resource (CMR) annotation. Functional annotation was primarily based on homologs identified by bidirectional best BLAST. To fill gaps in the reconstructed organism and to maximize the scope of the model, we expanded and refined each gene's annotation using primary research

articles and reviews (see Appendix C and Table S3). Third, we curated the structure of each gene product, including the posttranscriptional and posttranslational processing and modification of each RNA and protein and the subunit composition of each protein and ribonucleoprotein complex. After annotating each gene, we categorized the genes into 28 cellular processes. We curated the chemical reactions of each cellular process. The reconstruction was stored in an MySQL relational database. See Appendix C and Table S3 for further discussion of the reconstruction.

4.4.2 Cellular Process Submodels

Because biological systems are modular, cells can be modeled by the following: (1) dividing cells into functional processes; (2) independently modeling each process on a short timescale; and (3) integrating process submodels at longer timescales. We divided *M. genitalium* into the 28 functional processes illustrated in Figure 4.1 and modeled each process independently on a 1 s timescale using different mathematics and different experimental data. The submodels spanned six areas of cell biology: (1) transport and metabolism; (2) DNA replication and maintenance; (3) RNA synthesis and maturation; (4) protein synthesis and maturation; (5) cytokinesis; and (6) host interaction. Submodels were implemented as separate classes. See Appendix C for further discussion of each submodel.

4.4.3 Submodel Integration

We integrated the submodels in three steps. First, we structurally integrated the process submodels by linking their common inputs and outputs through 16 cell variables (shown in Figure 4.1), which together represent the complete configuration of the modeled cell: (1) metabolite, RNA, and protein copy numbers; (2) metabolic reaction fluxes; (3) nascent DNA, RNA, and protein polymers; (4) molecular machines; (5) cell mass, volume, and shape; (6) the external environment, including the host urogenital epithelium; and (7) time. Second, the common inputs to the submodels were computationally allocated at the beginning of each time step. Third, we refined the values of the submodel parameters to make the submodels mutually consistent. See Appendix C for further discussion.

4.4.4 Simulation Algorithm

The whole-cell model is simulated using an algorithm comparable to those used to numerically integrate ODEs. First, the cell variables are initialized. Second, the temporal evolution of the cell state is calculated on a 1 s timescale by repeatedly allocating the cell variables among the processes, executing each of the cellular process submodels, and updating the values of the cell variables. Finally, the simulation terminates when either the cell divides or the time reaches a predefined maximum value. See Appendix C for further discussion.

4.4.5 Single-Gene Disruptions

Single-gene disruptions were modeled by (1) initializing the cell variables, (2) deleting the in silico gene, and (3) calculating the temporal evolution of the cell state for the first generation postdisruption. We also calculated the mean growth rate of each single-gene disruption strain at successive generations postdisruption. See Appendix C for further discussion of the implementation of disruption strains and their computational analysis.

4.4.6 Computational Simulation and Analysis

We used the whole-cell model to simulate 192 wild-type cells and 3,011 single-gene deletants. All simulations were performed with MATLAB R2010b on a 128 core Linux cluster. The predicted dynamics of each cell were logged at each time point and subsequently analyzed using MATLAB. See Appendix C for further discussion.

4.4.7 Bacterial Culture

M. genitalium wild-type and mutant strains with single-gene disruptions by transposon insertion¹⁶⁹ were grown in Spiroplasma SP-4 culture media at 37°C and 5% CO₂. Growth was detected using the phenol red pH indicator. Cells were harvested for quantitative growth measurement at pH 6.3-6.7. See Appendix C for more information about media and culture conditions.

4.4.8 Colorimetric Assay to Measure Cell Growth

To measure the growth rates of the wild-type and mutant strains, cells were collected from 10 cm plate cultures at pH 6.3-6.7, resuspended in 3 ml of fetal bovine serum (FBS), and serial filtered

through 1.2, 0.8, 0.45, and $0.2\text{ }\mu\text{m}$ polyethersulfone filters to sterilize and separate individual cells. Cells were then plated at 5-, 25-, and 125-fold serial dilutions in triplicate on a 96-well plate and incubated at 37°C and 5% CO_2 . Six wells per plate were filled with blank SP-4 phenol red media as a negative control. Optical density readings were taken twice a day at 550 nm to measure the decrease in phenol red color as pH decreased. Growth rate constants were calculated from the additional time required for consecutive dilutions to reach the same OD₅₅₀ value and were averaged over two to three independent sets of three replicates. See Appendix C for further description of these calculations. We used a heteroscedastic two-sample two-tailed t test to determine whether the doubling time of each single-gene disruption strain differed significantly from that of the wild-type. The growth rates of several slow-growing strains were also measured by DNA quantification using a modified version of the procedure described in¹⁶⁹. See Appendix C for further discussion.

4.4.9 Source Code

The model source code, training data, and results are freely available at SimTK (<http://simtk.org/home/wholecell>).

Chapter 5

WholeCellViz: data visualization for whole-cell models

Abstract

Background: Whole-cell models promise to accelerate biomedical science and engineering. However, discovering new biology from whole-cell models and other high-throughput technologies requires novel tools for exploring and analyzing complex, high-dimensional data.

Results: We developed WholeCellViz, a web-based software program for visually exploring and analyzing whole-cell simulations. WholeCellViz provides 14 animated visualizations, including metabolic and chromosome maps. These visualizations help researchers analyze model predictions by displaying predictions in their biological context. Furthermore, WholeCellViz enables researchers to compare predictions within and across simulations by allowing users to simultaneously display multiple visualizations.

Conclusion: WholeCellViz was designed to facilitate exploration, analysis, and communication of whole-cell model data. Taken together, WholeCellViz helps researchers use whole-cell model simulations to drive advances in biology and bioengineering.

5.1 Background

Whole-cell computational models promise to predict how complex cellular behaviors such as growth and replication arise from individual molecules and their interactions. Recently, we developed the first whole-cell model of a single cell, the Gram-positive bacterium *Mycoplasma genitalium*¹⁷⁵. The model predicts the dynamics of every molecular species over the entire cell cycle, accounting for the specific function of every annotated gene product. The model's simulations produce rich data containing valuable insights into cellular behavior. For example, the model's simulations have generated

new insights into cell cycle regulation, energy usage, and gene essentiality¹⁷⁵.

However, the large number of whole-cell model predictions – over 50 billion data points in a typical dataset – makes directly analyzing the predictions time consuming and cumbersome. Furthermore, directly analyzing the model’s predictions requires deep knowledge of mathematical modeling, computer programming and the unique data structures used to represent the model’s predictions.

Data visualization software is critically needed to help researchers realize the full potential of whole-cell models by enabling researchers to more quickly and efficiently analyze whole-cell model simulations. We developed WholeCellViz to enable researchers to easily visualize whole-cell model predictions. WholeCellViz provides researchers interactive animations as well as time series plots to easily explore whole-cell model predictions. Furthermore, WholeCellViz facilitates comparisons within and across simulations by enabling researchers to view grids of animations and plots.

Interactive data visualization is becoming increasingly important as biological data continues to grow in complexity and volume. Data visualization can help scientists identify subtle patterns in large data sets leading to important scientific findings. For example, Lum et al. used Iris to visualize genetic data from 272 breast cancer patients³¹⁸. Iris revealed a specific genetic profile for women with low estrogen receptor expression, but high survival rates, a group which now receives targeted treatment for breast cancer. Shannon et al. used Cytoscape to visually link biomolecular networks with high-throughput data on various molecular states and functional annotations²⁹⁰. Baliga et al. used Cytoscape to obtain a systems-level understanding of Halobacterium energy transduction by visualizing its protein interaction network²⁸⁶. Pathway Tools enables researchers to visually integrate genomic, proteomic, and metabolomic data²⁹⁵. Chang et al. and Paley et al. used the Pathway Tools Omics Viewer to investigate the role of individual metabolic networks in bacterial infection^{90,367}. MulteeSum was developed to visualize three-dimensional gene expression data, and has been used to gain insight into Drosophila development^{55,228}.

Here we describe WholeCellViz’s implementation, features, and visualizations. We also provide two examples of how WholeCellViz can be used to analyze whole-cell model predictions.

5.2 Implementation

5.2.1 Software overview

WholeCellViz is composed of a web-based front-end application and a back-end web server. The front-end displays visualizations to the user. The back-end server stores over 2 TB of simulation data using a combination of a MySQL relational database and JSON (JavaScript Object Notation), and sends this data to the front-end as requested by the user. WholeCellViz was developed as a web application in order to enable platform independence, simple installation, instant developer updates, and data streaming.

5.2.2 Back-end storage server

Our whole-cell model software stores the predicted values of each biological variable at each time point using a set of MATLAB data files. We converted this data into the JSON format using custom Python scripts. We stored the metadata for each simulation, and the label and units for each data point in the database. The WholeCellViz front-end requests metadata and JSON file(s) from the back-end server as needed to display visualizations.

5.2.3 Graphical user interface

The WholeCellViz front-end was implemented in HTML5 and JavaScript using the native canvas to maximize performance. We used JQuery (<http://jquery.com>) to implement event handling, animations, and AJAX calls.

The visualizations were implemented using an extensible framework designed to enable additional visualizations to be easily added to WholeCellViz. Specifically, each visualization extends a common class by defining methods for requesting and displaying data. The source code contains a template for constructing additional visualizations.

We developed the time series plots using the Flot (<http://www.flotcharts.org>) plotting library. We used the JQuery and JQuery UI (<http://jqueryui.com>) libraries to implement WholeCellViz's grid layout and animation controls.

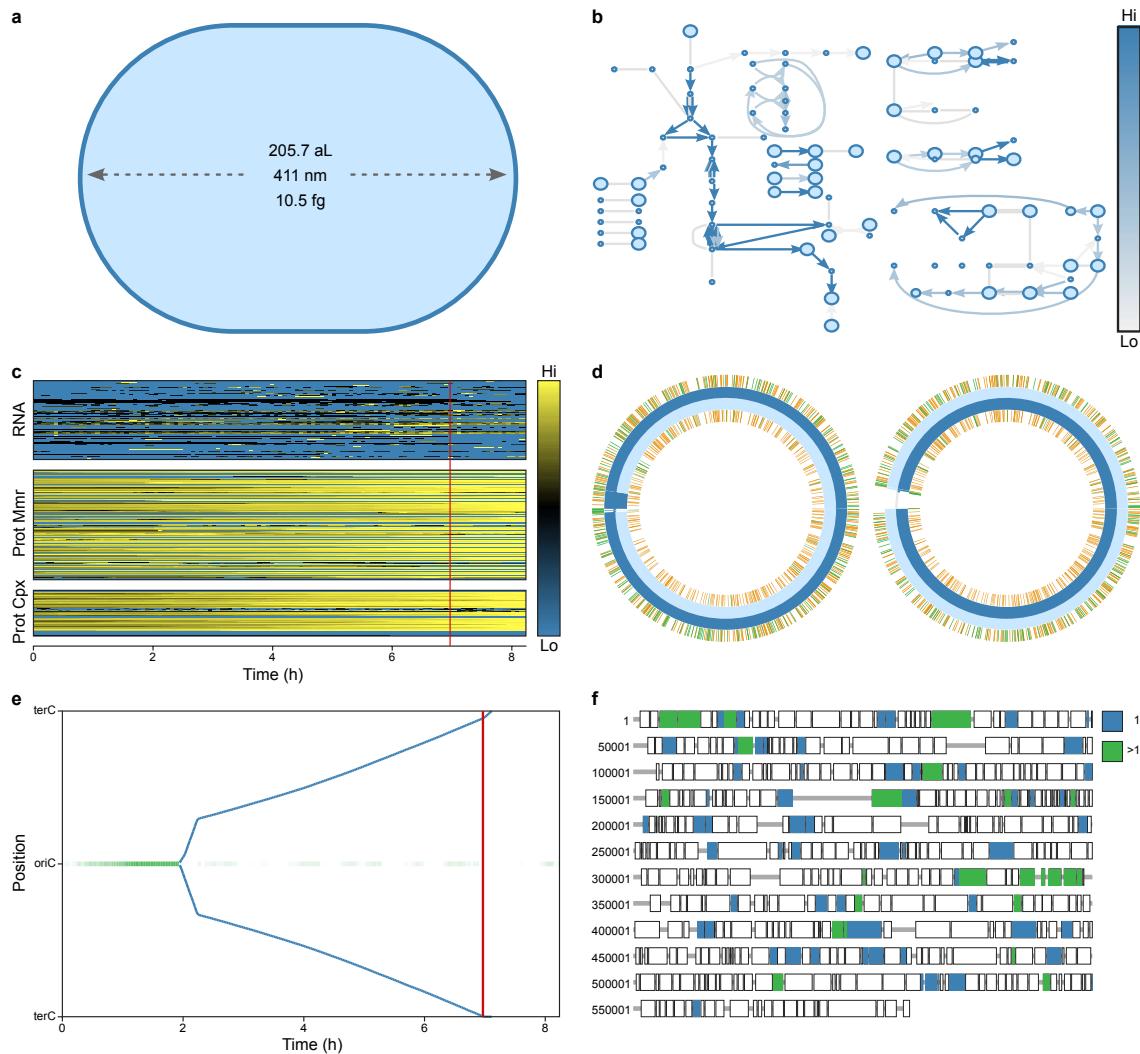


Figure 5.1. Cell cycle dynamics view of one wild-type in silico cell at 7 h post-cell cycle initiation. This view includes six animations which highlight the dynamics of the predicted metabolic fluxes and RNA and protein expression over the cell cycle. In particular, the view shows the onset of DNA replication, and the subsequent bidirectional movement of DNA polymerase on the chromosome. The view also highlights the onset of cytokinesis following the completion of DNA replication. **(a)** Instantaneous shape of *M. genitalium* as it initially elongates and later pinches at the septum, forming two daughter cells. **(b)** Metabolic map illustrating metabolite concentrations and reaction fluxes. Each metabolite is normalized to its mean concentration, and each reaction is normalized to its mean flux. Dark blue arrows indicate high reaction flux; light blue arrows indicate low reaction flux. Large circles indicate high metabolite concentrations; small circles indicate low metabolite concentrations. **(c)** Heatmap of the copy number of each RNA, protein monomer, and protein complex species. Each gene product is normalized to its mean copy number. Yellow indicates high expression; blue indicates low expression. **(d)** Instantaneous polymerization (blue), methylation (orange), strand break (red), and protein-binding status of the *M. genitalium* chromosomes. **(e)** Space-time plot illustrating the instantaneous chromosomal locations of the replication initiator DnaA and DNA polymerase. **(f)** Map of the protein-coding genes indicating protein synthesis. Each gene is colored according to the length of its longest nascent polypeptide. Green represents genes with one active ribosome and blue represents genes with multiple active ribosomes. An interactive version is available at <http://wholecellviz.stanford.edu/cellCycle>.

5.3 Results and discussion

We developed WholeCellViz to accelerate data-driven discovery by visualizing whole-cell model simulation data. WholeCellViz uses simulation data to render 14 visualizations that display model predictions in their biological context. Time series plots supplement the visualizations by showing the detailed dynamics of one or multiple biological variables over time. WholeCellViz lays out these visualizations in an easily configurable grid. The animation timeline controls the simultaneous playback of all displayed animations in the grid. Hence, WholeCellViz is able to simultaneously visualize and animate multiple model predictions.

5.3.1 Features

Figure 5.1 is a sample screenshot of WholeCellViz. We use this figure to describe the features of WholeCellViz.

Visualizations

WholeCellViz contains 14 visualizations that animate specific model predictions within their biological context. These visualizations are listed in Table 5.1 and illustrated in Figures 5.1 and 5.2. Together, these 14 visualizations are capable of displaying 88% of the model’s predictions. These visualizations are also interactive. For example, hovering over the metabolism (Figure 5.1b) visualization reveals tooltips which display metabolite names, compartments, and concentrations. The gene expression panel’s tooltips display gene names, descriptions, and instantaneous copy numbers (Figure 5.1c). Clicking on a gene in the translation panel (Figure 5.1f) opens a new tab which displays the gene’s entry in the WholeCellKB model organism database¹⁷⁴.

Time series plots

WholeCellViz can also display line plots showing the values of one or multiple biological variables over time. For example, the middle-left panel of Figure 5.3 illustrates the temporal dynamics of the intracellular ATP copy number. Time series plots can also display the dynamics of biological variables across simulations, facilitating comparisons across simulations.

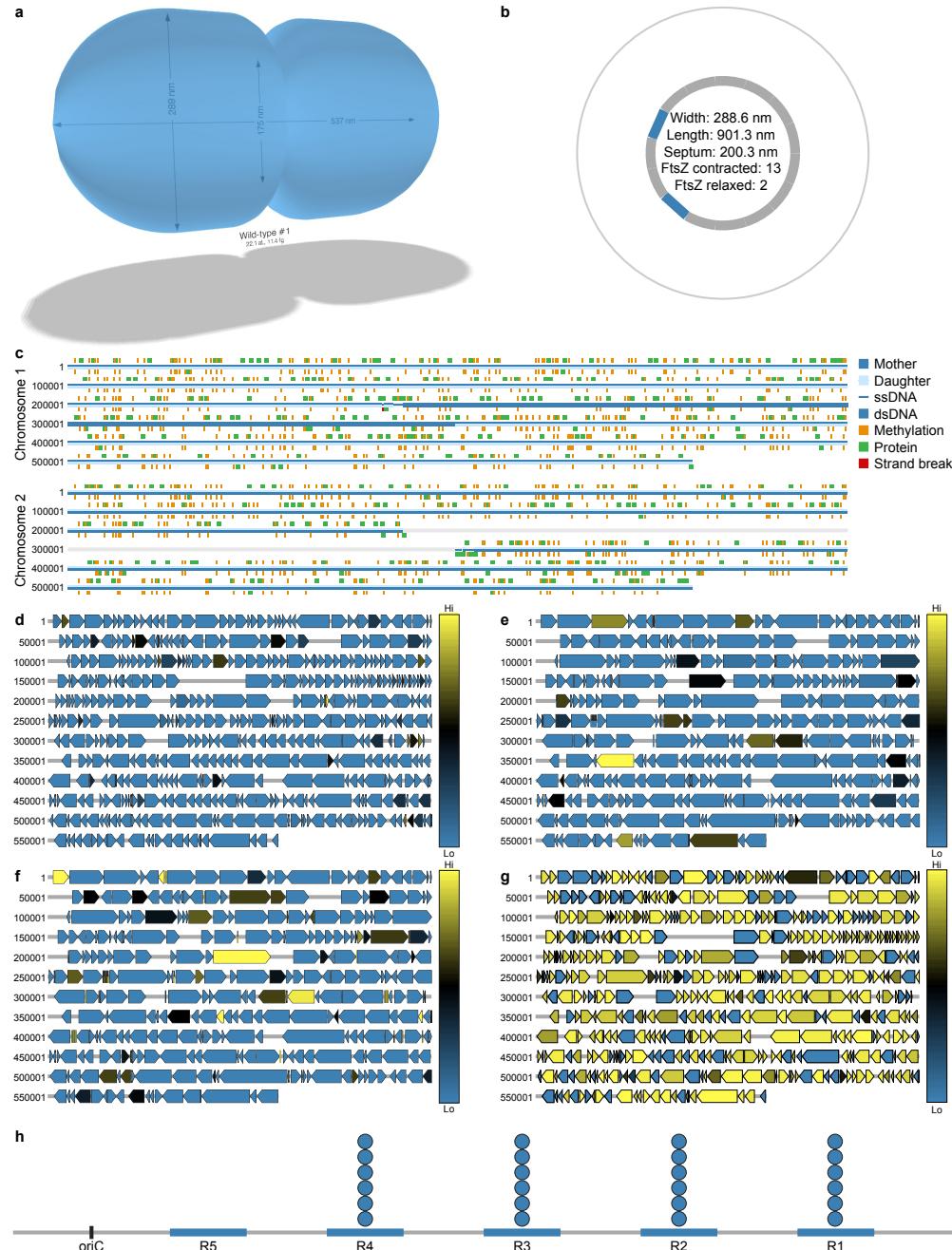


Figure 5.2. Additional WholeCellViz visualizations. Visualizations highlight one wild-type *M. genitalium* cell at various time points. (a) Instantaneous shape of *M. genitalium* as it initially elongates and later pinches at the septum, forming two daughter cells. (b) Instantaneous FtsZ contractile ring size. FtsZ rings iteratively contract at the cell septum to pinch the cell membrane during cytokinesis. (c) Instantaneous polymerization (blue), methylation (orange), strand break (red), and protein-binding status of the *M. genitalium* chromosomes. (d-g) Heatmaps of the copy number dynamics of immature proteins (d), immature RNA (e), mature proteins (f), and mature RNA (g). Each gene product is normalized to its maximal expression. Yellow indicates high expression; blue indicates low expression. (h) Occupancy of the oriC functional DNA boxes which recruit DNA polymerase to the oriC to initiate replication.

Animation timeline

The animation timeline at the bottom of the screen controls the simultaneous playback of all displayed visualizations. It provides play/pause, seek, speed, and repeat controls.

Layout editor

The layout editor is accessed by clicking the gear icon in the top-right corner of the visualization panels. The layout editor enables users to configure the grid dimensions and select the visualization or time series plot displayed in each panel.

Data import

Users can visualize data from any server running the server-side WholeCellViz software. By default, users can visualize simulations we've previously run and stored on our server. Users can install the whole-cell model and WholeCellViz server software on their own machines, or use the whole-cell Linux virtual machine to execute and visualize new simulations. See below for more information about availability.

Graphical and data export

WholeCellViz exports the plotted data in JSON format and exports graphics in SVG format.

Table 5.1. WholeCellViz visualizations.

Visualization	Figure	URL
Cell shape	5.1a	http://wholecellviz.stanford.edu/CellShape
Cell shape (3D)	5.2a	http://wholecellviz.stanford.edu/CellShape3D
Chromosome (linear)	5.2c	http://wholecellviz.stanford.edu/Chromosome1
Chromosome (circular)	5.1d	http://wholecellviz.stanford.edu/Chromosome2
Chromosome (space-time)	5.1e	http://wholecellviz.stanford.edu/ChrSpaceTime
Cytokinesis	5.2b	http://wholecellviz.stanford.edu/Cytokinesis
Gene expression	5.1c	http://wholecellviz.stanford.edu/GeneExp
Immature protein expression	5.2d	http://wholecellviz.stanford.edu/NascentProtExp
Immature RNA expression	5.2e	http://wholecellviz.stanford.edu/NascentRnaExp
Metabolism	5.1b	http://wholecellviz.stanford.edu/Metabolism
Mature protein expression	5.2f	http://wholecellviz.stanford.edu/MatureProtExp
Mature RNA expression	5.2g	http://wholecellviz.stanford.edu/MatureRnaExp
Replication initiation	5.2h	http://wholecellviz.stanford.edu/RepInit
Translation	5.1f	http://wholecellviz.stanford.edu/Translation

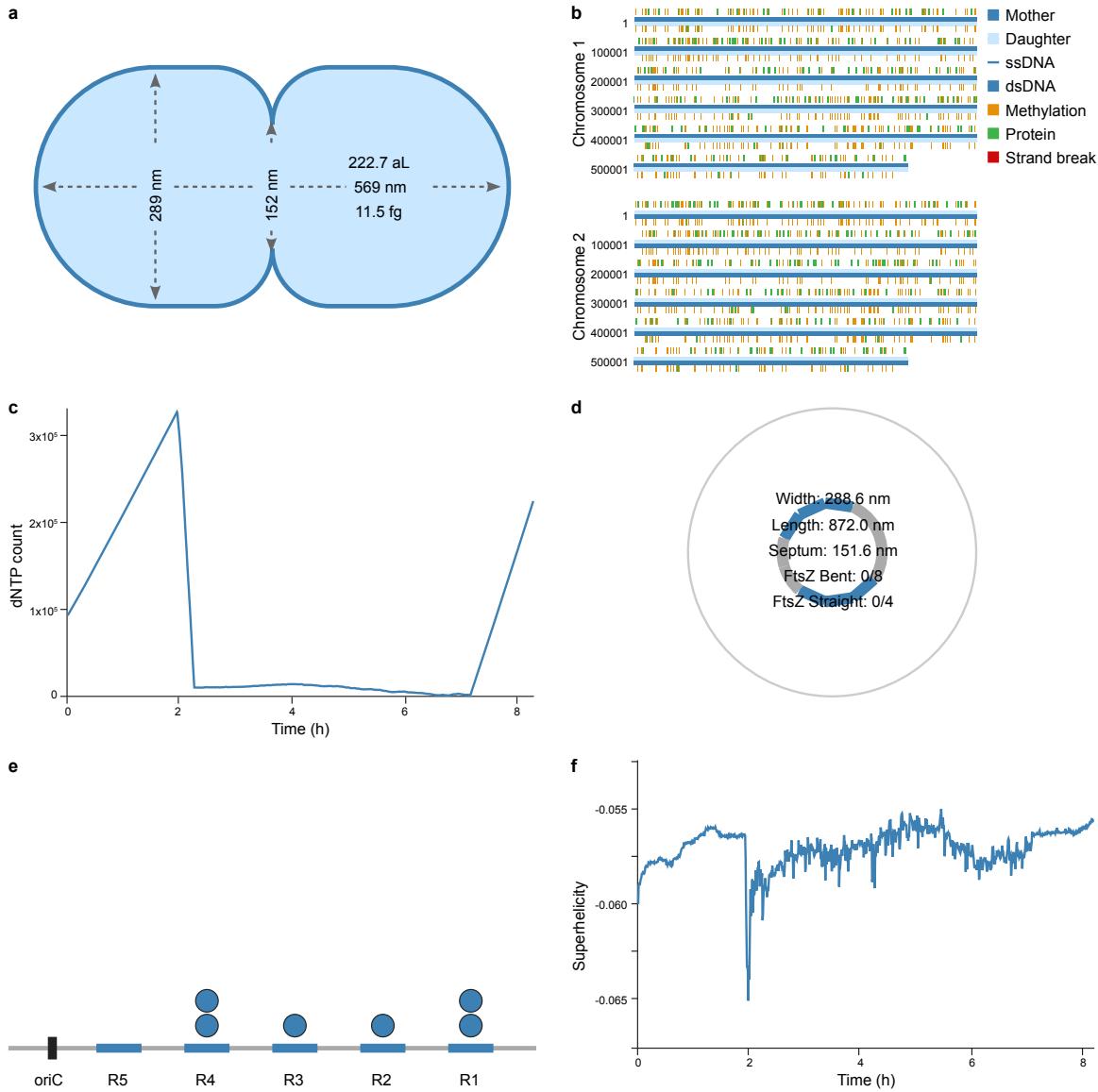


Figure 5.3. Replication dynamics view of one wild-type in silico cell at 7.5 h post-cell cycle initiation. (a) Instantaneous cell shape. (b) Instantaneous polymerization (blue), methylation (orange), strand break (red), and protein-binding status of the *M. genitalium* chromosomes. (c) Intracellular dNTP copy number dynamics. (d) Instantaneous FtsZ and cell septum sizes. (e) Instantaneous oriC DnaA box occupancy. (f) Superhelicity dynamics. An interactive version is available at <http://wholecellviz.stanford.edu/replication>.

5.3.2 Data exploration using WholeCellViz

WholeCellViz can display multiple visualization panels to facilitate comparative and simultaneous analysis of multiple aspects of simulated cell physiology. In particular, WholeCellViz provides six preconfigured views to help users quickly get started. Each of the six views is a grid of visualizations selected to represent a particular aspect of cellular or population dynamics. These views enable users to explore hypotheses about the data. Here we discuss two case studies to illustrate the power of WholeCellViz to facilitate data exploration.

Replication dynamics

Figure 5.3 shows a screen shot of the replication dynamics view. This view displays several perspectives on DNA replication and cytokinesis: cell shape, chromosome dynamics, cytokinesis, replication initiation, and dNTP copy number. First, the view shows that before replication initiates the cell contains a single chromosome and steadily accumulates an increasingly large pool of dNTPs. Second, the view shows that once a sufficiently large oriC DNA complex forms, replication begins accompanied by a sharp drop in the dNTP level. Third, the view shows that replication then proceeds quickly until the dNTP supply is depleted, at which point the rate of replication slows. Finally, the view shows that the FtsZ ring contracts immediately following replication completion.

Population variance

Figure 5.4 shows a screen shot of the population variance view. This view presents summary statistics - growth rate, ATP copy number, dNTP copy number, DNA mass, RNA mass, and protein mass - for eight wild-type *in silico* cells. The view shows that the growth rate, ATP copy number, RNA mass, and protein mass have relatively little variance at the population level. The dNTP copy number and DNA mass have substantially more variance. In three simulations, the dNTP copy number is depleted more than two hours earlier than in the other simulations, and the DNA mass increases earlier in these simulations. This suggests that the timing of DNA replication initiation does not impact the cellular growth rate, ATP copy number, RNA mass, protein mass, or cell cycle length. Rather, the view suggests that metabolism is the primary factor controlling and coordinating the cell's growth, chemical content, and division.

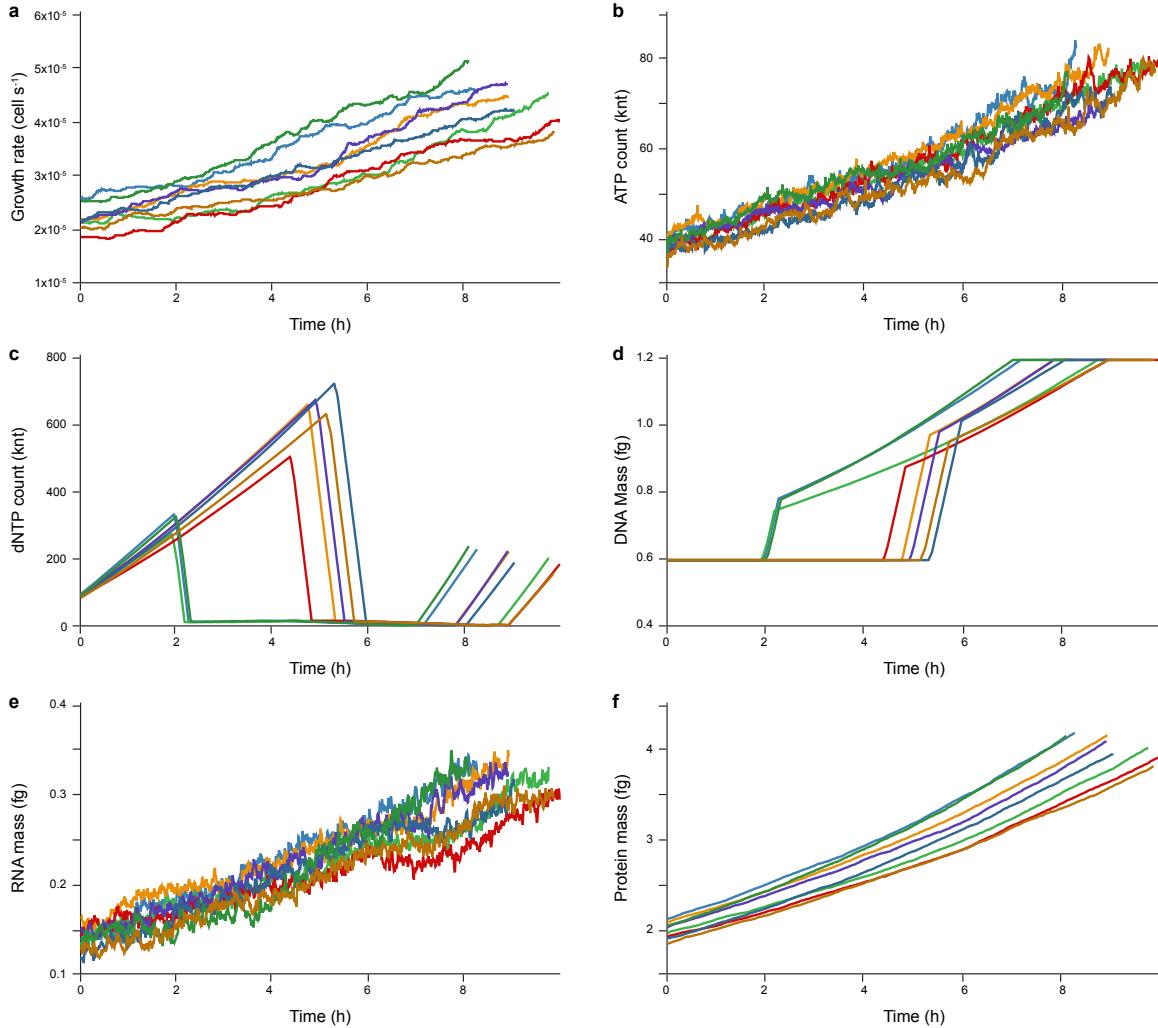


Figure 5.4. Population variance view of eight wild-type in silico cells at 6 h post-cell cycle initiation. View illustrates the temporal dynamics of the cellular growth rate (a), ATP and total dNTP copy numbers (b, c), and DNA, RNA, and protein masses (d–f). Colors indicate the eight in silico cells. An interactive version is available at <http://wholecellviz.stanford.edu/population>.

5.4 Conclusions

WholeCellViz is a web-based program designed to facilitate exploration, and analysis of in silico biological experiments of whole-cell models. The software enables users to fully explore whole-cell model simulations, and displays whole-cell model predictions in their biological context using visualizations and time series plots. Furthermore, WholeCellViz’s grid layout feature enables users to display multiple visualizations and plots, enabling comparative analysis both within and across in silico cells.

Going forward, we plan to improve WholeCellViz as a tool for novel model analysis. We plan to develop new visualizations to communicate additional model predictions including DNA supercoiling and RNA and protein maturation. We also plan to develop enhanced plotting tools for detecting complex relationships among model predictions and analyzing stochastic variation. For example, scatter plots could be used to drill-down to specific time points and examine correlations among multiple variable in a single simulation, or among one variable across multiple simulations. Box plots could be used to compare the variance of variables across simulations.

Currently, only one whole-cell model has been developed, the one whole-cell model is computationally expensive to simulate, and there are no broadly accepted standards for representing whole-cell model predictions. Consequently, we chose to focus WholeCellViz on simulations that we have previously run using our *M. genitalium* model and stored on our server. Going forward, we plan to integrate WholeCellViz with other whole-cell models and simulation data servers as they become publicly available. Currently users can visualize alternative whole-cell model simulations by (1) running their own simulations using either our *M. genitalium* model or a similarly detailed model, (2) storing their simulations on their own server using the hybrid MySQL/JSON format described here, and (3) editing the back-end server URL configuration option from the WholeCellViz front-end. Researchers can achieve this either by installing the whole-cell model and WholeCellViz software on their own machine or by using our Linux virtual machine which contains both the whole-cell model and WholeCellViz software (see below for more information about availability). In the future, we also plan to enable researchers to configure and run whole-cell simulations through a simple graphical interface within WholeCellViz. However, this will require the development of more computationally efficient whole-cell model simulations.

Overall, whole-cell modeling is an emerging field that has the potential to accelerate the pace of biological discovery and enable rational bioengineering and personalized medicine. Data visualization software such as WholeCellViz is critically needed to help researchers access, explore, and analyze complex, high-dimensional whole-cell model simulations, as well as to accelerate model-driven biological discovery. With the current influx of big data in research and industry, WholeCellViz also serves as an example of how to use animation for scientific communication. We anticipate that WholeCellViz will play a critical role in realizing the full potential of whole-cell models.

5.5 Availability and requirements

Project name: WholeCellViz

Project home page: <http://wholecellviz.stanford.edu>

Operating system(s): Platform independent

Programming language: HTML, JavaScript, PHP

Other requirements: Web browser

License: MIT license

Any restrictions to use by non-academics: None

WholeCellViz is available under the MIT license at <http://wholecellviz.stanford.edu>. The hosted version visualizes simulations we've previously run and stored on our server, and is also capable of visualizing simulations stored on other servers running the WholeCellViz server-side software. Researchers can install the whole-cell model and WholeCellViz software locally to execute and visualize new simulations. All source code is available open-source at SimTK: <http://simtk.org/home/wholecell>. A Linux virtual machine containing the whole-cell model and WholeCellViz server and client software is also available at SimTK.

Chapter 6

Toward more accurate whole-cell models: new data and tools

Whole-cell models promise to enable rational bioengineering and personalized medicine by predicting how complex cellular behaviors arise from the molecular level. Recently we demonstrated a novel integrative modeling approach that enabled the first whole-cell model of a single living organism (Chapter 4). The model accounts for every annotated gene function, and predicts the dynamics of every molecular species over the entire cell cycle. We trained the model using over 1,900 data points curated from more than 900 publications. We showed that the model predicts gene essentiality with almost 80% accuracy.

Going forward, more accurate whole-cell models are needed to truly enable biological design and personalized medicine. This chapter briefly describes three ongoing projects to improve the accuracy and capabilities of whole-cell models. Section 6.1 introduces an ongoing collaboration with Veronica Llorens, Maria Lluch-Senar, Luis Serrano, and Markus Covert to improve the accuracy of whole-cell models by training models using more consistent, more systematic, and broader experimental data. Section 6.2 describes a crowdsourced effort led by Pablo Meyer, Gustavo Stolovitzky, and I to develop improved algorithms for estimating whole-cell model parameters. Lastly, Section 6.3 outlines WholeCellDB, a new database for model predictions which enables researchers to easily retrieve model predictions and assess their quality. We are developing WholeCellDB in collaboration with Nolan Phillips and Yingwei Wang.

6.1 A whole-cell model of *Mycoplasma pneumoniae*

Whole-cell models contain thousands of parameters which must be trained using experimental data. We trained the first whole-cell model of *M. genitalium* using a highly heterogeneous and noisy collection of data which we curated from the scientific literature. The poor quality of this data limited the accuracy of the *M. genitalium* model.

Currently we are collaborating with Veronica Llorens, Maria Lluch-Senar, and Luis Serrano to develop a more accurate whole-cell model the Gram-positive bacterium *M. pneumoniae* using their wealth of consistent, systematic experimental data. We plan to use additional training data including the observed DNA fold, antisense RNA expression, and protein copy numbers and half-lives to further increase the accuracy of the *M. pneumoniae* model.

Recently we curated all of the available *M. pneumoniae* training data using our WholeCellKB software. Next, we plan to adapt our *M. genitalium* whole-cell model to *M. pneumoniae*. Finally, we plan to train the *M. pneumoniae* model to the curated data.

We are particularly interested in comparing the *M. pneumoniae* predictions to that of *M. genitalium* to explore why *M. pneumoniae* grows faster than *M. genitalium*. Our current hypothesis is that *M. genitalium* has comparatively low expression levels of key metabolic enzymes needed to drive faster growth. We plan to explore this hypothesis theoretically using the model, bioinformatically using measured expression data, and experimentally by overexpressing suspected control enzymes. Ultimately, we hope to use the model to rationally design a faster growing *M. genitalium* strain.

In addition to comparing *M. genitalium* and *M. pneumoniae*, we plan to use the *M. pneumoniae* model to analyze the systems level effects of transcriptional and translational regulation. In particular, we aim to better understand the contributions of transcriptional regulators, antisense RNA, RNA binding proteins, and RNA and protein stability to RNA and protein expression and the cellular growth rate.

Taken together, we anticipate that the *M. pneumoniae* model will illustrate the possible accuracy of whole-cell models, as well as illuminate why *M. genitalium* grows significantly slower than *M. pneumoniae*.

6.2 Crowdsourced development of improved whole-cell model parameter estimation methods

Whole-cell models contain thousands of parameters, many of which are critical to producing accurate predictions. Traditional methods for estimating these parameters, including gradient descent and scatter search, are poorly suited to whole-cell models because little data is available, the little data which is available is noisy and heterogeneous, and whole-cell models are computationally intensive to simulate. Consequently, training whole-cell model parameters is extremely challenging.

One possible solution to estimating whole-cell model parameters is to instead estimate the parameters of a reduced surrogate model of the same dimensionality. The time-population average is an effective surrogate because most biological experiments do not report temporal dynamics or individual variation. We've shown that this surrogate can be approximately and efficiently calculated analytically (Chapter 4).

Recently Pablo Meyer, Gustavo Stolovitzky, and I initiated a crowdsourced effort to identify additional methods for estimating whole-cell model parameters. In particular, we organized a competition in which participants were challenged to identify 30 modified parameters given the model structure and *in silico* high-throughput “experimental” data. We modified the parameters to increase the models’s predicted growth rate by 50%.

Initially we gave participants eight data sets “measured” with the modified parameters. In addition, we gave participants credits to purchase up to 50 of 480 available *in silico* data sets measured for perturbations to the modified parameters. We motivated researchers to participate by offering several prizes including invitations to present at conferences and publish scientific papers, \$1,000, and scientific computing software licenses. We awarded researchers prizes based on a combination of the accuracies of their predictions and estimated parameter values. The competition is described in detail at <http://www.synapse.org/#!Synapse:syn1876068>.

After the challenge we plan to review the best performing methods and combine them into an even better performing meta method. Overall, we are optimistic that researchers will participate in the competition and develop creative new methods for estimation whole-cell model parameters.

6.3 WholeCellDB: database for whole-cell model predictions

Whole-cell model simulations produce rich data containing valuable information about the determinants of cellular behavior. However, manually exploring and assessing the quality of the over 50 billion data points produced by a typical whole-cell simulation data set is challenging.

Currently we are developing WholeCellDB, a database which will enable researchers to efficiently store and query whole-cell model predictions. WholeCellDB is hybrid SQL/HDF5 database. WholeCellDB uses SQL to store simulation meta data and HDF5 to store predicted cellular trajectories. This design captures the advantages of both SQL and HDF5, enabling researchers to both efficiently search meta data, as well as efficiently store model predictions.

WholeCellDB provides both command line and web-based interfaces. The command line interface enables users to programmatically store and retrieve model predictions. The web-based interface enables users to easily browse and query model predictions with minimal knowledge of whole-cell modeling. The web-based interface includes references to the WholeCellKB family of model organism databases, as well as to the whole-cell model source code.

WholeCellDB is implemented in Python and MySQL using the Django framework and the PyTables HDF5 library. WholeCellDB will be available open-source under the MIT license at <http://github.com>.

Chapter 7

Conclusion

This thesis describes the development and application of a new methodology for constructing comprehensive computational models of biological systems from the molecular level. The new methodology described here is based on an integrative approach in which cellular processes are each modeled independently on short time scales and integrated together at longer time scales. A key feature of this approach is that it enables each process to be represented using the most appropriate mathematics and data, taking full advantage of all of the heterogeneous data and mathematics available in the scientific literature. For example, it allows metabolism to be modeled with flux-balance analysis and trained using cell composition data at the same time that transcription is modelled as a Markov process and trained using microarray data. All of the software described in this thesis is available open-source under the MIT license at <http://simtk.org/home/iFBA> and <http://simtk.org/home/wholecell>.

Chapters 2-4 report this integrative approach and its implementation in a comprehensive model of the Gram-positive bacterium *M. genitalium*. Chapter 2 describes the data used to train the *M. genitalium* model and the software we developed to curate the data. Chapter 3 outlines our early efforts to develop comprehensive models using this integrative approach. This chapter describes a dynamical model of *Escherichia coli* composed of three mathematically distinct sub-models of signaling, metabolism, and transcriptional regulation which are directly coupled to each other. Chapter 4 generalizes the integrative approach pioneered in Chapter 3 by using cell state variables which represent the instantaneous cellular configuration to indirectly couple sub-models. This chapter describes a dynamical model of *M. genitalium* which is composed of twenty-eight mathematically and biologically distinct sub-models.

Chapter 4 describes the application of the *M. genitalium* whole-cell model to biological discovery and bioengineering. In this chapter we use the *M. genitalium* model to explore previously unobserved cellular behaviors, including in vivo rates of protein-DNA association, an inverse relationship between the durations of DNA replication initiation and replication, and previously undetected kinetic parameters and biological functions.

Chapter 5 details a web-based software application for visually analyzing whole-cell model predictions and other high-dimensional biological data. We developed this software application to provide ourselves and other researchers tools for quickly exploring very large scale data with minimal knowledge of computational modeling, software engineering, and the particular data structures used to efficiently represent high-dimensional data.

Lastly, Chapter 6 outlines three ongoing projects to improve the scope and accuracy of whole-cell models. First, the chapter describes our efforts to develop a more comprehensive and accurate model of *M. pneumoniae* by using more consistent and more systematic data than we used to train our first whole-cell model of *M. genitalium*. Going forward, we plan to use this model in conjunction with our *M. genitalium* model to study why *M. pneumoniae* grows significantly faster than *M. genitalium*. Our ultimate goal is to engineer a faster growing strain of *M. genitalium*. Second, this chapter describes a community competition to develop better tools for accurately estimating whole-cell model parameters. After the competition we plan to develop a meta method which combines aspects of all of the best performing solutions. Third, this chapter describes the development of WholeCellDB, a database for storing and quickly retrieving whole-cell model predictions. We anticipate that WholeCellDB will help researchers gain more insight from whole-cell model simulations.

Whole-cell modeling is a new and exciting field with enormous potential to revolutionize basic biological science, bioengineering, and medicine. Despite the progress reported here, significant work remains to achieve truly accurate and reliable whole-cell models. Most importantly, researchers must develop more comprehensive and larger-scale models, starting with well studied bacteria such as *E. coli* and *Bacillus subtilis*. To realize the medical potential of whole-cell modeling, researchers ultimately must create multiscale models of eukaryotes, including model multicellular species such as worms and flies. This could be achieved by combining the integrative approach described here with agent-based methods to create multicellular models composed of agents which simulate individual cells, which in turn are modeled using integrative approaches. In addition, researchers must explore new whole-cell modeling applications such as drug screening and optimization.

In order to build and apply increasingly sophisticated whole-cell models, researchers must also continue to develop increasingly powerful tools for assembling and simulating whole-cell models. In particular, researchers must develop tools which enable large-scale collaborative development of whole-cell models. In turn, this will require stricter standards for sub-model development and testing. Improved tools for quickly simulating whole-cell models are also needed to enable researchers

to more quickly test whole-cell models during their development, as well as more quickly use whole-cell models to explore hypotheses.

In conclusion, we have pioneered a new modeling methodology which has enabled us to build the first comprehensive computational model of a single living organism. We have used our model to explore new areas of biology, discover new kinetic parameters, and identify previously unknown regulatory mechanisms. Most importantly, our methodology provides a stepping stone toward model-driven rational biological design and personalized medicine.

Appendix A

NetworkAnalyzer: intracellular pathway animation

Abstract

Summary: Flow and mass cytometry provide high-dimensional measurements of single cells. However, exploring and analyzing cytometry data is tedious and challenging. We developed a web-based software program, NetworkAnalyzer, to enable researchers to analyze cytometry data in the context of biological pathway diagrams. NetworkAnalyzer provides researchers a graphical interface to draw and paint pathway diagrams with experimental data, producing animated diagrams. NetworkAnalyzer also displays heatmaps to compare observations across conditions, cell types, and individuals.

Availability: NetworkAnalyzer is freely available at <http://covertlab.stanford.edu/projects/NetworkAnalyzer>. NetworkAnalyzer is also integrated into the Cytobank flow cytometry repository at <http://www.cytobank.org>. Source code is freely available at <http://simtk.org/home/networkanalyzer>.

A.1 Introduction

Cellular signaling is enormously complex, arising from interactions among thousands of molecules. Understanding signaling at the molecular level therefore requires high-dimensional measurements of individual cells. Fluorescence-based flow cytometry enables up to 17 simultaneous measurements³⁷⁸ and mass cytometry promises as many as 100 simultaneous measurements¹⁰⁰. However, exploring and analyzing cytometry data is challenging and tedious.

Visualization software is an effective technique for investigating complex data³⁶⁰. Graphical software including Cytobank (<http://www.cytobank.org>²⁷⁰, FlowJo (<http://www.flowjo.com>) and flowViz⁸³ is commonly used to inspect and gate cytometry data. Recently, Plethritis and colleagues developed

SPADE to cluster and visualize cytometry measurements across multiple cell types²⁸⁹. Furthermore, Cytoscape is frequently used to visualize complex biological networks²⁴⁶. Visualization software is becoming increasingly important as biological data continues to grow in complexity and volume.

NetworkAnalyzer is a web-based program for drawing and painting signaling network diagrams with high-dimensional cytometry data. Two versions of NetworkAnalyzer are available. The standalone version at <http://covertlab.stanford.edu/projects/NetworkAnalyzer> visualizes uploaded cytometry data. Additionally, we integrated NetworkAnalyzer into the Cytobank flow cytometry repository (<http://www.cytobank.org>) to facilitate analysis of flow and mass cytometry data. We believe that NetworkAnalyzer will help scientists interpret increasingly complex biological data.

Here we describe the features and implementation of NetworkAnalyzer. We present a mass-cytometry time-course of the human peripheral blood mononuclear cell (PBMC) immune signaling network¹² as an example use case. We conclude by discussing our future plans for NetworkAnalyzer.

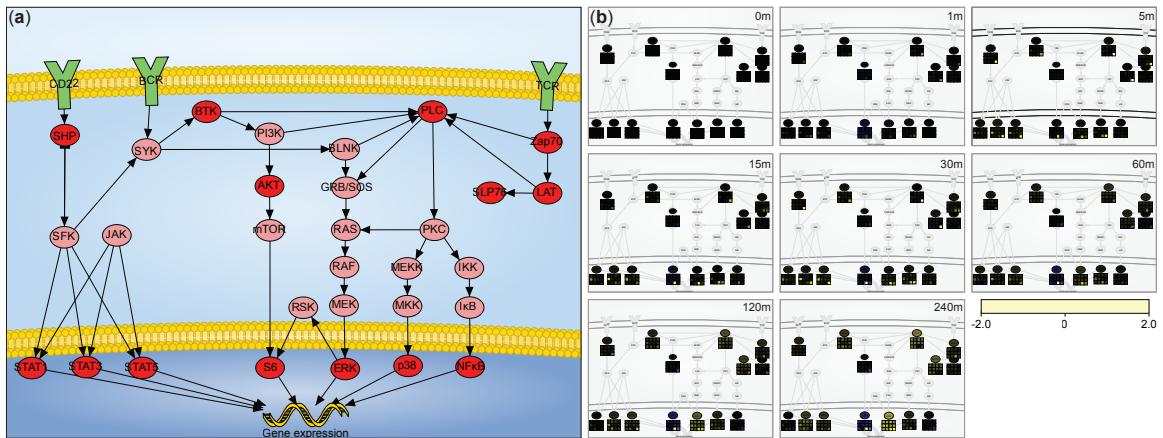


Figure A.1. NetworkAnalyzer visualizes high-dimensional cytometry data in the context of biological pathways. (a) PBMC immune signaling diagram created using NetworkAnalyzer and adapted from 12. Figure S6¹². (b) PBMC pathway painted with a mass cytometry time-course¹². An interactive, animated version of Figure A.1 is available at <http://covertlab.stanford.edu/projects/NetworkAnalyzer/KarrEtAl2013Fig1>.

A.2 Features

NetworkAnalyzer provides researchers a simple interface to visually analyze high-dimensional cytometry data. First, users draw pathway diagrams, formalizing their prior biological knowledge. Alternatively, users can import pathways from the KEGG PATHWAY database²²⁶. Second, users

upload experimental time-courses to the standalone version, or access experiments stored in Cyto-bank. Third, users link diagram nodes to observed channels. Next, NetworkAnalyzer paints nodes with experimental data, creating an animated diagram. Where applicable, NetworkAnalyzer displays small heatmaps below each linked node to illustrate each experimental condition, cell type, and individual.

NetworkAnalyzer saves pathways to a central server for later use. NetworkAnalyzer provides three permissions levels (read, write, administer) to enable researchers to collaboratively edit and analyze pathways. NetworkAnalyzer also provides permalinks to enable researchers to publish pathway diagrams.

NetworkAnalyzer exports diagrams in several graphical formats including GIF, JPG, PDF, PNG, and SVG as well as to the BioPax, CellML, and SBML markup languages and JSON. NetworkAnalyzer exports animations to GIF, PDF, and SWF formats. NetworkAnalyzer can import diagrams in JSON format or from the KEGG PATHWAY database. Additionally, NetworkAnalyzer can generate MATLAB scripts for Boolean pathway simulations.

Figure A.1a depicts a human PBMC immune signaling diagram created using NetworkAnalyzer. Figure A.1b depicts eight static snapshots of the same pathway painted with a mass cytometry time-course of 14 signaling nodes¹². An interactive, animated version of Figure A.1 is available at <http://covertlab.stanford.edu/projects/NetworkAnalyzer/KarrEtAl2013Fig1>.

A.3 Implementation

NetworkAnalyzer is composed of a web-based graphical pathway editor/painter, and a pathway storage server. The user interface was implemented in Adobe Flex (<http://www.adobe.com/products/flex.html>). The server was implemented using PHP (<http://www.php.net>), MySQL (<http://www.mysql.com>) and Apache (<http://www.apache.org>). The user interface and server communicate using AMFPHP (<http://www.silexlabs.org/amfphp/>).

A.4 Discussion

NetworkAnalyzer is a web-based program for visually analyzing dynamic, high-dimensional cytometry data in the context of pathway diagrams. NetworkAnalyzer provides a simple graphical interface

for drawing and linking pathways to either uploaded data or data stored in the Cytobank repository. Going forward, we plan to integrate NetworkAnalyzer with additional pathway and cytometry databases to make it easier for researchers to use these resources.

We believe that NetworkAnalyzer can help researchers collaboratively analyze increasingly complex biological data. Furthermore, we hope that researchers will use NetworkAnalyzer to provide interactive public access to raw experimental data.

Appendix B

Integrating metabolic, regulatory and signaling models: Supplemental information

Table B.1 lists the initial conditions, flux bounds, and additional parameters of the *Escherichia coli* central metabolism iFBA, rFBA, and ODE models. The GLCex, G6Pex, and LCTSex bounds were generally not used in the iFBA simulation. These fluxes were set to that calculated by the ODE model, except where the ODE model was found to be numerically unstable in which case the iFBA modeled defaulted to the rFBA model, and the GLCex, G6Pex, and LCTSex bounds were used. Except where noted in Table B.1, bounds for reversible fluxes were -500.0 and 500.0 and bounds for non-reversible fluxes were 0.0 and 500.0. Bounds listed in Table B.1 were determined experimentally by fitting the iFBA model to experimental data (178.).

The metabolic reactions and boolean regulation of the iFBA model are described in 269. with the addition of two fluxes corresponding to glucose-6-phosphate uptake by UhpT and the removal of the reaction constraints on galEKMPT, lacYZ, and pykF, which are encapsulated by the ODE model. The kinetic model of the iFBA model, and the values of its parameters, are described in 2.

Figures SB.1 and B.2 compare the iFBA, rFBA, and ODE predicted time courses of biomass and several metabolites and proteins for *E. coli* diauxic growth on glucose/glucose-6-phosphate.

Table B.2 lists the kinetic parameter-gene correspondences of the *E. coli* central metabolism iFBA model.

Table SB.3 lists the predicted phenotypes of all single gene perturbations for which all three models qualitatively predicted the same phenotype – healthy or unhealthy. Single gene perturbations with qualitatively different phenotypes predicted by the three models are illustrated in Figure 3.5.

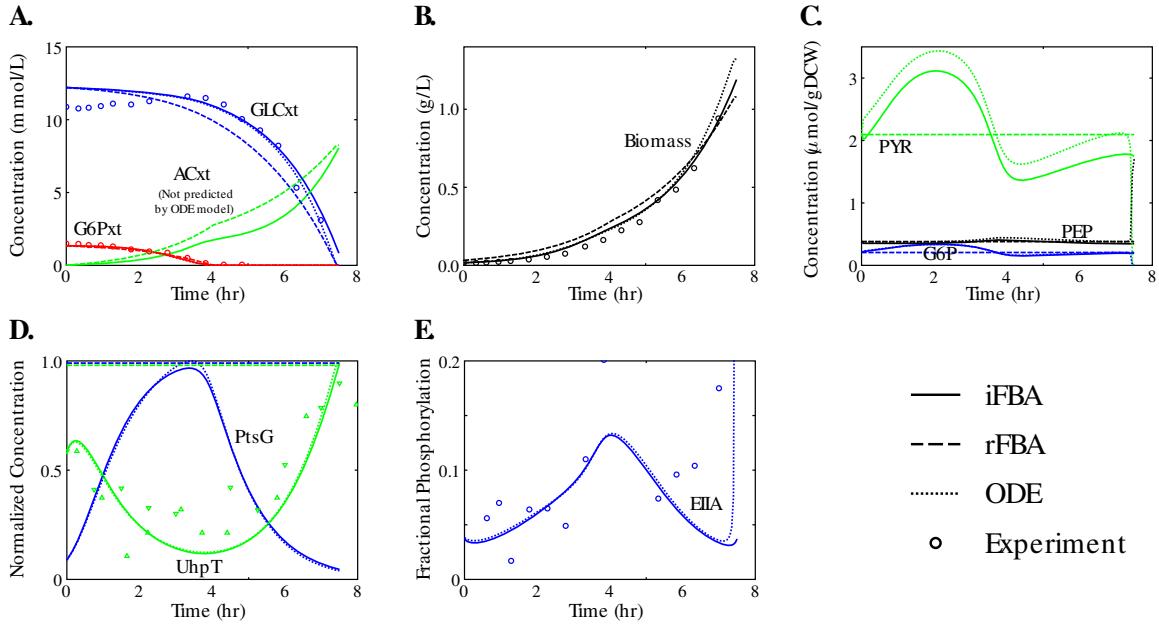


Figure B.1. Growth of the iFBA (solid lines), ODE (dotted), and rFBA (dashed) wild type models in an aerobic environment with glucose and glucose-6-phosphate as carbon sources, together with experimental data (178.) where available (circles). Dynamic time profiles of external (A) acetate, glucose, glucose-6-phosphate and (B) biomass concentrations; (C) internal glucose-6-phosphate, phosphoenolpyruvate, and pyruvate concentration; (D) key protein concentrations; and (E) degree of phosphorylation of regulatory protein EIIA^{Crr}.

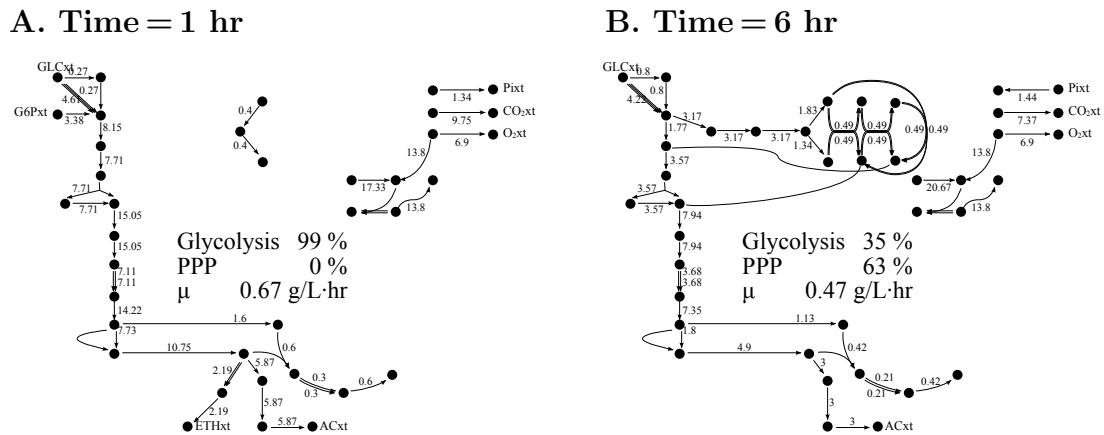


Figure B.2. Flux distributions for iFBA simulation of glucose/glucose-6-phosphate diauxic growth, at (A) one hour and (B) six hours. Detailed labels for the network are shown in Figure 3.1, and all values are in mmol/gDCW/hr.

Table B.1. Initial conditions, flux bounds, and additional parameters for iFBA, rFBA, and ODE simulations.

Parameter	Value	
Initial Condition	GLC/G6P	GLC/LCTS
Biomass (g L ⁻¹)	0.032	0.032
External Metabolites (m mol L ⁻¹)		
acetate	0.0	0.0
carbon dioxide	100.0	100.0
ethanol	0.0	0.0
formate	0.0	0.0
galactose	0.0	0.0
glucose-6-phosphate	1.3	0.0
glucose	12.2	1.2
lactate	0.0	0.0
lactose	0.0	3.4
oxygen	100.0	100.0
phosphate	100.0	100.0
pyruvate	0.0	0.0
ribose	0.0	0.0
succinate	0.0	0.0
Internal Metabolites (μ mol gDCW ⁻¹)		
glucose-6-phosphate	0.206	0.100
phosphoenolpyruvate	0.381	0.050
pyruvate	2.095	0.100
Proteins (μ mol gDCW ⁻¹)		
LacZ	0.0	0.00001
UhpT	0.0003	0.0
PtsG	0.007	0.001
Protein Phosphorylation States (-)		
EIHA ^{Crr}	0.004	0.010
Flux		
	GLC/G6P Bound	GLC/LCTS Bound
	Lower	Upper
ACex	0.0	500.0
ATPM	7.6	7.6
GLCex	-4.9	500.0
G6Pex	-2.9	500.0
LCTSex	0.0	500.0
CO2ex	-500.0	500.0
O2ex	-6.9	500.0
PIex	-10.0	500.0
Growth rate scale, β	1.2	
Protein Synthesis-degradation delay	24 min	
Time step	3 min	

Table B.2. Genes required for full activity of each enzyme in the ODE model.

Enzyme/Parameter	Gene Products
$dEIIA/dt$	crp
k_1	uhpT
k_2	ptsG crr (galP & glk)
k_3	lacY & lacZ & (glk & (gale & galK & galM & galT & galU & pgm))
k_{gly}	pgi & (pfkA pfkB) & fbaA & gapA & pgk & (gpmA gpmB) & eno
k_{pdh}	(aceE & aceF & lpd) (pflA & pflB) (pflC & pflD)
k_{pts}	ptsG crr (galP & glk)
k_{pyk}	pykA pykF
x_o	crp

Table B.3. Single gene perturbations with the same predicted phenotype in each of the iFBA, rFBA, and ODE models. All enzymatic perturbations are knockdowns.

	Healthy (> 80% wild type growth)				Unhealthy (< 80% wild type growth)			
	Enzymes		Transcription Factors		Enzymes		Transcription Factors	
GLC/ G6P	GLC/ LCTS	GLC/ G6P	GLC/ LCTS	GLC/ G6P	GLC/ LCTS	GLC/ G6P	GLC/ LCTS	
aceAB	aceAB	arcA=0,1	arcA=0,1	eno		crp=0	crp=0	
ackA	ackA	cra=0,1	cra=0,1	fbaA		galEKMUTU		
acnAB	acnAB	crp=1	crp=1			glk		
acs	acs	dcuRS=0,1	dcuRS=0,1	gapA		lacYZ		
actP	actP	fadR=0,1	fadR=0,1			pgm		
adhE	adhE	galRS=0,1	galRS=0					
adk	adk	glpR=0,1	glpR=0,1	pgk				
crr	crr	iclR=0,1	iclR=0,1					
cydAB	cydAB	lacI=0,1	lacI=0					
cyoABCD	cyoABCD	mlc=0,1	mlc=0,1					
dctA	dctA	rbsR=0,1	rbsR=0,1					
dcuABC	dcuABC	rpiR=0,1	rpiR=0,1					
ddl	ddl							
fdnGHI	fdnGHI							
fdoGHI	fdoGHI							
fdp	fdp							
fnr	fnr							
focA	focA							
frdABCD	frdABCD							
fumABC	fumABC							
galeKMTU	galeKMTU	galP						
glpABCDFK	glpABCDFK							
gnd	gnd							
gpmAB	gpmAB							
gpsA	gpsA							
lacYZ								
ldhA	ldhA							
maeAB	maeAB							
mdh	mdh							
ndh	ndh							
pck	pck							
pdhR	pdhR							
pfkAB	pfkAB							
pflABCD	pflABCD							
pgl	pgl							
pgm								
pntAB	pntAB							
ppa	ppa							
ppsA	ppsA							
pta	pta							
ptsGHI	ptsGHI							
pykAF	pykF							
rbsABCK	rbsABCK							
rpe	rpe							
rpiB	rpiB							
sdhABCD	sdhABCD							
sucABCD	sucABCD							
talAB	talAB							
tktAB	tktAB							
	uhpT							
zwf	zwf							

Appendix C

A whole-cell computational model predicts phenotype from genotype: Supplemental information

C.1 Computational Methods

The goal of this study was to predict the complex phenotypes of individual cells in terms of individual molecules and their interactions. The primary challenges to building a unified whole-cell model of a single cell are three-fold: (1) *complexity*, processes relevant to cellular behavior are diverse and span a wide range of length and time scales, (2) *heterogeneity*, cellular networks have heterogeneous mathematical structures and are typically investigated using heterogeneous experimental methods, and (3) *sparsity*, little quantitative data rigorously describing single cell physiology is available. We felt that the flexibility of a hybrid model allowed us to best meet our goal while navigating these challenges. This chapter discusses the mathematical foundation and construction of the *M. genitalium* whole-cell model, as well as model fitting and validation. Appendix C.5 provides further discussion of the implementation, execution, and testing of the whole-cell model.

C.1.1 Whole-Cell Simulation Algorithm

Overall, the whole-cell model is similar to a system of ordinary differential equations (ODEs) where the 16 cellular states are analogous to the state variables and the 28 cellular processes are analogous to the differential equations. Therefore, the whole-cell model is simulated using an algorithm comparable to those used to numerically integrate ODEs, such as the Runge-Kutta 4th order method³¹⁶. Algorithm C.1 summarizes the whole-cell simulation algorithm. First, the cell state variables are initialized. Second, the temporal evolution of the cell state is calculated on a 1 s time scale by repeatedly allocating the cell state variables among the processes, executing each of the cellular process sub-models, and updating the values of the cell states. Finally, the simulation terminates

when either the cell divides, or the time reaches a predefined maximum value. The principal differences between the whole-cell model algorithm and numerical ODE integration methods are (1) the whole-cell model “equations” are grouped into 28 processes, (2) the whole-cell model “variables” are grouped into 16 cellular states, and (3) the state variables must be allocated among the processes at each time step to satisfy the sub-model independence assumption.

Algorithm C.1. Whole-cell dynamic simulation algorithm.

Construct whole-cell simulation objects using the KnowledgeBase classes
 Computationally align processes and fit parameters
 Identify initial conditions variance control parameters using Algorithm C.2
 Initialize cell state using Algorithm C.3 and the fit values of the cell state variance control parameters
repeat
 Increment the time by 1s
 Set the external conditions based on Table S3F and Table S3H
 Allocate shared resources:
 foreach metabolite i in compartment j **do**
 foreach process k **do** Calculate the demand, d_{ijk} , of process k for metabolite i in compartment j
 Divide the total count, m_{ij} , of metabolite i in compartment j into temporary dedicated pools, m_{ijk} , for
 each process proportional to demand, $m_{ijk} \leftarrow m_{ij} \frac{d_{ijk}}{\sum_k d_{ijk}}$
 Compute temporal evolution:
 foreach process i **do**
 Retrieve the current values of cell state variables and the counts of metabolites allocated to process i
 Compute the contribution of process i to the temporal evolution of the cell state
 Update the values of the cell state variables
until cell divided or time $> 1.5 \times$ average mass doubling time

C.1.2 Cellular Processes

Because biological systems are modular, cells can be modeled by (1) dividing cells into functional processes, (2) independently modeling each process on a short time scale, and (3) integrating process sub-models at longer time scales. We divided *M. genitalium* into the 28 functional processes illustrated in Figure 1 of the accompanying manuscript, and modeled each process independently on a 1s time scale using different mathematics and different experimental data. The sub-models spanned six areas of cell biology: (1) transport and metabolism, (2) DNA replication and maintenance, (3) RNA synthesis and maturation, (4) protein synthesis and maturation, (5) cytokinesis, and (6) host interaction. Sub-models were implemented as separate classes. See Section C.3 for further discussion of each sub-model.

C.1.3 Cellular Process Integration

Cell States

We integrated the sub-models in three steps. First, we structurally integrated the process sub-models by linking their common inputs and outputs. However, rather than directly linking these inputs and outputs, we mapped the inputs and outputs of each sub-model onto 16 state variables which together represent the complete configuration of the modeled cell: (1) metabolite, RNA, and protein copy numbers, (2) metabolic reaction fluxes, (3) nascent DNA, RNA, and protein polymers, (4) molecular machines, (5) cell mass, volume, and shape, (6) the external environment including the host urogenital epithelium, and (7) time. Each cellular state variable was implemented as a separate class. See Section C.2 for further discussion of the mathematics and computational implementation of each state variable.

Shared Resource Allocation

Second, to satisfy our sub-model independence assumption, at each time step we computationally allocated common sub-model inputs. At each simulation time step, prior to the evaluation of the sub-models, we estimated the metabolite resources required by each process and divided the total pool of each metabolite among the processes proportional to demand. Resource requirements were estimated by calculating the expected metabolite consumption of each process conditioned on the current cell configuration and an infinite metabolite supply. Algorithm C.1 outlines the shared metabolite allocation algorithm. Because macromolecules and enzymatic capacity are less heavily shared by the cellular processes, we chose not to implement similar procedures for these sub-model inputs.

Process Alignment & Parameter Fitting

Third, because the 28 cellular processes were trained using different experimental data obtained by different investigators under different conditions using different techniques and different model organisms, we refined the values of the sub-model parameters to make the processes mutually consistent. This was necessary for example, because amino acid production by the **Metabolism** process, which was trained using the observed amino acid composition of *M. gallisepticum* reported by Morowitz et al.²⁶¹, conflicted with the amino acid requirements of the **Translation** process, which was trained

using the observed *Mycoplasma* genetic code^{87,364,429}, the reported *M. pneumoniae* mRNA expression⁴³¹, and the N-end rule⁴⁰⁴. Specifically, Morowitz et al. reported that cysteine was only present in trace amounts, whereas the combination of the genetic code and observed mRNA expression is consistent with low, but not insignificant cysteine incorporation.

First, we manually identified constraints among groups of parameters spread across multiple processes. Among these constraints, we identified equality constraints between the cell chemical composition used to train the flux-balance analysis (FBA) metabolic objective and the expected metabolite requirements of the cellular processes. We also identified inequality constraints among the kinetic rate, expression, and required enzymatic capacity of each enzyme which describe the minimum expression of each gene consistent with all 28 sub-models.

Second, we computationally identified a set of parameter values which (1) satisfy all of these constraints and (2) deviate minimally from their experimentally observed values. Initially, we attempted to rigorously formulate this problem as a non-linear constrained optimization problem, and identify the parameter values which minimize the sum of squared differences between the adjusted and observed parameter values among all sets of parameters which satisfy the constraints. However, we were unable to find a feasible solution, much less the globally optimal solution, to this optimization problem. Instead, we focused on identifying a mutually consistent set of parameter values, and developed a heuristic procedure that uses the constraints to calculate a consistent set of parameter values from a subset of the parameters. This procedure primarily adjusted the gene expression. The adjusted gene expression correlated highly with that observed by Weiner et al.⁴³¹ ($R^2 = 0.68$, see Figure S1A). Section C.5.7 and the `FitConstants` class provide further discussion of the implementation of the parameter refinement procedure.

C.1.4 Initial Conditions

Cell theory³³⁰ states that all cells are created from old cells, or, more mathematically, that on the time scale of a single-generation, mother and daughter cells are statistically identical. This principle relates the initial and final cell state distributions of the whole-cell model, providing a rigorous way to define the initial cell state distribution in terms of the dynamic model.

We applied this principle in seven steps (see Algorithm C.2). First, we developed a method that calculates the expectation value of each cell state variable. Second, we approximated the distribution of each cell state variable by a standard, well-behaved statistical distribution, and set the mean of

each distribution to its calculated value. For example, we assumed that the copy number of each RNA and protein species is multinomially distributed, and that the total cell mass is normally distributed. Third, we parameterized the variance of the distribution of each cell state variable and initially set the variance of each distribution to zero. Fourth, we developed the procedure outlined in Algorithm C.3 to sample these distributions and set the initial value of each state variable. Fifth, we simulated a population of wild type cells using this cell state initialization procedure and calculated the variance of the final distribution of each cell state variable. Sixth, we set the initial variance of each state variable to its calculated final variance. Finally, we repeated steps five and six until convergence.

Algorithm C.2. Initial conditions identification algorithm.

Initialize the initial cell state variance control parameters: $\sigma_m \leftarrow 0$, $\eta_r \leftarrow 0$, $\eta_p \leftarrow 0$

repeat

- Simulate the life cycle of a population of wild type cells using Algorithm C.3 to initialize the value of each cell state variable
- Randomly segregate the cellular content into two daughter cells
- Calculate the variances of the total cell mass, RNA copy number, and protein copy number states
- Set the values of the initial distribution control parameters of each state equal to that of the final distribution
- $\sigma_m \leftarrow$ standard deviation of the final cell mass distribution
- $\eta_r \leftarrow \sigma_r^2/N_r$, where N_r and σ_r^2 are the mean and variance of the final RNA copy number distribution
- $\eta_p \leftarrow \sigma_p^2/N_p$, where N_p and σ_p^2 are the mean and variance of the final RNA copy number distribution

until the initial variance control parameters (σ_m , η_r , and η_p) converge

C.1.5 Reconstruction

The *M. genitalium* whole-cell model was based on a detailed reconstruction of *M. genitalium* physiology developed from over 900 primary sources, reviews, books, and databases. First, we reconstructed the organization of the *M. genitalium* chromosome including the locations of each gene, transcription unit, promoter, and protein binding site primarily based on studies by Weiner et al.⁴³⁰, Güell et al.¹⁴⁰, and the CMR genome annotation⁸⁷. We also reconstructed the affinity of RNA polymerase for each promoter based on the reported expression⁴³¹ and half-life²⁹ of each RNA species.

Second, we functionally annotated each gene beginning with the CMR⁸⁷ annotation. We annotated genes with additional information from the BioCyc¹⁸⁸, KEGG¹⁸³, NCBI²³², and UniProt⁷¹ genome annotations. To fill gaps in the reconstructed organism, such as observed reactions without reported enzyme catalysts, and to maximize the scope of the model, we also expanded and refined each gene's annotation using primary research articles and reviews identified by systematically searching

Algorithm C.3. Cell state initialization procedure.

Input: $\sigma_m \leftarrow$ standard deviation of the initial total cell mass distribution
Input: $\eta_r, \eta_p \leftarrow$ RNA and protein copy number distribution initial variance control parameters
Input: $f_r, f_p \leftarrow$ reconstructed fractional cell RNA and protein composition
Input: $e_r, e_p \leftarrow$ expected relative expression of each RNA and protein species
Input: $w_r, w_p \leftarrow$ molecular weight of each RNA and protein species
Input: $N_i(f_i, e_i, w_i) \leftarrow f_i m / (e'_i * w_i)$ total initial RNA ($i = r$) and protein ($i = p$) copy number functions
Set time $\leftarrow 0\text{s}$
Set values of external stimuli and metabolites according to Table S3F and S3H
Set total cell mass, $m \leftarrow \sim N(\mu, \sigma_m)$, and calculate cell volume and shape
Set the metabolite counts according to the total cell mass and reconstructed cell composition (see Table S3I)
Initialize the Chromosome state with one methylated chromosome; decrement dNMP counts to maintain cell mass
Set mature RNA copy numbers according to $\text{multinomialRand}(\eta_r N_r, e_r) / \eta_r$; decrement NMP counts
Set mature protein monomer copy numbers according to $\text{multinomialRand}(\eta_p N_p, e_p) / \eta_p$; decrement amino acids
Form macromolecules by calculating the steady-state of the Macromolecular Complexation process
Set the RNA Polymerase and Transcript states to a steady-state of the Transcription sub-model
Set the Ribosome and Polypeptide states to a steady-state of the Translation sub-model
Set the FtsZ state to a steady-state of the FtsZ Polymerization sub-model with no septal rings
Set the growth rate and metabolic reaction fluxes to a steady-state of the Metabolism sub-model
Set the Host state to a steady-state of the Host Interaction sub-model
Set the chromosome protein occupancy to a steady-state of the chromosome-interacting sub-models

PubMed and Google Scholar for each gene and homologs of each gene. Table S3J lists all functional annotations assigned beyond the CMR annotation. Additionally, we curated the reported essentiality¹³⁴ of each gene product.

Next, we curated the structure of each gene product, including the sequence of each protein, the post-transcriptional and post-translational processing and modification of RNA and protein, the signal sequence and localization of each protein, the DNA footprint of each DNA-binding protein, the chaperones and prosthetic groups required to fold each protein, the subunit composition of each protein and ribonucleoprotein complex, and the disulfide bonds of each protein and complex.

After annotating each gene, we categorized the genes into 28 cellular processes. We curated the chemical reactions of each cellular process with particular emphasis on reactions needed to interface the processes. For example, we added several metabolic reactions to provide the metabolites required for RNA modification. We also added metabolic reactions to catabolize modified nucleotides produced by the degradation of modified RNA. We reconstructed the stoichiometry and catalysis of each chemical reaction based on the databases BioCyc¹⁸⁸ and KEGG¹⁸³, a *M. genitalium* FBA metabolic developed by Suthers et al.³⁸⁹, and hundreds of additional primary research articles. We reconstructed the kinetics of each reaction primarily based on the databases BRENDA⁵⁹ and

SABIO-RK⁴³⁶.

We reconstructed the *M. genitalium* metabolome based on the substrates and products of the reconstructed chemical reactions, the observed chemical compositions of *M. gallisepticum*²⁶¹ and *E. coli*²⁷⁸, and the reported chemical composition of SP-4 *Mycoplasma* growth medium^{371,372,374–377}. We reconstructed the structure of each metabolite based on several metabolomic databases including PubChem⁴²⁶. We reconstructed the protein regulatory properties of each metabolite primarily based on DrugBank¹⁹².

Finally, we calculated the empirical formula, molecular weight, and several other physical properties of each metabolite, RNA, protein, and macromolecular complex using the **KnowledgeBase** classes, ChemAxon Marvin²³⁶, and ExPASy ProtParam⁴³².

Because *M. genitalium* is not well-studied, the *M. genitalium* reconstruction was primarily based on studies of *M. genitalium* homologs identified by bi-directional best BLAST. Where possible, the *M. genitalium* reconstruction was based on closely related organisms.

Table S3A-S3S define the reconstructed *M. genitalium* organism including the structure of every metabolite; the sequence of every RNA and protein; and the stoichiometry, kinetics, and catalysis of every reaction. Table S3T-S3BK describe the how the reconstructed organism was developed, including detailed notes on how the value of each reconstructed property was derived by consensus of all available experimental observations and computational predictions. Table C.1 lists the principal sources of the *M. genitalium* reconstruction; Table S3S provides a complete list of all the sources of the reconstruction. Table S2B-S2C list the computationally refined values of the reconstructed cellular composition and gene expression. See Section C.2 for further discussion of the reconstruction of each cell variable. See Section C.3 for further discussion of the reconstruction of each cellular process sub-model. See Section C.1.3 for further discussion of modeling fitting and computational refinement of the reconstruction.

C.1.6 Experimental Validation

The *M. genitalium* whole-cell model was validated by comparing the model’s predictions to three experimental datasets: (1) the essentiality of each *M. genitalium* gene reported by Glass et al.¹³⁴, (2) the measured growth rates of 12 non-essential *M. genitalium* single-gene disruption strains, and (3) the cytosolic concentrations of 39 *E. coli* metabolites reported by Bennett et al.²⁴ and curated

Table C.1. Primary sources of the *M. genitalium* reconstruction.

Data source	Content
Bernstein et al., 2002 ²⁹	mRNA half-lives
BioCyc ¹⁸⁸	Genome annotation, metabolic reactions
BRENDA ⁵⁹	Reaction kinetics
CMR ⁸⁷	Genome annotation
Deuerling et al., 2003 ⁹⁵	Chaperone substrates
DrugBank ¹⁹²	Antibiotics
Eisen et al., 1999 ¹¹²	DNA repair
Endo et al., 2007 ¹¹³	Chaperone substrates
Feist et al., 2007 ¹²⁰	Metabolic reactions
Glass et al., 2006 ¹³⁴	Gene essentiality
Güell et al., 2009 ¹⁴⁰	Transcription unit structure
Gupta et al., 2007 ¹⁴⁵	N-terminal methionine cleavage
KEGG ¹⁸³	Genome annotation, orthology
Kerner et al., 2005 ¹⁸⁷	Chaperone substrates
Krause et al., 2004 ¹⁹⁷	Terminal organelle assembly
Lindahl et al., 2000 ²¹⁸	DNA damage
Morowitz et al., 1962 ²⁶¹	Cell chemical composition
NCBI Gene ^{232,396}	Genome annotation
Neidhardt et al., 1990 ²⁷⁸	Cell chemical composition
Peil, 2009 ²⁹⁸	RNA modification
PubChem ⁴²⁶	Metabolite structures
SABIO-RK ⁴³⁶	Reaction kinetics
Solabia ^{371,372,374–377}	Media chemical composition
Suthers et al., 2009 ³⁸⁹	Metabolic reactions
UniProt ⁷¹	Genome annotation
Weiner et al., 2000 ⁴³⁰	Promoters
Weiner et al., 2003 ⁴³¹	mRNA expression

by Sundararaj et al.³⁸⁷. The *M. genitalium* whole-cell model also reproduces several experimental measurements which were used to train the model including the published cellular composition of *M. gallisepticum*²⁶¹, the measured RNA composition of *E. coli*²⁷⁸, the reported *M. pneumoniae* mRNA expression⁴³¹, the observed *E. coli* mRNA half-lives²⁹, and the measured growth rate of wild type *M. genitalium*.

To validate the model against the observed gene essentiality and the observed disruption strain growth rates, we first simulated the individual disruption of each gene. We ran 5 simulations of each single-gene disruption strain by (1) randomly initializing the cell state using Algorithm C.3, (2) deleting the *in silico* gene, and (3) calculating the temporal evolution of the cell state for the first generation post-disruption. Gene disruption was implemented in two steps: (1) we modeled insertion of a transposon of length zero which reduces the stability of the terminal products of the deleted gene, and set the half-life of the RNA and protein products of the deleted gene to zero; and (2) to more quickly highlight altered phenotypes, we deleted all RNA and protein products of the deleted gene. Next, we calculated the mean predicted mass doubling time, cell cycle length, terminal organelle protein mass, and damaged protein copy number of each disruption strain.

Third, we calculated the mean growth rate of each single-gene disruption strain at successive generations post-disruption. Rather than simulating the complete dynamics of successive generations, which was infeasible due to the significant computational cost of each simulation, we predicted only the growth rate of each disruption strain at successive generations post-disruption by initializing simulations using a modified version of the method described above. (1) We initialized cells using the wild type cell initialization method. (2) We deleted the *in silico* gene as previously described. (3) To simulate the long-term effects of the gene disruption and dilution resulting from cellular growth and division, we reduced the copy numbers of macromolecules which are normally synthesized by the deleted gene product. (4) We calculated the growth rate using the **Metabolism** process.

Fourth, we classified each *in silico* single-gene disruption strain as (quasi-)essential if the predicted first generation cell cycle length was significantly ($P \leq 0.01$) less than that of wild type *in silico* *M. genitalium*, if the terminal organelle protein mass was significantly ($P \leq 0.01$) less than that of wild type *in silico* *M. genitalium*, if the predicted damaged protein copy number was significantly ($P \leq 0.01$) greater than that of wild type *in silico* *M. genitalium*, or if the growth rate declined over successive generations. We found that the model reproduces the observed gene essentiality with 79% accuracy. Figure 6B of the accompanying manuscript illustrates these distinct disruption strain

phenotypes. Figure S2 illustrates the distribution of growth rates among wild type *M. genitalium* and the quasi-essential and essential single-gene disruption strains. Table S2G lists the predicted growth rate of each disruption strain.

Fifth, we compared the experimentally observed and predicted growth rates of 12 non-essential single-gene disruption strains (see Figure 7A of the accompanying manuscript and Table S1), and found that the model correctly predicts the measured growth rates of 67% of the disruption strains.

To validate the model against the Bennett et al.²⁴ and Sundararaj et al.³⁸⁷ datasets, we calculated the mean concentration of each cytosolic metabolite in a population of 128 wild type cells. Figure 2E of the accompanying manuscript and Table S2E compare the predicted and measured concentrations of 39 cytosolic metabolites, illustrating that 70% of the model’s predictions are statistically consistent with the Sundararaj et al. dataset. The model doesn’t reproduce the Bennett et al. dataset, and interestingly, there is significant disagreement between the Bennett et al. and Sundararaj et al. datasets.

C.1.7 Computational Implementation

The whole-cell model was implemented in MATLAB, and consisted primarily of the main **Simulation** class and one class for each cellular state and process. The computational correctness of the whole-cell model algorithm was validated using unit testing. The *M. genitalium* reconstruction was stored using a modified version of the BioWarehouse schema²¹³ in a MySQL relational database. The knowledge base was viewed and edited using a web-interface implemented in PHP. Several **KnowledgeBase** classes represented the knowledge base in MATLAB. Appendix C.5 provides further discussion of the whole-cell model architecture and it’s computational validation.

C.1.8 Computational Simulation & Analysis

We used the whole-cell model to simulate 192 wild type cells and 3,011 single-gene deletants. All simulations were performed with MATLAB R2010b on a 128 core Linux cluster. The predicted dynamics of each cell was logged at each time point and subsequently analyzed using MATLAB. Appendix C.5 provides further discussion of the execution, logging, and analysis of the whole-cell model.

C.2 Cellular State Methods

The whole-cell model used 16 state variables to represent the instantaneous configuration of *M. genitalium* and integrate the 28 modeled cellular processes. The 16 state variables represented seven areas of cellular physiology: (1) copy numbers of metabolites, RNA and proteins, (2) metabolic reaction fluxes, (3) nascent DNA, RNA, and protein polymers, (4) molecular machines, (5) cell-level properties, including mass, volume and shape, (6) nascent polymers of DNA, RNA and protein, and (7) time. This chapter provides detailed discussions of the mathematics and computational implementation of each state variable.

The **Metabolite**, **Rna**, **Protein Monomer**, and **Protein Complex** states represented the copy number of each metabolite, RNA, protein monomer, and macromolecular complex. The complement of metabolite species was indirectly reconstructed by reconstructing the chemical reactions of each cellular process. The RNA complement was primarily reconstructed from the *M. genitalium* genomic annotation⁸⁷, the experimentally defined operon structure of *M. pneumoniae*¹⁴⁰, and the reported complement of *E. coli* RNA polycistronic cleavages and non-coding RNA modifications. The protein complement was primarily reconstructed from the predicted localization and signal sequence of each protein gene product (see Table S3AM-S3AO), the observed chaperone interactions of *E. coli*^{95,187} and *B. subtilis*¹¹³, the observed complement of *M. genitalium* and *M. pneumoniae* protein modifications^{93,163,200,231,386}, the reported N-terminal methionine cleavage of *Shewanella oneidensis* MR-1¹⁴⁵, and the inferred subunit composition of each macromolecular complex (see Table S3AS).

The **Metabolic Reaction** state recorded the predicted flux of each metabolic reaction. The *M. genitalium* metabolic network was reconstructed as described in Section C.3.10.

The **Chromosome**, **Transcript**, **Polypeptide**, and **FtsZ Ring** states represented the configurations of the chromosome, nascent RNA and protein polymers, and FtsZ septal ring. The **Chromosome** state represented the polymerization, protein occupancy, and modification status of the chromosomes. The *M. genitalium* chromosome was sequenced by Fraser et al.¹²². Protein binding sites and DNA footprints were reconstructed from the primary literature and several databases (see Table S3M and S3N). DNA modification sites were predicted based on the reported DNA motif of the MuiI methylase⁷¹. The **Transcript** and **Polypeptide** states represented the sequence of each nascent RNA transcript and polypeptide. The sequence of each RNA and protein species was reconstructed as described in Section C.3. The **FtsZ Ring** state represented the configuration of the FtsZ septal

ring. The structure of the FtsZ septal ring was reconstructed based on the Li et al. iterative pinching model²¹⁶.

The **RNA Polymerase** and **Ribosome** states represented the detailed configuration of each RNA polymerase and ribosome. The **RNA Polymerase** state represented the status – free, non-promoter bound, promoter-bound, or actively transcribing – of each RNA polymerase molecule, and the chromosomal location and direction of each DNA-bound RNA polymerase. The affinity of RNA polymerase for each promoter was primarily reconstructed from the observed expression of each RNA gene product⁴³¹ and the observed half-life of each *E. coli* mRNA²⁹. The genetic code was reconstructed based on that of *M. pneumoniae*³⁶⁴.

The **Mass** state represented the total cell mass. The **Geometry** state represented the cell shape and volume. The average cell size, density, and mass were reconstructed from studies by Baldwin et al.¹⁶, Bray³⁹, and Zhao et al.⁴⁴⁷.

The **Metabolite**, **Stimulus**, **Host**, and **Geometry** states represented the configuration of the external environment, including the extracellular copy number of each metabolite which was reconstructed as described in Section C.3.10. The **Stimulus** state represented the temperature, the fluxes of six types of radiation, and the status of three common experimental stress conditions. Table S3F describes the reconstruction of the **Stimulus** state. The **Host** state represented four properties of the host human urogenital epithelium: (1) *M. genitalium* adherence, (2) activation of Toll-like receptors 1, 2, and 6, (3) activation of the host transcriptional regulator NF- κ B, and (4) activation of the host inflammatory response. The **Geometry** state represented the volume of the extracellular environment.

Finally, the **Time** state represented the time elapsed from the beginning of the simulation.

C.2.1 Chromosome

Biology

Chromosomes encode the structure and function of every RNA and protein, and thereby control cellular behavior. Due to their critical function and large size, cells dedicate considerable resources to chromosome replication, maintenance, and compaction. This state represents the polymerization, winding, modification, protein occupancy, and (de)catenation status of the chromosome(s).

Reconstruction

Structure and Organization

The reconstructed *M. genitalium* chromosome contains 525 genes based on the Comprehensive Microbial Resource (CMR) genomic annotation⁸⁷ (see Table S3J) organized into 335 transcription units based on the *M. pneumoniae* operon organization experimentally defined by Güell et al.¹⁴⁰ and the *M. genitalium* promoters computationally predicted by Weiner et al.⁴³⁰ (see Table S3U). The reconstructed genome also contains 17 transcriptional regulatory elements based on 15 studies and databases^{9,14,61,64,71,114,140,266,273,275,279,293,297,362,449} (see Table S3P), 2,283 DnaA binding sites computationally identified based on the consensus binding motif reported by Grimwade et al.¹³⁸ and Speck et al.³⁷⁹ (see Table S3L), 760 MunI restriction/modification (R/M) sites computationally identified based on the reported MunI binding motif⁷¹ (see DNA Repair process), and 19 short tandem repeats identified by Ma et al.²³⁰ and Washio et al.⁴²⁸ (see Table S3V). Additionally, the location of the replication origin was reconstructed based on studies by Lobry²²², Jensen et al.¹⁶⁸, and others^{315,385}, and the predicted DnaA box sites. Finally, the reconstructed steady-state superhelicity is based on the observed equilibrium helical repeat length of plasmid DNA⁴²⁵ and the observed superhelicity of *E. coli* DNA²¹⁹.

Protein Binding

The binding motif and footprint of each DNA-binding protein was reconstructed from several experimental studies^{45,86,123,124,153,168,245,251,264,274,285,327,335,351,361,382,383,385,418–420,433} and the databases 3D-Footprint⁷², NDB²⁸, and ProNIT²⁰⁵. DNA-bound protein displacement reactions were assembled describing which proteins each protein species is able to displace from the chromosome (see Table S3O). Functionally, displacement of DNA-bound proteins enables proteins to access the chromosome and fulfill their chromosomal replication or maintenance role. The affinity of RNA polymerase for each promoter was first reconstructed from the observed RNA composition of *E. coli*²⁷⁸ (see Table S3U), the expression of each *M. genitalium* mRNA reported by Weiner et al.⁴³¹ (see Table S3W), the half-life of each *E. coli* mRNA reported by Bernstein et al.²⁹ (see S3Y, the observed amino acid composition of *M. gallisepticum*²⁶¹, and the *Mycoplasma* genetic code^{87,364,429} (see Table S3X). Subsequently, the affinity of RNA polymerase for each promoter was fit to match the additional data used to train the 28 modeled cellular processes (see Section C.1.3).

Computational Representation

This state represents the polymerization, winding, modification, and protein occupancy of each nucleotide of each strand of each copy of the *M. genitalium* chromosome, and the (de)catenation status of the two sister chromosomes following replication. Table C.2 summarizes the mathematical representation of the *M. genitalium* chromosome(s) including the size and type of each variable.

Mathematically, each quantity except winding, base and sugar-phosphate modification, protein occupancy and catenation, is represented as a 3-dimensional Boolean tensor. Winding is represented as a 3-dimensional real tensor which indicates the linking number density of each nucleotide. Base and sugar modification are represented as 4-dimensional tensors which indicate the chemical identity $l = \{1..M\}$ of each nucleotide, where $M = 722$ is the number of distinct metabolite species represented by the **Metabolite** state. Protein monomer and complex occupancy are represented as 4-dimensional tensors which indicate the identity, $l = \{1..B^m\}$ and $l = \{1..B^c\}$ respectively, of the protein bound at each nucleotide, where $B^m = 482$ and $B^c = 201$ are the numbers of distinct protein monomer and complex species represented by the **Protein Monomer** and **Protein Complex** states. (De)catenation is represented as a Boolean scalar.

Table C.2. Mathematical representation of nucleotide $i = \{1..L\}$ of strand $j = \{1..2\}$ of chromosome copy $k = \{1..2\}$.

Physical Property	Symbol	Size	Type
Polymerization	p_{ijk}	$L \times 2 \times 2$	Boolean
Winding	w_{ijk}	$L \times 2 \times 2$	Real
Modification			
Gap site	m_{ijk}^g	$L \times 2 \times 2$	Boolean
Abasic site	m_{ijk}^a	$L \times 2 \times 2$	Boolean
Sugar-phosphate	m_{ijkl}^p	$L \times 2 \times 2 \times M$	Boolean
Base	m_{ijkl}^b	$L \times 2 \times 2 \times M$	Boolean
Intrastrand cross link	m_{ijk}^c	$L \times 2 \times 2$	Boolean
Strand break	m_{ijk}^s	$L \times 2 \times 2$	Boolean
Holliday junction	m_{ijk}^h	$L \times 2 \times 2$	Boolean
Protein occupancy			
Monomer	b_{ijkl}^m	$L \times 2 \times 2 \times B^m$	Boolean
Complex	b_{ijkl}^c	$L \times 2 \times 2 \times B^c$	Boolean
Catenation	s	1×1	Boolean

Integration

The **Metabolite** state describes the identity of each modified base and sugar-phosphate. The **Protein Monomer** and **Protein Complex** states represent the total DNA-bound copy number of each

protein monomer and macromolecular complex. For computational efficiency, the **RNA Polymerase** state redundantly represents the chromosomal location of each bound RNA polymerase. The mass of the chromosome(s), including all modifications and all DNA-bound proteins, is included in the cell mass calculated by the **Mass** state.

Ten processes access and modify the **Chromosome** state. The **Replication Initiation** process models DnaA DNA-binding and formation of the oriC DnaA complex which promotes replication initiation. The **Replication** process models bidirectional DNA polymerization from the oriC, continuously on the leading strand and discontinuously on the lagging strand, and Okazaki fragment ligation. The **Chromosome Segregation** process models sister chromosome decatenation following successful chromosome replication. The **Cytokinesis** process models cell division following successful decatenation. The **Chromosome Condensation and DNA Supercoiling** processes model SMC- and supercoiling-mediated chromosome compaction. The **DNA Damage** process models stochastic and radiation- and chemically-induced DNA damage. The **DNA Repair** process models three DNA repair pathways – base excision repair, nucleotide excision repair, and homologous recombination. The **DNA Repair** process also models methylation and restriction of MunI (MG184) restriction/modification sites. Finally, the **Transcriptional Regulation and Transcription** processes model the binding of transcription factors and RNA polymerases to promoters and the synthesis of RNA.

Initial Conditions

First, the **Chromosome** state is initialized with one supercoiled and fully methylated chromosome,

$$\begin{aligned}
 p_{ij1} &= 1 \quad \forall i, j \\
 p_{ij2} &= 0 \quad \forall i, j \\
 w_{ij1} &= (1 + \sigma_{ss}) \frac{L}{h} \quad \forall i, j \\
 w_{ij2} &= 0 \quad \forall i, j \\
 m_{ijk}^g &= 0 \quad \forall i, j, k \\
 m_{ijk}^a &= 0 \quad \forall i, j, k \\
 m_{ijk}^p &= 0 \quad \forall i, j, k \\
 m_{ij1}^b &= \begin{cases} 1 & i \in \text{MunI methylation sites, } \forall j \\ 0 & \text{otherwise} \end{cases} \\
 m_{ij2}^b &= 0 \quad \forall i, j \\
 m_{ijk}^c &= 0 \quad \forall i, j, k \\
 m_{ijk}^s &= 0 \quad \forall i, j, k \\
 m_{ijk}^h &= 0 \quad \forall i, j, k \\
 s &= 0,
 \end{aligned}$$

where $L = 580076$ nt is the length of the *M. genitalium* chromosome, $\sigma_{ss} = -0.06$ is the observed bacterial steady-state superhelicity²¹⁹, and $h = 10.5$ nt lk⁻¹ is the observed equilibrium helical repeat length⁴²⁵.

Second, the **Protein Monomer** and **Protein Complex** states initialize the total copy number of each protein species. Finally, the **Chromosome Condensation**, **DNA Supercoiling**, **Replication Initiation**, **Transcriptional Regulation**, and **Transcription** processes initialize the protein occupancy of the chromosome.

Fitting

The chemical composition of *M. genitalium* and the objective of the flux-balance analysis metabolic model were fit to match the mass and dNMP composition of the *M. genitalium* chromosome (see

Section C.1.3 and C.2.5). The affinity of RNA polymerase for each promoter was fit to provide the gene products required by the 28 modeled processes to reproduce the observed 9 h *M. genitalium* mass doubling time (see Section C.1.3). The oriC DnaA DNA-binding cooperativity was fit to match our intuition for the duration of the replication initiation cell cycle phase (see Section C.3.19). The expression and activity of DNA gyrase and topoisomerase I were balanced to match the observed steady state superhelical density (see Section C.3.6).

C.2.2 FtsZ Ring

Biology

Cytokinesis is the division of a cell into two daughter cells. *M. genitalium* contains a protein called FtsZ that assembles into long filaments that are implicated in cell pinching. These filaments bind to the membrane at the midline of the cell⁷. FtsZ is a GTPase, and when the GTPs bound in a membrane-bound FtsZ filament hydrolyze to GDP, the filaments bend constricting the membrane¹⁵⁶. We use a geometric model of iterative filament bending modified from a model proposed by Li et al.²¹⁶. See the **Cytokinesis** process and Figure C.6, for more details about this model. In summary, the FtsZ ring at the midline of the cell can exist in many states of various numbers of filaments in the bent and straight configurations. The purpose of this state class is to keep track of the state of the FtsZ ring across timesteps.

Reconstruction

In our model, the FtsZ ring is represented as a polygon of FtsZ filaments. As the circumference of the midline of the cell (C) decreases with time, the number of edges in the inscribed FtsZ polygon decreases. The FtsZ filaments are of a fixed length (l) of 40 nm⁷.

Computational Representation

The number of edges in the polygon (E) is determined by the maximum number of edges of length l that can be inscribed in the given cell circumference. (The use of a fixed filament length is a simplification. See the **Cytokinesis** and **FtsZ Polymerization** processes for more details about the assumptions used.) This state class computes the number of edges in the FtsZ polygon using

the inscribed polygon formula and the current cell midline diameter (d):

$$E = \frac{\pi}{\arcsin(\frac{l}{d})} \quad (\text{C.1})$$

The filaments may not perfectly inscribe the cell circumference, and so we round down, such that E only accounts for full filaments in the FtsZ ring.

The **Cytokinesis** process determines how many straight and/or bent filaments reside at each edge of the FtsZ polygon. At each edge, any of the following occupancies are possible:

- 1 straight filament
- 2 straight filaments
- 1 bent filament
- 2 bent filaments
- 1 bent filament and 1 straight filament
- 1 bent filament and 2 straight filaments

The **FtsZ Ring** state class keeps track of the filament occupancy of each edge. As the filament occupancy in a timestep is highly dependent on the occupancy at the previous timestep, the tracking of the state of the FtsZ ring is extremely important in the time evolution of **Cytokinesis**.

Integration

The **FtsZ Ring** state class reads the diameter of the cell's midline (d) from the **Geometry** state class.

The **Cytokinesis** process class reads the number of FtsZ ring edges (E) and the filament occupancy at each edge from the **FtsZ Ring** state class. The **Cytokinesis** process class updates the filament occupancy at each edge, and this is stored in the **FtsZ Ring** state class.

Initial Conditions

An FtsZ ring is not assembled at the beginning of the simulation. Therefore, no initialization steps are required for this state class.

C.2.3 Geometry

Biology

While *M. genitalium* divides by binary fission similarly to other bacterial species, its non-uniform shape, lack of cell wall, and lack of complete division machinery make its growth and division distinctive from most other bacteria. *M. genitalium* has a flask/pear-like shape, with a protruding adhesion structure called the terminal organelle. This flask-like shape is rather fluid due to the lack of a cell wall³³⁶.

Reconstruction

Since the growth and division of *M. genitalium* are not well understood, we chose to model the cell as varying from a spherical to short rod shape. The cell is modeled as a cylinder with two hemispherical caps, and growth is modeled in the cylinder length. Once cell pinching commences at the midline of the cell, the shape and size of a “septum region” is also modeled. The **Geometry** state class calculates and keeps track of the physical shape of the cell including its width, length, volume, and surface area. This state also keeps track of the progress of cytokinesis, and the completion of cytokinesis is the trigger to end the entire simulation.

Computational Representation

The **Geometry** state uses a set of geometric equations to calculate the shape of the cell, given the width (w), density (ρ), and cytokinesis progress of the cell. The state houses the calculations of cell length (l), volume (V), surface area (SA), and progress of cell pinching. The geometric representation requires us to assume that the cell density and cell width are constant across the cell cycle. Various sources have indicated that the volumetric density of a cell remains constant overtime^{235,254}. The cell density used is that of *E. coli*, which has been estimated as 1100 g/L¹⁶. Several sources have described the width of a rod-shaped bacterial cell as remaining approximately constant across the cell cycle and across cell divisions¹²⁵. The cell width is calculated based on the initial cell mass (m_0) and density and the assumption that the cell is a sphere. The initial cell mass was fit to result in the measured *M. genitalium* cell width of 200 nm²¹⁷. The general geometric equations to represent the shape of a cell are inspired by Domach et al.⁹⁸. These geometric equations use the fixed width and fixed cell density assumptions to calculate all the other aspects of the cell geometry.

Cell Geometry Calculations

The mass (m) and density (ρ) enable calculation of the volume at all time points:

$$V = \frac{m}{\rho} \quad (\text{C.2})$$

Before cell pinching starts

The cell is modeled as a cylinder (length: l_c , diameter: w) with two hemispherical caps (diameter: w). We use the cell volume to calculate the cell length and surface area:

$$V = \underbrace{\frac{1}{6}\pi w^3}_{\text{2 hemispheres}} + \underbrace{\frac{1}{4}\pi w^2 l_c}_{\text{cylinder}} \quad (\text{C.3})$$

$$l = \underbrace{w}_{\text{2 hemispheres}} + \underbrace{l_c}_{\text{cylinder}} \quad (\text{C.4})$$

$$= w + \frac{4}{\pi w^2} \left(V - \frac{1}{6}\pi w^3 \right) \quad (\text{C.5})$$

$$SA = \underbrace{\pi w^2}_{\text{2 hemispheres}} + \underbrace{\pi w l_c}_{\text{cylinder}} \quad (\text{C.6})$$

After cell pinching starts

Once cell pinching commences at the midline of the cell, there is a “septum region” as well (Figure C.1). Here the cylinder length (l_c), is the combined length of the two cylinders. The septum length (s) is the length from the cell midpoint to the edge of the septum region. This septum length is calculated from the “pinched diameter” property that is calculated in the **Cytokinesis** Process. Each half of the septum volume is calculated as the area of two quarter circles (radius: s) and one rectangle (width: s , height: $w - 2s$), integrated around the midline cylindrically (see Figure C.2). We use the cell volume to calculate the cell length and surface area:

$$V = \underbrace{\frac{1}{6}\pi w^3}_{\text{2 hemispheres}} + \underbrace{\frac{1}{4}\pi w^2 l_c}_{\text{2 cylinders}} + \underbrace{\frac{s\pi}{2} \left((8s^2 - 4sw + w^2) + s\pi(w - 2s) - \frac{4}{3}s^2 \right)}_{\text{septum region}} \quad (\text{C.7})$$

$$l = \underbrace{w}_{\text{2 hemispheres}} + \underbrace{l_c}_{\text{2 cylinders}} + \underbrace{2s}_{\text{septum region}} \quad (\text{C.8})$$

$$= w + \frac{4}{\pi w^2} \left(V - \frac{1}{6}\pi w^3 - \left(\frac{s\pi}{2} \left((8s^2 - 4sw + w^2) + s\pi(w - 2s) - \frac{4}{3}s^2 \right) \right) \right) + 2s \quad (\text{C.9})$$

$$SA = \underbrace{\pi w^2}_{\text{2 hemispheres}} + \underbrace{\pi wl_c}_{\text{2 cylinders}} + \underbrace{4\pi s(w - s)}_{\text{septum region}} \quad (\text{C.10})$$

Once the pinched diameter is zero, this state records that the cell has divided, and this determines the end point of the simulation.

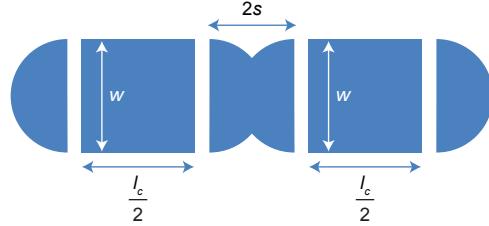


Figure C.1. Model representation of cell geometry.

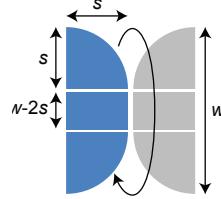


Figure C.2. Representation of volume of septum region.

Integration

The **Geometry** state class obtains the initial mass of the cell (to determine the cell width) from the **Mass** state class. It also obtains the mass of the cell from the **Mass** state class at all time points to determine the cell volume.

The **Metabolism**, **Protein Activation**, **Replication Initiation**, and **FtsZ Polymerization** process classes all obtain the cell volume from the **Geometry** state class. The **Cytokinesis** process class

reads the pinched diameter from the `Geometry` state class, and sends back the updated pinched diameter.

Initial Conditions

The initial cell volume is calculated based on the initial cell mass (m_0) and density (ρ). The initial cell width is calculated from the volume and the assumption that the cell is a sphere. All of the other parameters describing cell geometry are evaluated from the mass, volume, density, and width.

C.2.4 Host

Biology

As discussed in Section C.3.8, the *M. genitalium* terminal organelle is believed to mediate attachment to the human urogenital epithelium and enable its parasitic lifestyle. Upon attachment, lipoproteins are believed to activate host Toll-like receptors (TLRs) 1, 2, and 6, the host transcriptional regulator NF- κ B, and ultimately the host inflammatory response. This state represents the instantaneous configuration of the human host. The `Host Interaction` process describes the host response to *M. genitalium*.

Reconstruction

Section C.3.8 describes the reconstruction of the interaction of *M. genitalium* with its human host.

Computational Representation

This state uses six Boolean variables to represent the instantaneous configuration of the human host:

- Adherent – true if *M. genitalium* is attached to its human host and false otherwise.
- TLR 1 activation – true if TLR receptor 1 is active and false otherwise.
- TLR 2 activation – true if TLR receptor 2 is active and false otherwise.
- TLR 6 activation – true if TLR receptor 6 is active and false otherwise.
- NF- κ B activation – true if NF- κ B is active and false otherwise.
- Inflammatory response activation – true if the host inflammatory response is active and false otherwise.

Integration

None of the other 15 states directly interact with the Host state. The Host Interaction process completely determines the initial conditions and temporal dynamics of all six Boolean variables represented by this state.

Initial Conditions

As discussed in Section C.3.8, this state is initialized to the steady-state of the host-interaction dynamic model.

C.2.5 Mass

Biology

Cells are composed primarily of water, DNA, RNA, and protein enclosed in a bilipid membrane. The Metabolite, Chromosome, Rna, Protein Monomer, and Protein Complex states represent the detailed molecular composition of *M. genitalium*. The Mass state calculates the total cell mass from those states. The total cell mass is used as a proxy for membrane surface area in several processes which are based on mathematical models originally developed outside our integrated whole-cell modeling framework without calculations of the membrane surface area.

Reconstruction

The mass and chemical composition of *M. genitalium* were reconstructed based on an extensive review of the primary literature (see Table S3I, S3AR-S3BE) and fit to match the 28 modeled cellular processes. First, the average initial *M. genitalium* cell mass (13.1 fg) and dry mass (3.93 fg) were self-consistently calculated assuming a spherical geometry with 200 nm diameter⁴⁴⁷, 1.1 g ml⁻¹ density¹⁶, and 70% water composition by mass³⁹. Note this is comparable to the 18.9 fg *M. genitalium* cell mass reported by Morowitz²⁶⁰. Second, the molecular composition of the dry mass was hierarchically reconstructed:

1. The *M. genitalium* dry mass was divided into eight classes – DNA/dNMPs, RNA/NMPs, protein/amino acids, lipids, carbohydrates, polyamines, vitamins & cofactors, and ions.
2. The fractional mass (see Table S3AS) and molecular composition (see Table S3AU-S3BE) of each class was reconstructed from the literature.

3. The complete molecular composition of *M. genitalium* was assembled (see Table S3I and S3AT).
4. The dry mass composition was fit to match the 28 modeled cellular processes as described below and in Section C.1.3.

Computational Representation

This state calculates the total cell mass. The total cell mass is equal to the sum of that represented by the **Chromosome**, **Rna**, **Protein Monomer**, **Protein Complex**, **Transcript**, **Polypeptide**, and **Metabolite** states.

Integration

This state interacts with the **Geometry** state which calculates the cell shape, surface area, and volume from the calculated cell mass and observed density. None of the processes directly modify the **Mass** state. The **Metabolism** process uses the cell mass to bound the rates of exchange reactions. The **Replication Initiation** process models DnaA-ADP reactivation to DnaA-ATP as a function of membrane lipid mass. The **Metabolism** and **Replication Initiation** processes use the cell mass rather than the cell volume because these processes are based on mathematical models that were originally developed outside our whole-cell modeling framework without more detailed calculations of the membrane surface area.

Initial Conditions

The cell mass is hierarchically initialized. First, the total cell mass is initialized according to a normal distribution with mean 13.1 fg and standard deviation 0.66 fg. The mean initial cell mass was set to the reconstructed value. The variance was fit to match that of the predicted final cell mass following cell division. Second, the total cell mass is divided among individual metabolites according to the reconstructed chemical composition (see Section C.2.7). Third, the **Chromosome**, **Rna**, **Protein Monomer**, **Protein Complex**, **Transcript**, and **Polypeptide** states are initialized as functions of the metabolite copy numbers, and the metabolite copy numbers are decremented to maintain the initial cell mass.

Fitting

The mean initial cell mass was fit to the reconstructed value. The variance was fit to match that of the predicted final cell mass following cell division.

Cellular composition is the balance of the production and usage of molecules by cellular processes. Consequently, it was necessary to reconcile the reconstructed cellular composition with that predicted by the 28 modeled cellular processes. See Section C.1.3 for further discussion. First, the DNA dry mass fraction, including both chromosomal DNA and free dNTP, was increased 322% to be consistent with the predicted mass of the chromosome and the free dNTP pool. The predicted chromosome mass accounted for the observed chromosome sequence³⁰⁴, predicted modifications (see **DNA Repair** process), and predicted time-average copy number (see **Replication**, **Replication Initiation**, and **Metabolism** processes). The predicted free dNTP pool accounted for the free dNTPs needed support the observed 9 h *M. genitalium* mass doubling time. Second, the RNA dry mass fraction, including both RNA and free NTPs, was increased 39% to be consistent with the amounts of m-, r-, s-, and tRNA needed to support the observed 9 h mass doubling time. The NMP composition of the RNA fraction was set equal to that predicted by the expression, sequence, and modification of each RNA species (see **Transcription**, **RNA Processing**, and **RNA Modification** processes). Third, the vitamin & cofactor and ion dry mass fractions were increased to match the amounts of vitamins, cofactors, and ions needed for protein folding and catalysis (see **Protein Folding** process). Finally, the protein dry mass fraction, including both proteins and free amino acids, was decreased 23% to offset the increased DNA and RNA mass fractions. The amino acid composition of the protein fraction was set equal to that predicted by the expression, sequence, and modification of each protein species (see **Translation**, **Protein Processing I**, **Protein Processing II**, and **Protein Modification** processes).

C.2.6 Metabolic Reaction

Biology

Cells grow by importing nutrients from the external environment and using those nutrients to construct macromolecules. The **Metabolism** process models the dynamics of the 645 transport and chemical reactions which provide the metabolic building blocks required for macromolecular synthesis and drive cellular growth. Table S3O lists the reconstructed transport and chemical reactions.

Reconstruction

The *M. genitalium* metabolic network was reconstructed based on extensive review of the primary literature, several databases, and the metabolic demands of all of the processes. See Section C.3.10 for further discussion.

Computational Representation

This state records the instantaneous flux of each metabolic reaction in reactions per second as a floating point vector. This state also records the instantaneous cellular growth rate predicted by the **Metabolism** process in cells per second as a floating point scalar.

Integration

None of the other 15 states directly interact with the **Metabolic Reaction** state. The **Metabolism** process calculates the flux of each metabolic reaction and the total cellular growth rate using flux-balance analysis (FBA). The **Metabolism** process uses the predicted fluxes to update the copy number of each metabolite.

Initial Conditions

The **Metabolism** process initializes the flux of each metabolic reaction and the total cellular growth rate to a steady-state of the FBA model.

Fitting

The FBA objective was set to the reconstructed *M. genitalium* chemical composition (see **Mass** and **Metabolite** states and **Metabolism** process). The cellular growth rate was fit to match the observed 9 h *M. genitalium* mass doubling time using a modified version of minimization of metabolic adjustment (MOMA)³⁵² (see Section C.1.3 and **Metabolism** process).

C.2.7 Metabolite

Biology

The *M. genitalium* model accounts for the dynamics of 722 distinct metabolites (see Table S3G) which serve many important functions in over 1,100 chemical reactions across three compartments – cytosol, membrane, and extracellular space. First, cells use nucleic and amino acids to synthesize DNA, RNA, and protein. Cells also use ions and other prosthetic groups to stabilize macromolecules. Second, cells use small molecule bonds and gradients to store energy and drive cellular processes. In particular, cells drive many energetically unfavorable reactions through hydrolysis of the high energy intermediates ATP and GTP. Third, cells use coenzyme functional moieties to facilitate chemical catalysis. Additionally, cells use small molecules such as Ca^{2+} , ATP, and (p)ppGpp for communication and regulation. Finally, cells use small molecule antibiotics to defend against predators and attack prey.

Reconstruction

The *M. genitalium* metabolite complement was indirectly reconstructed by reconstructing the chemical reactions of each of the 28 modeled cellular processes. Table S3G lists the 722 reconstructed metabolites.

Metabolite Physical Properties

The empirical formula and structure of each metabolite was curated based on an extensive review of the primary literature including two genome-scale metabolic models of *M. genitalium*³⁸⁹ and *E. coli*¹²⁰, and the databases BioCyc¹⁸⁸, ChEBI²³⁸, Delta Mass⁹², FindMod¹²¹, KEGG¹⁸³, LIPID MAPS¹¹⁶, Modomics¹⁰², PubChem⁴²⁶, RESID¹²⁷, and UniMod⁸⁰. The molecular weight, van der Waals volume, pI, logP, and logD of each metabolite was computed using ChemAxon Marvin²³⁶. Table S3G lists the physical properties of each metabolite.

Metabolite Regulatory Properties

The regulatory properties of several antibiotics were reconstructed from the primary literature and the database DrugBank¹⁹². See Protein Activation process for further discussion.

Cellular Chemical Composition

The molecular composition of *M. genitalium* was reconstructed based on an extensive review of the primary literature. See Section C.2.5 for further discussion.

Extracellular Medium Chemical Composition

M. genitalium was cultured in complex *Spiroplasma* medium #4 (SP-4 medium)⁴¹⁴ for all experiments presented in this study. Accordingly, the chemical composition of the *in silico* external environment modeled that of SP-4 medium. Because the chemical composition of SP-4 medium is undefined, the composition of *in silico* medium was reconstructed based on the characterized composition of each SP-4 medium component^{1,6,348,371,372,374–377} (see Table S3BJ), and supplemented with additional metabolites to support *in silico* growth. Addition of supplemental metabolites was guided by the **Metabolism** process and by the *M. pneumoniae* minimal medium defined by Yus et al.⁴⁴⁵. Table S3BI lists the composition of the *in silico* medium.

Computational Representation

This state represents the copy number of each metabolite in each of 3 compartments – cytosol, membrane, and extracellular space – as an integer matrix.

Integration

At each time step of the simulation, the **Simulation** object simulates an *M. genitalium* cell culture incubator which maintains the extracellular partial pressures of carbon dioxide and oxygen by setting the dissolved extracellular copy numbers of these gases. Table S3BI lists the simulated partial pressures of carbon dioxide and oxygen. To satisfy the assumption that each process is independent on the time scale of the 1 s simulation time step, and because many metabolite species participate in multiple processes, the **Simulation** object also allocates shared metabolites among processes at each time step. See Section C.1.3 for further discussion.

Cytosolic- and membrane-localized metabolites are included in the cell mass calculated by the **Mass** state. The **Metabolism** process models the import of extracellular nutrients into the cytosol and membrane, the conversion of those metabolites into the precursors of DNA, RNA, protein, and lipids, and the export of byproducts into the extracellular space. The exchange rate of each metabolite

of the flux-balance analysis metabolic model is limited by its extracellular copy number. 23 additional processes access and modify the **Metabolite** state, primarily drawing cytosolic metabolites to support macromolecule synthesis. Four processes do not directly interact with the **Metabolite** state: **Host Interaction**, **Macromolecular Complexation**, **Terminal Organelle Assembly**, and **Transcriptional Regulation**.

Initial Conditions

After the **Mass** state initializes the total cell mass, the **Metabolite** state initializes the total cytosolic and membrane copy numbers of each metabolite according to the calculated chemical composition of *M. genitalium* (see Section C.2.7 and C.2.7 and Table S3I). Additionally, after the **Geometry** state initializes the volume of the extracellular compartment, the **Metabolite** state initializes the extracellular copy number of each metabolite according to the reconstructed medium composition (see Section C.2.7 and C.2.7 and Table S3H). Following **Metabolite** state initialization, the **Chromosome**, **Rna**, **Protein Monomer**, **Protein Complex**, **Transcript**, and **Polypeptide** states initialize the copy number of each macromolecule species, and decrement the metabolite copy numbers to maintain the initial cell mass.

Fitting

The chemical compositions of *M. genitalium* and the extracellular medium were initially reconstructed based on the experimentally characterized compositions of *M. genitalium* and SP-4 medium. The cell and medium composition were both supplemented to support the metabolic demands of the 28 modeled cellular processes. See Section C.1.3, C.2.5, and C.2.7 for further discussion.

C.2.8 Polypeptide

Biology

The **Polypeptide** state class keeps track of the progress of translation. In our model, binding of free ribosomes to mature mRNAs is determined by mRNA availability. Once bound, a polypeptide is synthesized at a rate of up to 16 amino acids per second⁴⁴³. Therefore, polypeptides may take multiple 1 second timesteps to be completed. The **Polypeptide** state class holds information about which mRNAs are ribosome bound, as well as ribosomal progress of translating a transcript across timesteps.

For reasons such as limited resources, a ribosome may stall resulting in an incomplete polypeptide. In such cases, a proteolysis tag (a short peptide added to the end of a nascent polypeptide) may be synthesized to mark the ribosome for release and the aborted polypeptide for degradation. This state also houses the progress of proteolysis tag synthesis and the amino acid sequences of the incomplete polypeptides that are to be degraded.

Reconstruction

The **Polypeptide** state class serves as a “support system” for translation, providing it with information that is required to translate polypeptides.

The state holds fixed information such as the length, tRNA sequence, and amino acid sequence of every *M. genitalium* monomer and the length, molecular weight, tRNA sequence, and amino acid sequence of every possible *M. genitalium* proteolysis tag.

Computational Representation

As the simulation progresses, this state class holds transient information such as the ribosome-bound mRNAs, the lengths of polypeptides that are being translated, and the sequences of aborted polypeptides.

The state can calculate information from the transient properties such as the counts of each amino acid in each polypeptide and the length of aborted sequences. Lastly, this state is able to calculate the weight of all nascent polypeptides as the sum of the weights of the amino acid components.

Integration

The **Translation** process class reads all of the fixed parameters describing polypeptides and proteolysis tags. It both reads and updates the time evolving parameters describing polypeptides and proteolysis tags. The **Protein Decay** process class reads the sequence and length of each aborted polypeptide.

Initial Conditions

The simulation begins in a state in which ribosomes are already bound to mRNAs and in the process of elongating. Each growing polypeptide is accounted for in the **Polypeptide** state class.

No proteolysis tags exist at the start of the simulation.

C.2.9 Protein Complex

Biology

Protein complexes are used by almost all of the process classes in the system to perform their respective functions. A protein complex could be as small as transketolase, a dimer, or as large as pyruvate dehydrogenase that has 192 monomeric subunits. A complex may contain a mixture of protein and RNA subunits (e.g. ribosome), metabolites (e.g. DnaA protein bound to ATP), or ions (e.g. oxidized form of the protein thioredoxin). A complex can also exist in various locations within a cell including in the cytoplasm, membrane, terminal organelle, and bound to the chromosome. The **Protein Complex** state class holds information about complex composition, as well as the counts of each complex in the cell.

Reconstruction

The **Protein Complex** state class holds fixed parameters including the complex subunit composition, molecular weights, amino acid composition, half-lives, and localization. Some essential functions require a certain threshold abundance of particular protein complexes, and so this state class also holds the minimum expression of each complex.

Computational Representation

As complexes are produced by the Macromolecular Complexation process and other processes in the model, their counts are stored in this state. This state can also compute the dry weight of each protein complex in the cell.

Integration

Table C.3. Connections between the Protein Complex state class and other processes in the cell.

Connected Processes	Read from Protein Complex	Written to Protein Complex
Macromolecular Complexation	<ul style="list-style-type: none"> • Counts of protein complexes • Complex subunit composition 	<ul style="list-style-type: none"> • Updated counts of protein complexes
Ribosome Assembly	<ul style="list-style-type: none"> • Counts of ribosomes 	<ul style="list-style-type: none"> • Updated counts of ribosomes
Replication	<ul style="list-style-type: none"> • Counts of replication complexes 	<ul style="list-style-type: none"> • Updated counts of replication complexes
Replication Initiation	<ul style="list-style-type: none"> • Counts of DnaA complexes 	<ul style="list-style-type: none"> • Updated counts of DnaA complexes
Protein Decay	<ul style="list-style-type: none"> • Counts of protein complexes • Complex half lives • Complex amino acid composition 	<ul style="list-style-type: none"> • Updated counts of protein complexes
Transcription	<ul style="list-style-type: none"> • Minimum average expression 	
Various other processes	<ul style="list-style-type: none"> • Protein complex counts to determine maximal enzyme activity 	

Initial Conditions

The system is initialized with a set of proteins. The determination of which proteins are expressed and at what quantities is determined randomly based on expected protein expression and expected total protein mass fraction.

C.2.10 Protein Monomer

Biology

Protein Monomers are the direct result of successful translation events. Upon Translation, a monomer undergoes various steps towards maturation including deformylation, translocation, folding, and phosphorylation. As a result, a monomer can exist in many forms (nascent, processed (I),

translocated, processed (II), folded, and mature) as it moves through the maturation pipeline (See Figure C.3).

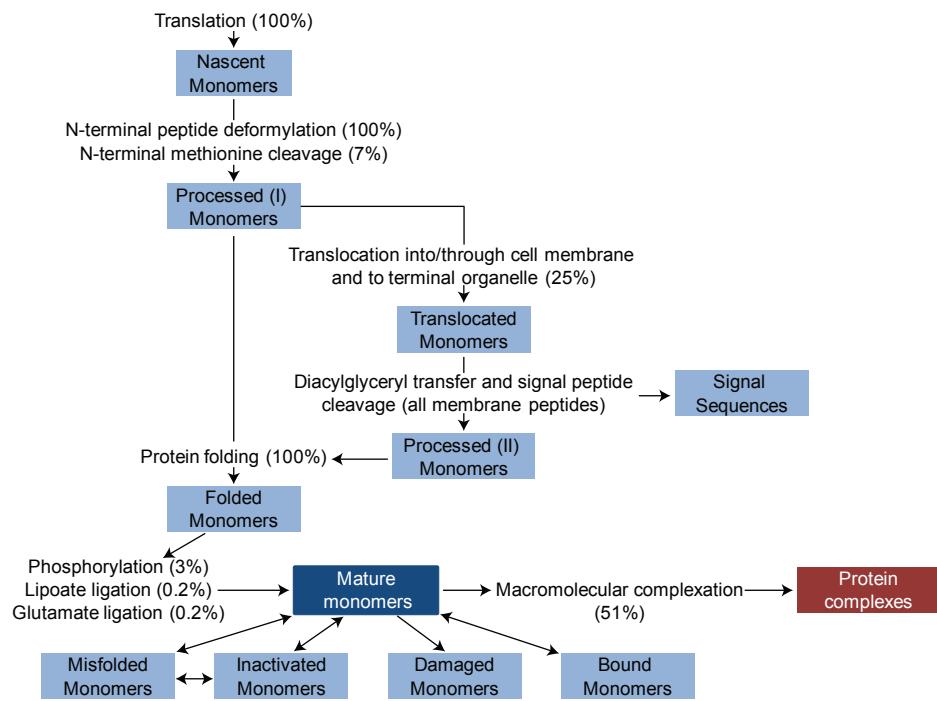


Figure C.3. Protein monomer forms diagrammed in the context of the maturity pipeline.

In addition to maturation, various processes can render a monomer non-functional, resulting in the misfolded, inactive, or damaged forms. Further, a mature monomer may be freely floating in the cytoplasm or bound to another molecule in the cell. For example, a translation factor may be bound to a mRNA, or a topoisomerase may be bound to a chromosome. The quantities of monomers in each of the maturation phases, functional forms, and non-functional forms are stored in this state class. Note that some functional enzymes are a complex of monomers, and these complexes are stored in a separate state class called **Protein Complex**.

Reconstruction

The main purpose of the **Protein Monomer** state class is to hold the counts and attributes of the monomeric species in the system.

This state class holds important information describing protein monomers such as their molecular weights, amino acids composition, length (in amino acids), and half-lives. A subset of monomers are

translocated to different compartments in our model such as the membrane or terminal organelle. The **Protein Monomer** state class contains information about where each monomer localizes.

The **Protein Monomer** state class also houses information that is important for the fitting and initialization of the model. Some essential functions require a certain abundance of particular proteins. For example, the Cell Division process may require at least some threshold abundance of the division protein FtsZ for cell division to ever be possible. The system is then fit such that in an unperturbed state, at least this threshold amount of FtsZ is produced. This state stores the minimum expression of each monomer.

The current version of this model involves a stochastic generation of the initial protein abundances in the cell. The initial abundance of certain monomers is more rigid than others for the maintenance of a stable system that can grow and divide. The **Protein Monomer** state class stores the degree of variation we allow in the initial abundances of each monomer.

Computational Representation

The **Protein Monomer** state class holds the counts of the monomeric species in each of the forms shown in blue in Figure C.3.

Integration

Table C.4. Connections between the ProteinMonomer state class and other processes in the cell.

Connected Processes	Read from Protein Monomer	Written to Protein Monomer
Translation	• Counts of nascent monomers	• Updated counts of nascent monomers
Protein Processing I	• Counts of nascent and processed (I) monomers	• Updated counts of nascent and processed (I) monomers
Protein Translocation	• Counts of processed (I) and translocated monomers	• Updated counts of processed (I) and translocated monomers
Protein Processing II	• Counts of translocated and processed (II) monomers and signal sequences	• Updated counts of translocated and processed (II) monomers and signal sequences
Protein Folding	• Counts of processed (II) and folded monomers	• Updated counts of processed (II) and folded monomers
Protein Modification	• Counts of folded and mature monomers	• Updated counts of folded and mature monomers
Macromolecular Complexation	• Counts of mature monomers	• Updated counts of mature monomers
Protein Decay	• Counts of monomers • Monomer half lives • Monomer amino acid composition	• Updated counts of monomers • Assignment of damaged monomers • Assignment of misfolded monomers
Protein Activation	• Counts of mature and inactivated monomers	• Updated counts of mature and inactivated monomers
Various other processes	• Counts of mature and bound monomers • Protein monomer counts to determine maximal enzyme activity	• Updated counts of mature and bound monomers

Initial Conditions

The system is initialized with a set of proteins. The determination of which proteins are expressed and at what quantities is determined randomly based on expected protein expression and expected total protein mass fraction.

C.2.11 Ribosome

Biology

Composition

Ribosomes are large ribonucleoproteins which synthesize polypeptides. The *M. genitalium* 70S ribosome is composed of two subunits – the 30S and 50S ribosomal subunits – which assemble on mRNA with assistance from initiation factors 1-3 (MG173, MG142, MG196). The 30S subunit is composed of 1 RNA and 20 protein monomer subunits. The 50S subunit is composed of 2 RNA and 32 protein monomer subunits. The 30S and 50S ribosomal subunits are believed to assemble in stereotyped patterns^{81,283}, and six GTPases – EngA (MG329), EngB (MG335), Era (MG387), Obg (MG384), RbfA (MG143), and RbgA (MG442) – have been associated with ribosomal subunit assembly^{36,41,189,298,304,343,416}. The exact functions of the six GTPases are unknown.

Translation

Following 70S ribosome assembly, ribosomes synthesize amino acid polymers according to nucleic acid templates specified by mRNA and decoded by tRNA. Upon reaching the stop codon UAG, ribosome release factor (MG258) binds to the 70S ribosome, recognizes the stop codon, hydrolyzes the peptidyl-tRNA bond, and releases the terminal tRNA. Finally, elongation factor G (MG089) and initiation factor 3 dissociate the 30S and 50S ribosomal subunits, the mRNA, and the ribosome release factor. See Section C.3.27 for further discussion of translation.

Stalled Ribosome Response

Upon prolonged translational stalling, tmRNA can displace both ribosome-bound tRNA and mRNA, leading to the generation of chimeric polypeptides containing an N-terminal mRNA-coded domain and a C-terminal tmRNA-coded degradation domain, or SsrA tag. Following translation termination, the protein degradation machinery recognizes the SsrA tag and degrades the chimeric polypeptide. See Section C.3.27 and C.3.12 for further discussion.

Computational Representation

The **Ribosome** state represents (1) the status – actively translating or stalled – of each 70S ribosome, (2) the mRNA, or tmRNA in the case of stalled ribosomes, species each 70S ribosome is bound to and translating, and (3) the position, in codons, of each 70S ribosome from the start codon. The status, bound (t)mRNA, and (t)mRNA position of each ribosome are implemented as scalar integer variables.

Integration

The **Polypeptide** state represents the sequence of each nascent polypeptide. The **Protein Complex** state represents the copy numbers of free 30S and 50S ribosomal subunits, and of mRNA-bound 70S ribosomes. The **Rna** state represents the copy number of each mRNA species and of tmRNA.

Three processes – **Translation**, **RNA Decay**, and **Protein Decay** – access and modify the **Ribosome** state. The **Translation** process models the formation of 70S ribosomes on mRNA, polypeptide synthesis catalyzed by 70S ribosomes, and 70S ribosome disassembly following translation termination. The **Translation** process also models ribosome stalling, tmRNA substitution, and SsrA degradation tag synthesis. Computationally, the **Translation** process predicts the state of each 70S ribosome, the mRNA or tmRNA species each 70S ribosome is bound to, and the elongation rate of each nascent polypeptide.

mRNA and tmRNA degradation events modeled by the **RNA Decay** process trigger early 70S ribosome disassembly resulting in incomplete polypeptides. Similarly, 70S ribosome degradation events modeled by the **Protein Decay** process result in incomplete polypeptides. Both processes decrement the copy number of mRNA-bound 70S ribosomes represented by the **Protein Complex** and **Ribosome** states.

The **Ribosome Assembly** process models the assembly of 30S and 50S ribosomal subunits from rRNA and ribosomal protein monomers. The **Protein Activation** process models the effect of antibiotics on the catalytic activity of 30S and 50S ribosomal particles.

Initial Conditions

After the **Rna** and **Protein Monomer** states initialize the total copy number of each RNA and protein monomer species and the **Ribosome Assembly** process initializes the total copy numbers of the 30S

and 50S ribosomal subunits, the **Translation** process initializes the **Ribosome** and **Polypeptide** states. As described in Algorithm C.4, the **Translation** process forms 70S ribosomes equal to the minimum of the 30S and 50S subunit copy numbers, and randomly positions each 70S ribosome on mRNA weighted by the copy number of each mRNA.

Algorithm C.4. Ribosome and Polypeptide state initialization.

```

Free 70S ribosomes ← min(free 30S, 50S ribosomal particles)
Decrement the copy numbers of free 30S and 50S ribosomal particles
foreach 70S ribosome i do
    Select the mRNA of 70S ribosome i weighted by the product of mRNA copy number and length
    Set the bound mRNA species of 70S ribosome i
    Select the position of 70S ribosome i along the bound mRNA with uniform probability
    Set the status of 70S ribosome i to actively translating
    Decrement the copy number of free 70S ribosomes
    Increment the copy number of bound 70S ribosomes
    Set the sequence of the nascent polypeptide corresponding to 70S ribosome i
```

C.2.12 RNA

Biology

Transcription leads to the production of nascent RNA that, once mature, may be used in various cell functions. mRNAs serve as a template for protein synthesis, rRNAs are a part of the ribosome structure, tRNAs carry amino acids to growing polypeptides, and sRNAs act as regulators. tmRNAs come into action when ribosomes stall on an mRNA during translation, helping recycle the stalled ribosome and adding a proteolysis tag to the incomplete polypeptide.

Different nascent RNAs undergo various steps towards maturation, including cleavage from polycistronic (transcription unit) to single RNA form, methylation, and thiolation (see Figure C.4). Maturation is carried out by the **RNA Processing** and **RNA Modification** process classes. There is an additional step for tRNAs which are coupled with amino acids by the **tRNA Aminoacylation** process.

In addition to the various forms an RNA can take in the maturation pathway, an RNA can also transition between various non-functional and functional forms. RNAs may exist with proteins in a macromolecular complex, and when the complex is marked for decay by the **Protein Decay** process class, its RNA subunits are marked as damaged in the **Protein Monomer** state class. Damaged RNAs are degraded by the **RNA Decay** process class. Further, a mature RNA may be freely floating

in the cytoplasm or bound to another molecule in the cell. For example, an mRNA may be bound to a ribosome. This is accounted for in the Bound RNA form. The quantities of RNA in each of these immature/mature and functional/non-functional forms is stored in this state class (see Figure C.4).

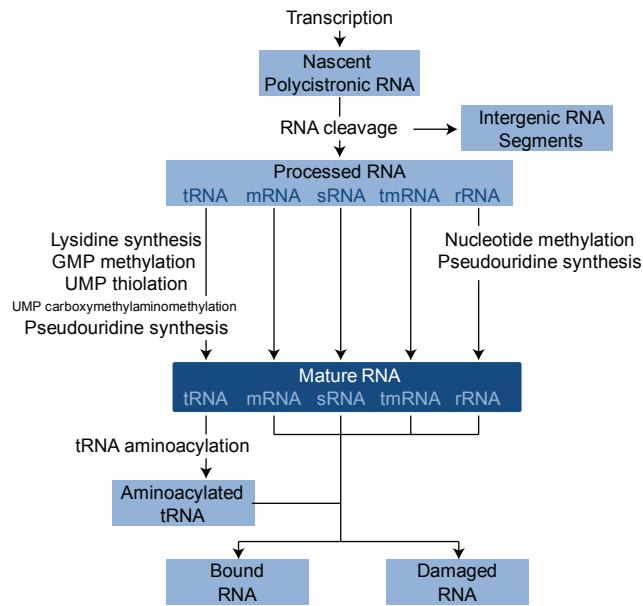


Figure C.4. RNA forms diagrammed in the context of RNA maturation.

Reconstruction

This state class holds information about RNAs such as their weights, compositions, lengths, and localizations. It also holds important parameters for the maturation pathway of RNA, such as a number of matrices that map one form of RNA to another, and the gene composition of transcription units. Other parameters such as expected RNA half-lives and RNA weight fractions are included in this class, and are used to determine the RNA Polymerase-DNA binding parameters associated with gene expression.

Computational Representation

The Protein Monomer state class contains the number of each RNA species in each RNA form.

Other properties are calculated by this state class. The RNA decay rates are calculated from the experimentally measured half lives. The actual decay rates and RNA expression values used in the

model are fit from the experimentally measure values such that the cell will double its mass and successfully divide by the end of the cell cycle.

Integration

Table C.5. Connections between ProteinMonomer state and other processes in the cell.

Connected processes	Pro- cesses	Read from Rna	Written to Rna
Transcription	• Counts of nascent RNAs		• Updated counts of nascent RNAs
RNA Processing	• Counts of nascent, processed, and intergenic RNAs		• Updated counts of nascent, processed, and intergenic RNAs
RNA Modifica- tion	• Counts of processed and mature RNA		• Updated counts of processed and mature RNA
tRNA Aminoac- ylation	• Counts of mature and aminoacylated tRNA		• Updated counts of mature and aminoacylated tRNA
RNA Decay	• Counts of mature, misfolded, and damaged RNA		• Updated counts of mature, misfolded, and damaged RNA
Various Processes	• Counts of mature and bound RNAs		• Updated counts of mature and bound RNAs

Initial Conditions

The system is initialized with a set of RNAs. The determination of which RNAs are expressed and at what quantities is determined randomly based on expected RNA expression and expected total RNA mass fraction.

C.2.13 RNA Polymerase

Biology

RNA polymerases are protein complexes that bind to gene promoters on the chromosomes and mediate the synthesis of RNA transcripts. This state class helps keep track of the conditions and chromosomal locations of RNA polymerases in the cell.

Reconstruction

An RNA polymerase in our model can exist in one of four conditions:

1. Free (not bound to DNA)
2. Non-specifically bound (bound to the DNA, but not to a specific gene promoter and not transcribing)
3. Specifically bound (bound to a specific gene promoter)
4. Actively transcribing (moving along a gene to produce an RNA)

This state class stores properties relating to RNA polymerases such as the expected probabilities of a polymerase being in the above 4 conditions.

Computational Representation

As the simulation progresses, this state class stores the condition of each RNA polymerase in the simulated cell. Transition between the conditions may involve RNA polymerase association and dissociation from the cell's chromosome(s). This state class holds the precise positions on the chromosome(s) where polymerases are bound. The **RNA Polymerase** state class also handles basic accounting. For example, upon an RNA polymerase decay event, a free polymerase is decremented. Further, this class records the premature release of RNA polymerases from the chromosome(s) and passes information about the aborted transcript to the **Transcript** state class.

Further, the **Transcription** process class requires the RNA Polymerase-promoter binding probability for each transcription unit. These probabilities may vary during the cell cycle due to the effects of other processes in the system, such as Transcriptional Regulation or DNA supercoiling. The fold changes to the base binding probabilities incurred by these other processes is stored in the **RNA Polymerase** state class.

This state class can also calculate the total number of RNA polymerases in each condition.

Integration

The **Chromosome** state class updates the chromosomal positions of RNA polymerases in the **RNA Polymerase** state class. The **Transcript** state class reads the aborted RNA sequences from and writes the updated aborted sequences to the **RNA Polymerase** state class.

The **Transcription** process class reads the RNA polymerase conditions and progress from and updates the RNA polymerase conditions and progress to the **RNA Polymerase** state class. The **Transcription** process class also reads the RNA polymerase condition transition probabilities and RNA polymerase binding probabilities from the **RNA Polymerase** state class. The **Transcriptional Regulation** and **DNA Supercoiling** process classes record the fold change in RNA polymerase binding probabilities to the **RNA Polymerase** state class.

Initial Conditions

NA polymerases are initialized as follows:

1. Each RNA polymerase is randomly assigned (with replacement) to one of the actively transcribing, specifically bound, non-specifically bound, or free states weighted by the expected occupancy of each state
2. Actively transcribing and specifically bound polymerases are randomly assigned to transcription units weighted by the transcription unit binding probabilities (P_{tu}).
3. Actively transcribing polymerases are randomly assigned to positions within the assigned segment of their assigned transcription unit (positions near the segment border are not allowed to prevent polymerases from being too close to each other) with uniform probability.
4. Non-specifically bound polymerases are randomly assigned to an accessible region on the chromosome.

C.2.14 Stimulus

Biology

Cells live in complex environments where they are exposed to physical and chemical stresses including antibiotics, radiation, heat and cold. Furthermore, manipulation of the external environment is a powerful tool for discovering new biology. This state represents the status of 10 properties of the external environment.

Reconstruction

Table S3F describes the reconstruction of the values of these 10 properties of the external environment.

Computational Representation

Specifically, this state represents the temperature in °C, six types of radiation in W m^{-2} , Gy s^{-1} , or $\text{m}^{-2} \text{s}^{-1} \text{sr}^{-1}$, and three Boolean-valued stress conditions. Computationally, each property is implemented as a floating point scalar. Table S3F lists the values of these 10 properties used throughout each simulation.

Integration

This state does not directly interact with any of the other 15 states. The **Geometry**, **Metabolite**, **Protein Monomer**, **Protein Complex**, and **Host** states represent additional properties of the extracellular environment. The **Geometry** state represents the volume of the external environment. The **Metabolite** state represents the copy numbers of extracellular metabolites including antibiotics. The **Protein Monomer** and **Protein Complex** states represent the copy numbers of extracellularly localized proteins. The **Host** state represents six properties of the host urogenital epithelium.

Three processes – **Protein Activation**, **DNA Damage**, and **Metabolism** – access the **Stimulus** state. The **Protein Activation** process regulates the activity of four transcription factors as functions of temperature and three stress conditions (see Table S3Q). The **DNA Damage** process models the rates of several types of DNA damage as functions of radiation (see Table S3O). The **Metabolism** process models the generation of hydroxyl radicals from water and γ -radiation (see Table S3O). None of the processes modify the 10 properties of the external environment.

Initial Conditions

For this study the 10 properties of the external environment were initialized to the values listed in Table S3F, and were not modified during the simulation.

C.2.15 Time

Biology

The physical, chemical, and biological processes relevant to cell physiology span a wide range of time scales. To limit the scope of this study, we modeled *M. genitalium* on a 1 s time scale and averaged out the effects of faster processes. *M. genitalium* can be approximated as a well-mixed system at

this time scale.

Computational Representation

This state represents the time elapsed from the start of the simulation in seconds as a single integer variable.

Integration

The **Time** state directly interacts with only three parts of the simulation:

- The **Simulation** linearly advances time in 1 s increments (See Section C.1.1).
- If the cell has not yet divided, each simulation terminates at a predetermined maximum time of 50,000 s, approximately equal to 150% of the observed mean *M. genitalium* mass doubling time (See Section C.1.1).
- The concentrations of extracellular metabolites and the values of extracellular stimuli depend on time (see Section C.2.7 and C.2.14).

None of the processes or other states depend directly on the **Time** state.

Initial Conditions

The **Time** state is initialized to zero and the other 15 states are initialized to the beginning of the cell cycle. Consequently, the **Time** state also represents the time elapsed from the beginning of the cell cycle.

C.2.16 Transcript

Biology

During **Transcription**, an RNA polymerase binds to a gene promoter and facilitates the synthesis of an RNA transcript. This state stores all of the information that pertains to nascent RNA transcripts and aids in the time evolution of **Transcription**.

Reconstruction

There are multiple fixed properties relating to the gene templates of transcripts that are essential for transcription and stored in the **Transcript** state. For each transcription unit, we store the 5' coordinate of the gene on the genome. This is the template for the 1st nucleotide in a growing transcript. In addition to this, we store the direction in which the template is read, towards or away from the origin. We also store the sequence and length of each transcription unit.

Computational Representation

Transcription proceeds at a maximal rate of 50 nucleotides per second^{94,421}. Therefore, it may take several timesteps to synthesize a transcript, and the progress of the RNA polymerase along the transcription unit must be stored. This state stores the transcription unit to which each RNA polymerase is bound, including which chromosome it is bound to and the progress of the RNA polymerase along the transcription unit.

An RNA polymerase may stall in the **Transcription** process or be knocked off of a gene by another protein. In these cases, an incomplete transcript may result, and will be targeted for degradation by the **RNA Decay** process. This state holds the sequence of each aborted transcript, such that the nucleotides can correctly be accounted for upon degradation.

Multiple RNA polymerases may be bound to a given transcription unit, and this state can calculate the total number of polymerases on each unit. Finally, the **Transcript** state also calculates the dry weight of the nascent transcripts.

Integration

The **Transcript** state class reads about the existence of an aborted sequence from the **RNA Polymerase** state class. Upon RNA polymerase displacement, the **Transcript** state class updates the **Transcript** state in the **Chromosome** state class. It also records the start sites and directions of transcription units in the **Chromosome** state class.

The **Transcription** process class uses all of the fixed properties and time evolving properties housed in the **Transcript** state class.

Initial Conditions

An RNA polymerase may be initialized in the actively transcribing state. For all such polymerases, the growing transcript is accounted for in the `Transcript` state class.

C.3 Cellular Process Methods

The whole-cell model is composed of 28 sub-models which span six major areas of cellular physiology: (1) transport and metabolism, (2) DNA replication and maintenance, (3) RNA synthesis and maturation, (4) protein synthesis and maturation, (5) cytokinesis, and (6) host interaction. The sub-models were modeled using different mathematics and trained using different experimental data. Each of the 28 functional processes, or cellular processes, represents a group of chemical reactions which transform chemical substrates into products using enzyme catalysts. Computationally, the inputs and outputs of each sub-model are the copy numbers of metabolites and macromolecules; the configurations of RNA, protein and DNA polymers; and the catalytic capacity and configurations of the enzymes which participate in each sub-model. This chapter provides detailed discussions of the mathematics and computational implementation of each cellular process.

Metabolism

The `Metabolism` process modeled the import of nutrients from the external environment and their conversion primarily into the metabolic building blocks required by the 27 other processes for macromolecule synthesis. Therefore, the `Metabolism` process served as the primary interface between the external environment and the 27 other processes, providing the nutrients required for each of the other processes and recycling and/or exporting the metabolic byproducts of the other processes. The `Metabolism` process was reconstructed primarily based on DNA sequence homology of *M. genitalium* to *E. coli* and a comprehensive metabolic model of *E. coli*¹²⁰. The `Metabolism` process was modeled using FBA and trained using the observed growth rate of *M. genitalium* and the observed chemical compositions of *M. gallisepticum*²⁶¹ and *E. coli*²⁷⁸. The composition of the *in silico* growth medium was reconstructed based on the reported chemical composition of the individual components of *Mycoplasma* SP-4 medium, with additional metabolites added to support sustained *in silico* growth.

RNA Synthesis & Degradation

RNA synthesis and maturation was modeled by four processes: **Transcription**, **RNA Processing**, **RNA Modification**, and **tRNA Aminoacylation**. **Transcription** modeled the state – free, non-promoter bound, promoter-bound, or actively transcribing – of each RNA polymerase, transcription initiation at specific promoters, transcription elongation and NTP allocation among nascent transcripts, and transcription termination. The state of each RNA polymerase was modeled as a Markov chain and trained with the observed occupancies of each of the four modeled states²⁴². Transcription initiation was modeled as a stochastic process and trained using the reconstructed expression and decay rates of each transcription unit^{29,431}. The transcription unit organization of the chromosome was reconstructed from the observed operon organization of *M. pneumoniae*¹⁴⁰. Because transcription termination is not well-characterized, termination was modeled as a deterministic process which proceeds to completion within the 1s simulation time step if there is least one copy of each of the characterized transcription termination factors.

Transcriptional Regulation modeled the fold change effects of transcriptional regulators on the affinity of RNA polymerase for specific promoters. The **Transcriptional Regulation** sub-model was reconstructed using the database DBTBS³⁶². **Transcriptional Regulation** was modeled as a stochastic process and trained using reported fold change effects.

Following transcription, non-coding transcripts are cleaved, modified, and aminoacylated. The **RNA Processing** sub-model modeled the cleavage of polycistronic non-coding RNA into individual non-coding RNA gene products. The **RNA Processing** sub-model was reconstructed primarily based on reported *E. coli* RNA cleavages^{271,281,298} and the complement of RNA processing enzymes contained in the *M. genitalium* genome. The **RNA Modification** process modeled the modification of specific bases of specific non-coding RNAs and was reconstructed based on the observed complement of *E. coli* RNA modifications^{4,31,32,49,111,210,224,298,341}. The **tRNA Aminoacylation** sub-model modeled the aminoacylation of tRNAs according to the observed *Mycoplasma* genetic code^{87,364,429}. **RNA Processing**, **RNA Modification**, and **tRNA Aminoacylation** were modeled motivated by mass-action kinetics and parameterized using the observed kinetics of the RNA cleavage and modification enzymes.

RNA transcripts are hydrolytically cleaved into 10-30 nucleotide fragments by ribonuclease R, which in turn are further degraded into individual NMPs. The **RNA Decay** sub-model modeled the decay of

each RNA species as a first-order Poisson process with rate parameters equal to the observed decay rates of *E. coli* RNA²⁹.

Protein Synthesis & Degradation

Protein synthesis and maturation was modeled by nine processes: **Translation**, **Protein Processing I**, **Protein Translocation**, **Protein Processing II**, **Protein Folding**, **Protein Modification**, **Macromolecular Complexation**, **Ribosome Assembly**, and **Terminal Organelle Assembly**. **Translation** modeled the assembly of 70S ribosomes on mRNA Shine-Dalgarno sequences, the polymerization of amino acids into polypeptides, and the allocation of aminoacylated tRNAs among the multiple active ribosomes. The **Translation** sub-model also modeled the role of tmRNA in the termination of stalled ribosomes as a rare stochastic event.

Following translation, polypeptides are deformylated, cleaved, translocated, folded, and modified before forming macromolecular complexes. **Protein Processing I** modeled the first steps in protein maturation: polypeptide deformylation and N-terminal methionine cleavage. **Protein Translocation** modeled integral membrane, lipoprotein, and secreted protein translocation into the membrane by the SecA translocase. **Protein Processing II** modeled lipoprotein diacylglycerol transfer, and lipoprotein and secreted protein signal sequence cleavage. **Protein Folding** modeled the folding of polypeptides into compact three-dimensional structures. **Protein Modification** modeled the covalent modification of specific amino acids. **Macromolecular Complexation** modeled the formation of protein and ribonucleoprotein complexes. **Ribosome Assembly** modeled the role of several GT-Pases in the assembly of 30S and 50S ribosomal particles. **Terminal Organelle Assembly** modeled the observed hierarchical assembly of the protein content of the *M. genitalium* terminal attachment organelle.

The eight post-translational processing sub-models were reconstructed based on specific N-terminal methionine cleavages observed in *Shewanella oneidensis* MR-1¹⁴⁵, the predicted localization and signal sequence of each protein monomer (see Table S3AM-S3AO), the observed chaperone interactions of *E. coli*^{95,187} and *B. subtilis*¹¹³, the observed complement of *M. genitalium* and *M. pneumoniae* protein modifications^{93,163,200,231,386}, and the inferred subunit composition of each macromolecular complex (see Table S3AS). The eight post-translational processing sub-models were modeled motivated by mass-action kinetics and were parameterized using the reported kinetics of the post-translational processing enzymes.

The **Protein Decay** sub-model modeled the hydrolytic cleavage of proteins into 10-20 amino-acid-long fragments by protease La, as well as the cleavage of aborted polypeptides by protease FtsH. Similar to **RNA Decay**, **Protein Decay** was modeled as a first-order Poisson process. The half-life of each protein was predicted using the N-end rule⁴⁰⁴. Additionally, the **Protein Decay** sub-model modeled protein misfolding and ClpB chaperone-mediated refolding. Protein misfolding was modeled as a stochastic process. Protein refolding was modeled as a deterministic process governed by a single Boolean rule.

The **Protein Activation** process modeled the chemical regulation of protein activity. The **Protein Activation** process was reconstructed primarily based on the database DrugBank¹⁹². Because the exact mechanisms of protein chemical regulation are poorly characterized, **Protein Activation** was modeled as a Boolean network.

Chromosomal Replication & Maintenance

Seven processes modeled chromosomal replication, damage, and maintenance: **Replication Initiation**, **Replication**, **Chromosome Segregation**, **Chromosome Condensation**, **DNA Supercoiling**, **DNA Damage**, and **DNA Repair**. The **Replication Initiation** process modeled the recruitment of DNA polymerase to the oriC by the formation of a large DnaA complex at the R1-5 functional oriC DnaA boxes. **Replication Initiation** was reconstructed using the reported DnaA DNA-binding motif, and implemented similar to the *E. coli* replication initiation model developed by Messer²⁴⁹.

The **Replication** process modeled bidirectional DNA replication, dissolution of the oriC DnaA complex, single stranded binding protein (SSB) binding to exposed single-stranded DNA, and Okazaki fragment ligation. Allocation of dNTPs among elongating DNA polymerases was modeled similarly to **Transcription and Translation**.

The **Chromosome Segregation** process modeled daughter chromosome segregation following chromosomal replication. Because **Chromosome Segregation** is poorly characterized, **Chromosome Segregation** was modeled as a single event governed by a simple Boolean rule.

Chromosome Condensation and **DNA Supercoiling** modeled DNA compaction mediated by structural maintenance of chromosome (SMC) proteins and topoisomerases. **Chromosome Condensation** was reconstructed based on the reported DNA density of SMC proteins¹⁶⁸ and SMC DNA-binding was modeled as a stochastic process. **DNA Supercoiling** was modeled as the balance of the competing winding and unwinding effects of topoisomerase I and DNA gyrase, and parameterized by

the observed kinetics of topoisomerase I and DNA gyrase^{46,91,384}.

DNA Damage modeled spontaneous and chemically- and radiation-induced DNA damage as a stochastic process parameterized by the reported efficiencies of DNA damage. **DNA Repair** modeled four modes of DNA repair – direct damage repair (DDR), base excision repair (BER), nucleotide excision repair (NER), and homologous recombination double strand break repair (HR-DSBR) – as well as the *M. genitalium* MunI restriction/methylation (R/M) system. DDR, BER, NER, and HR-DSBR were modeled as stochastic processes parameterized by their observed kinetic rates. R/M was reconstructed based on the reported MunI DNA-binding motif and modeled as a stochastic process.

Cytokinesis

Following chromosomal replication and segregation, *M. genitalium* divides by binary fission. According to the Li et al. hypothesis²¹⁶, *M. genitalium* divides by iteratively pinching its the septal membrane using the filamentous protein FtsZ. The **FtsZ Polymerization** process modeled the formation of septal FtsZ rings. **FtsZ Polymerization** is implemented using the ordinary differential equations described by Surovtsev et al.³⁸⁸. The **Cytokinesis** process modeled the iterative contraction of successively smaller FtsZ septal rings. The **Cytokinesis** sub-model was parameterized using the observed rate of FtsZ-GTP hydrolysis¹⁵⁶.

Cell Cycle

Three *M. genitalium* cell cycle phases were modeled: (1) pre-replication, or replication initiation, (2) replication, and (3) cytokinesis. All simulations reported in this study were initialized to the beginning of the replication initiation phase. Starting at the beginning on each simulation, the **Replication Initiation** process modeled the formation of the oriC DnaA complex, ultimately resulting in the recruitment of DNA polymerase to the replication origin and the start of DNA replication. Following DNA polymerase recruitment to the replication origin, the **Replication** process modeled DNA replication. After chromosomal replication the **Chromosome Segregation** process modeled daughter chromosomal segregation. Finally, the **FtsZ Polymerization** and **Cytokinesis** processes modeled the formation and contraction of FtsZ septal rings, ultimately resulting in cell division.

Host Interaction

The Host Interaction process modeled the interaction of *M. genitalium* with its host human urogenital epithelium. The Host Interaction process was reconstructed based on the observed composition of the protein content of the *M. genitalium* terminal organelle^{197,198} and the reported *M. genitalium*-host interactions^{101,244,324,325,357,358}. Because the interaction of *M. genitalium* with its human host is poorly characterized, Host Interaction was implemented as a Boolean model.

C.3.1 Chromosome Condensation

Biology

Bacterial chromosomes are compacted 10^4 -fold in volume in order to physically fit inside of a cell⁴²². Compaction is also necessary to support various cellular processes including chromosome replication, cell division, and chromosome segregation to opposing poles of the cell⁴²². Bacteria employ several mechanisms to compact DNA including "clamping" of the DNA by structural maintenance of chromosome (SMC) proteins, DNA supercoiling, macromolecular crowding, and charge neutralization⁴²². *M. genitalium* has the machinery to perform all of these levels of DNA compaction, and we explicitly model DNA Supercoiling in a separate process class and DNA clamping by SMC complexes here in the Chromosome Condensation process class.

Reconstruction

SMC complexes consist of an SMC core protein (Smc: MG298) and segregation and condensation proteins A and B (ScpA: MG213, ScpB: MG214). Together, the SMC complex is a "V" shaped structure (with a head and two legs) that induces positive supercoils in double stranded DNA³¹⁵. The complexes are believed to work with a lock and key mechanism in which first DNA is looped around the legs of the SMC complex, and then an ATP is bound between the two tails to lock the SMC complex in place. The complexes bind and clamp the DNA causing many loops in the DNA and DNA compaction. The loops around each leg occupy 90bp. A loop of about 450bp forms between the two SMC complex legs^{168,385}. Further, it has been inferred that there is about one SMC complex bound for every 7130bp of DNA (See Figure C.5)¹⁶⁸. All of the parameters used in the Chromosome Condensation process class are described in Table C.6.

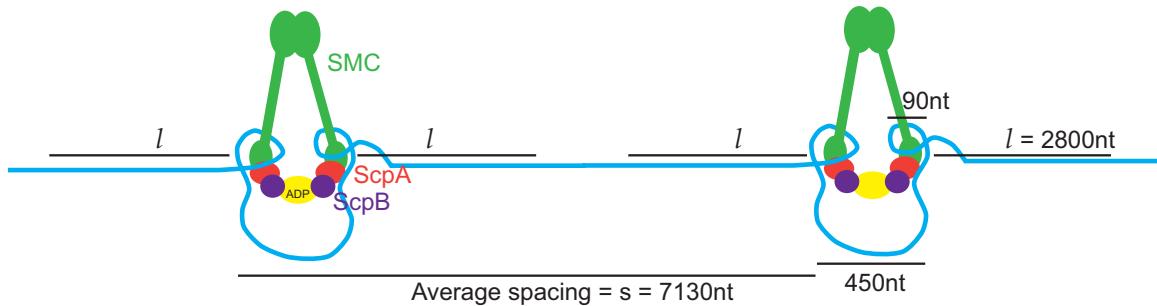


Figure C.5. SMC complex occupation of the DNA.

Table C.6. Fixed parameters used in the Chromosome Condensation process class.

Parameters	Value	Symbol	Source
Average separation of SMC complexes on the chromosome(s)	7130 nt	<i>s</i>	168.
SMC complex threshold spacing	2800 nt	<i>l</i>	Fit
SMC DNA footprint	630 nt		168.
SMC arm DNA footprint	90 nt		168.
SMC inter-arm loop length	450 nt		168.

Parameter Assignment

Given that the DNA footprint of an SMC complex is much less than its average spacing, if we simply applied a $1/s$ probability of each base being bound at each timestep, over time given an SMC abundance we would see SMCs bound at intervals much less than s . The SMC complex threshold spacing parameter (l) prevents this phenomenon by inhibiting SMC complex binding where the gap between existing SMC complexes on the DNA is already much less than the desired spacing. Biologically, this parameter represents the notion that SMC complex binding is not only determined by SMC complex abundance, but also the writhe and physical properties of the DNA that prevent SMC complex to bind too close to each other. This parameter was fit such that the average SMC complex spacing in the model is 7130 nt. The fit of this parameter was assessed by various tests to assure an average SMC spacing of 7130 nt.

Computational Representation

This module models the contribution of SMC proteins to chromosome condensation. SMC-related DNA condensing could be considered along with the effects of topoisomerases, but we model the effect of the topoisomerases by tracking the effect of each topoisomerase activity on the DNA linking number (in the DNA supercoiling process). Since SMCs do not act by strand-passing events (the causing of a nick in the DNA, enabling two double stranded DNA regions to pass through each other, and the re-ligation of the DNA), and since we do not know the exact effect of SMC complex activity on the DNA linking number, we consider SMC condensation as compacting the DNA at a different level and model it independently of the DNA linking number calculations in the DNA Supercoiling process. Macromolecular crowding and charge neutralization are not presently modeled.

SMCs are bound at an average spacing (s) of 7130 bases¹⁶⁸. Since it is unknown whether SMC complexes bind to specific DNA motifs, at each timestep, we bind SMC complexes to random positions on the chromosomes. The binding is weighed by calculated probabilities (P) of a free SMC complex binding to each accessible double-stranded DNA base. The probability distribution is calculated as a step function: bases within a threshold distance (l) from another SMC complex have a zero probability of being bound, and bases beyond a threshold distance, l , from other SMC complexes bind at a probability of $1/s$. SMC complex disassociation from the DNA occurs upon interaction with other DNA binding proteins and is handled by the `Chromosome` state class.

Integration

The `Chromosome Condensation` process class reads from and writes to the `Chromosome` state class. It reads in the regions of DNA that SMC complexes can bind to, and writes back the positions of SMC complexes on the chromosome(s).

Initial Conditions

Before the start of the simulation, we iteratively run the SMC complex binding calculations described above, to bind SMC complexes to the first chromosome until a steady state is reached, that is no more SMC complexes can be bound.

Dynamic Computation

At each timestep, we perform the following algorithm:

1. Calculate the maximum number of SMC complexes that can bind in the given timestep.

$$\text{SMCBindingLimit} = \min(\text{Number of free SMC complexes}, \text{Number of availableATPmolecules})$$

2. Bind SMCs
 - (a) Query the Chromosome state to determine the free bases where SMC complexes can bind.
Accessible bases are $> l$ bases away from any DNA bound SMC complexes.
 - (b) Given that the probability of binding each accessible base is,

$$P_{\text{Binding Accessible Base}} = \frac{1}{s - 2l}$$

randomly choose sites to bind based on this probability up to the SMCBindingLimit

- (c) Form an SMC-ATP complex, and hydrolyze the ATP to ADP.
 - i. Decrement a free SMC
 - ii. Increment and decrement free metabolites for ATP hydrolysis
- (d) Bind the SMC-ADP complex to chromosome (facilitated by the Chromosome state)

C.3.2 Chromosome Segregation

Biology

DNA replication produces two chromosomes that are catenated, or connected like links of a chain. Before cell division can take place, these chromosomes need to migrate to opposing sides of the cell and be decatenated. Chromosome segregation in *M. genitalium* is not well understood, but is believed to result from both entropic factors and enzymatic activity^{35,176}.

Reconstruction

The Chromosome Segregation process class assumes that the chromosomes entropically segregate during replication as described in Bloom et al. and Jun et al.^{35,176}. That is, it is unfavorable for the two copies of replicated DNA to exist too close to each other, thus as the replication fork

moves, the replicated DNA starts to migrate towards the poles. Enzymatic segregation activity is also modeled. The five chromosome segregation proteins are a nucleotide binding domain (CobQ/CobB/MinD/ParA: MG470), an MraZ protein (MraZ: MG221), a GTP binding protein (Era: MG387), a GTPase (Obg: MG384), and topoisomerase IV (ParE: MG203, ParC: MG204). The other segregation protein, FtsZ is accounted for in the **Cytokinesis** process class. These proteins assist in the chromosomal migration towards the cell poles and decatenate the chromosomes following the completion of replication. However, there is no detailed understanding of the function of these proteins, and much fewer proteins are present in *M. genitalium* than that required for the detailed mechanisms for segregation described for other bacterial species. Further, the specific kinetics, and metabolic costs of these proteins are not known.

Note that topoisomerase IV is also used in the **DNA Supercoiling** process class, as it is known to relieve coils that form just downstream of the replication loops. Here it performs a decatenation function, the unlinking of the two chromosomes.

All of the parameters used in the **Chromosome Segregation** process class are described in Table C.7.

Table C.7. Fixed parameters used in the Chromosome Segregation process class.

Parameter	Value	Symbol	Source
GTP cost of chromosome segregation	1 GTP	E_{seg}	Value unknown
Superhelical density tolerance	± 0.1	σ_{tol}	Value unknown
Equilibrium superhelical density	-0.06	σ	46.

Parameter Assignment

The energetic cost of segregation has not been experimentally characterized. The energetic cost is set to a nominal value of 1 GTP per segregation event. The superhelical density tolerance is also unknown, so the allowed range of DNA superhelical density was set to be quite wide.

Computational Representation

This process models enzymatically catalyzed sister chromosome separation and decatenation. Entropically driven sister chromosome separation is not well understood and is not presently modeled.

Because the molecular biology of chromosome segregation is not well understood, we have chosen to

implement this process as a simple Boolean process: the chromosomes are regarded as segregated immediately after the following four conditions are met:

- The chromosome is replicated
- The chromosomes are properly supercoiled within a given tolerance of superhelical density
- There is at least one free and functional molecule of each of the five segregation proteins, and
- There are enough available GTP molecules

Note that Glass et al. gene essentiality study suggests that the *cobQ/cobB/minD/parA* gene is non-essential, but we model this gene as essential because we don't know its specific function and how the other segregation proteins compensate in its absence.

Integration

Table C.8. State classes connected to the Chromosome Segregation process class.

Connected States	Read from state	Written to state
Chromosome	<ul style="list-style-type: none"> • Is chromosome replicated? • Are chromosome superhelical densities within tolerance? • Are chromosomes segregated? 	<ul style="list-style-type: none"> • Chromosomes are segregated

Initial Conditions

The chromosome state is initialized with 1 chromosome with superhelical density within the acceptable tolerance.

Dynamic Computation

At each timestep, the following conditional statement is performed: If

- The chromosome is replicated
- The superhelical density is in the range: $\sigma \pm \sigma_{tol}$
- There is at least one free and functional molecule of each of the 5 segregation proteins
- Available GTP $\geq E_{seg}$

Then, mark the chromosome as segregated.

C.3.3 Cytokinesis

Biology

Cytokinesis, the division of the cell into two separate daughter cells, is the final step in the cell cycle. The major step of cytokinesis is the pinching of the cell membrane in a certain location (typically the midline of the cell) until the membrane is separated and division is complete. Typically, a membrane-bound ring of a tubulin homologue forms at the cell membrane around the pinching site, and the localization and assembly of this ring requires a set of accessory proteins⁷. Both contractile forces of the ring filaments, and external forces by the accessory proteins contribute to the pinching of the cell membrane⁷.

Reconstruction

M. genitalium has the tubulin homologue, FtsZ (MG224), but does not contain the accessory proteins required for cell division in other bacterial species. Therefore, cytokinesis in *M. genitalium* is not fully understood. However, Li et al. propose an “iterative pinching” model by which a cell can divide using only FtsZ²¹⁶. Their proposed mechanism invokes the formation of a contractile Z ring along the interior surface of the midline of the cell membrane. The ring is comprised of GTP-activated FtsZ polymer filaments that bend when their GTP is hydrolyzed. The bending draws the membrane-bound ends of the filaments together, thereby constricting the cell membrane. Many such pinching events result in a complete constriction of the cell. *M. genitalium* cytokinesis is thus a cycle of filament binding, bending, and dissociation.

Lluch-Senar et al. suggest that *M. genitalium* cell division can occur in the absence of FtsZ through motility²²¹. We chose to use the Li et al. model because we do not have a descriptive model of *M. genitalium* motility and FtsZ is known to be an essential gene in *M. genitalium*.

All of the parameters used in the Cytokinesis process class are described in Table C.9.

Table C.9. Fixed parameters used in the Cytokinesis process class.

Parameters	Value	Symbol	Source
Length of an FtsZ filament	40 nm	L	7.
Number of FtsZ subunits in an FtsZ filament	9 subunits	S	7.
Rate of FtsZ-GTP hydrolyzing to FtsZ-GDP	0.15 s^{-1}	k_{hyd}	156.
Rate of FtsZ polymers binding to the membrane	0.7 s^{-1}	k_{bind}	Fit
Rate of FtsZ polymers dissociating from the membrane	0.7 s^{-1}	k_{unbind}	Fit

Parameter Assignment

The kinetic rates of FtsZ filaments binding and dissociating from the membrane, k_{bind} and k_{unbind} , were uncharacterized. Therefore, the rates were set to be rapid such that their values under typical protein expression do not significantly limit the progression of cytokinesis.

Computational Representation

We represent cytokinesis using a modified version of the Li et al. model²¹⁶. There are four main steps to our iterative pinching model, as seen in Figure C.6. First straight FtsZ-GTP filaments bind inside the cell membrane, such that they form an inscribed polygon (Figure C.6A). Next the GTP molecules in these filaments hydrolyze to GDP, causing the filaments to bend. Since the filaments are cell membrane bound, this bending pinches the membrane inwards. The filaments bend just enough to approximately form a new circle. The new circumference of the pinched region is now smaller, and equal to the perimeter of the inscribed polygon (Figure C.6C). The precise timing of the binding, hydrolysis, and dissociation of the contractile rings allows us to maintain the contraction progress across iterative pinching steps. After hydrolysis, one of the two bent filament rings is dissociated. The other remains, to maintain the progress of cell pinching (Figure C.6D). Now a new set of straight filaments can bind and form a new polygon that will be smaller than the previous (Figure C.6A). Upon binding, the residual ring of bent filaments can dissociate (Figure C.6B). The cycle repeats, resulting in smaller and smaller pinched circumferences, until the cell has divided.

We adapted the Li et al. model into our model framework as follows. First, we implemented cytokinesis as a process that is only initiated upon chromosomal segregation. Next, the Li et al. model only explicitly involves a single set of filaments binding at each edge during the binding stage.

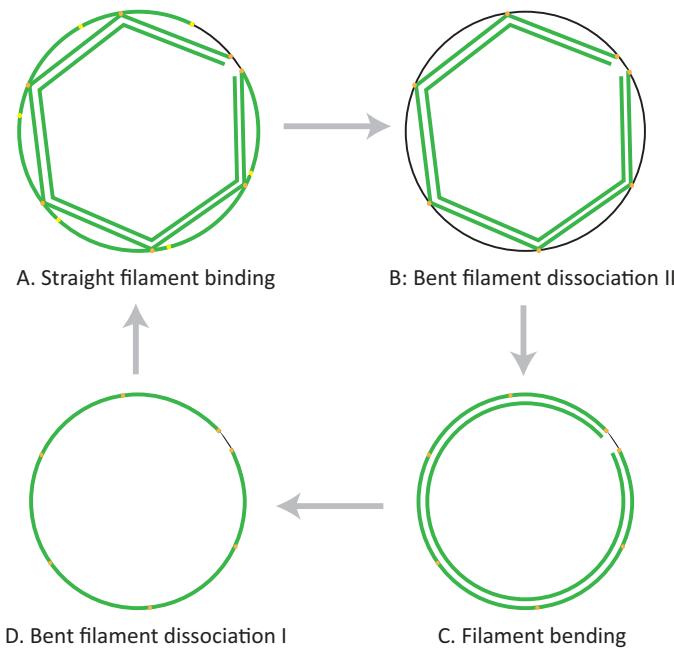


Figure C.6. Algorithm to bind, bend, and dissociate FtsZ filaments to pinch cell.

We have modified this such that two sets of filaments bind (Figure C.6A). Multiple filament sets are required to maintain the progress of membrane constriction. Indeed, Li et al. propose that the FtsZ filaments are more abundant and that a new ring of filaments attaches to stabilize the membrane and maintain progress before the depolymerization event. Osawa et al. also report that the FtsZ ring is 3-9 filaments thick depending on the bacterial strain²⁸⁸. Using the lower bound, since *M. genitalium* is a very small bacterium, we assume that the FtsZ maintains a thickness of up to 3 filaments. Our modified version of the Li et al. model requires two sets of straight FtsZ filaments to inscribe the diameter of the cell. Then up to one set of filaments can depolymerize while two new sets of straight filaments bind the diameter of the cell. Only once the two new sets of straight filaments have bound, can the second bent set of filaments depolymerize.

The output of this process is an indication of the progress of cell pinching, including the determination of when the cell has successfully split into two daughter cells. Complete cell division is also the trigger to end the entire simulation.

Integration

Table C.10. State classes connected to the Cytokinesis process class.

Connected State	Read from state	Written to state
FtsZ Ring	<ul style="list-style-type: none"> Number of filaments bound to each edge, and whether they are bent or straight Calculation of number of polygon edges in current cell diameter 	<ul style="list-style-type: none"> Updated number of filaments bound to each edge, and whether they are bent or straight
Geometry	<ul style="list-style-type: none"> Pinched diameter of the cell Filament length parameters (L, S) 	<ul style="list-style-type: none"> Updated pinched diameter of the cell
Chromosome	<ul style="list-style-type: none"> Has chromosome segregated? 	

Initial Conditions

Initialization steps are not required for this process, as Cytokinesis does not take place until late in the cell cycle. Initialization of FtsZ polymers is handled by the FtsZ Polymerization process class.

Dynamic Computation

At each timestep we perform our modified version of the Li. et al. algorithm²¹⁶. This involves binding a pair of full length (9mer) FtsZ polymers along the circumference of the midplane of the cell. When these filaments hydrolyze and bend we use circle and arc formulas to calculate the new smaller circumference of the cell. Depending on the state of the FtsZ ring, one or more of the following five steps is performed.

1. Straight filament binding (Figure C.6A): If cytokinesis has just initiated (triggered by Chromosome Segregation) or if the previous FtsZ ring has hydrolyzed and undergone dissociation I, attempt to bind two straight filaments at each polygon edge, forming a regular polygon that inscribes the pinched cell circumference:
 - (a) Determine the number of filaments to bind, based on the number of edges on the polygon

that inscribes the pinched cell circumference. The number of edges (`numEdges`) is determined using the pinched diameter of the cell calculated in the previous timestep (D_{prev}), and the length of the FtsZ filament (L):

$$\text{numEdges} = \left\lfloor \left(\frac{\pi}{\sin^{-1} \left(\frac{L}{D_{\text{prev}}} \right)} \right) \right\rfloor$$

- (b) Bind straight filaments such that each edge has a random chance of being bound and the probability of binding each of two filaments is equal to k_{bind}
 - (c) Upon each binding event, update the state of the FtsZ ring
2. Bent filament dissociation II (Figure C.6B): If there was a previous cycle and at least one straight filament has bound each polygon edge, then dissociate the bent filament from each edge.
- (a) Randomly dissociate bent FtsZ-GDP filaments such that the probability of dissociating each filament is equal to k_{unbind} .
 - (b) Dissociate unbound FtsZ-GDP filaments into FtsZ-GDP monomers.
 - (c) Upon each dissociation event, update the state of the FtsZ ring.
3. Filament bending (Figure C.6C): If all polygon edges have two filaments bound and all residual bent filaments have dissociated, bend the edges of the newly completed polygon. When the filaments bend, their length does not change, and the amount of bending is only sufficient to form a new circle. The new circumference is therefore equal to the old polygon's perimeter. Each fragment is now an arc. For simplicity, all of the GTP molecules in the pair of filaments at a particular edge hydrolyze at the same time. In this version of the model, all filaments are of a fixed length. When they are joined end-to-end to form a regular polygon, the polygon may not fully inscribe the entire circumference. This remaining portion of the circumference does not bend when the polygon does. It is preserved and accounted for in the next iteration.
- (a) Randomly determine whether to bend the filaments, such that the probability of bending each edge is k_{hyd} . Bending at an edge can only take place if there are sufficient water molecules available to hydrolyze all of the GTP molecules in the two filaments.
 - (b) Update the state of the the FtsZ ring and change the bent filament from FtsZ-GTP to FtsZ-GDP.

- (c) If all the edges have been bent, calculate the new pinched diameter of the cell:

$$\begin{aligned}\text{Diameter}_{\text{AfterHydrolysis}} &= \frac{\text{circumference}}{\pi} \\ &= \frac{\text{numEdges} \times L}{\pi}\end{aligned}$$

Adjust this diameter to account for the polygon not perfectly inscribing the cell circumference by adding back the length of the old perimeter not accounted for by the inscribed polygon.

4. Bent filament dissociation I (Figure C.6D): If all the filaments have been bent, dissociate one bent filament from each polygon edge. The other ring must remain to preserve pinching progress by maintaining the new smaller circumference.
 - (a) Randomly dissociate bent FtsZ-GDP filaments such that the probability of dissociating each filament is equal to k_{unbind} .
 - (b) Dissociate unbound FtsZ-GDP filaments into FtsZ-GDP monomers.
 - (c) Upon each dissociation event, update the state of the FtsZ ring.
5. Termination: If the pinched diameter is smaller than the length of one filament, conclude cytokinesis.

C.3.4 DNA Damage

Biology

DNA is not absolutely stable. Rather, DNA is susceptible to spontaneous modification including base loss and deamination. DNA is also susceptible to modification by many exotic agents including UV-A and UV-B radiation, α -, β -, and γ -radiation, oxygen, hydroxyl, carbon, and nitrogen radicals, and alkylating agents. These diverse DNA modification modes generate equally diverse DNA structures including missing and modified sugar-phosphates, missing and modified nucleobases, intra- and inter-strand cross links, and strand breaks. Table S3O lists the modeled causes and types of DNA damage.

DNA damage is an evolutionarily important source of genetic variation. However, damaged DNA has also been shown to deleterious to many processes including transcription and replication^{68,409–411}. Consequently, cells employ dedicated DNA repair machinery. See Section C.3.5 for further discussion. The exact effect of DNA damage on physiology is not well characterized. Tornaletti and Hanawalt

have extensively reviewed the effects of several specific DNA modifications on transcription^{409–411}.

Reconstruction

DNA damage modes were reconstructed based on a review by Lindahl and Barnes²¹⁸ and the primary literature^{27,30,48,50,69,97,108,123,129,141,142,149,150,186,202,211,253,259,350,399,437}. Table S3O lists the reconstructed causative agent, resultant base configuration, metabolite stoichiometry, and kinetic rate of each DNA damage reaction.

Computational Representation

Mathematical Model

Because DNA damage is rare, this process models each DNA damage reaction as an independent Poisson process. The rate, r_i , of each spontaneous DNA damage reaction i is given by the observed specific rate k_i . The rate, r_i , of each DNA damage reaction i caused by radiation j is given by the product of the experimentally observed specific rate, k_i , and the radiation flux, s_j ,

$$r_i = k_i s_j. \quad (\text{C.11})$$

Chemically-induced damage reactions were reconstructed, but not modeled because *M. genitalium* is not typically cultured with DNA damaging agents.

Integration

The **Chromosome** state represents the copy number, modification status, and protein occupancy of each chromosome. The **Metabolite** state represents the copy number of each metabolite. The **Stimulus** state represents the fluxes of six types of radiation: α -, β -, and γ -particles, electrons, protons, and UV-A and UV-B.

The **Metabolism** process models the generation of hydroxyl radicals. The **DNA Repair** process models four DNA repair pathways: direct damage reversal, base excision repair, nucleotide excision repair, and homologous recombination. The **DNA Repair** process also models DNA methylation at restriction/modification (R/M) sites. All DNA modifications except methylations of R/M sites are assumed to impede protein DNA-binding.

Initial Conditions

The Chromosome state initializes one chromosome, fully methylated at R/M sites and otherwise unmodified. Several other processes including Transcription initialize the protein occupancy of the chromosome.

Dynamic Computation

Algorithm C.5 outlines the implementation of the DNA damage model.

Algorithm C.5. DNA damage simulation. See the Mathematical Model section above and Table C.2 for definition of the mathematical notation.

```

Input:  $m_i$  copy number of metabolite  $i$ 
Input:  $M_{ij}$  stoichiometry of metabolite  $i$  in reaction  $j$ 
Input:  $z_i^g$ ,  $z_i^a$ ,  $Z_{\bullet i}^p$ ,  $Z_{\bullet i}^b$ ,  $z_i^c$ , and  $z_i^s$ : final base configuration resulting from reaction  $i$ 

foreach DNA modification reaction  $i$  do
    Calculate base modification rate
    switch trigger of reaction  $i$  do
        case spontaneous
             $r_i \leftarrow k_i$ 
        case radiation
             $j \leftarrow$  index of radiation trigger
             $r_i \leftarrow k_i s_j$ 

        foreach base  $j$  in strand  $k$  of chromosome  $l$  susceptible to reaction  $i$  in a random order do
            if insufficient metabolic resources to support reaction  $i$  ( $\exists j$  s.t.  $m_j < -M_{ji}$ ) then
                break
            if poissonRand( $r_i$ )  $\geq 1$  then
                Update the configuration of base  $j$  of strand  $k$  of chromosome  $k$ 
                 $m_{jkl}^g \leftarrow z_i^g$ 
                 $m_{jkl}^a \leftarrow z_i^a$ 
                 $m_{jkl}^p \leftarrow Z_{\bullet i}^p$ 
                 $m_{jkl}^b \leftarrow Z_{\bullet i}^b$ 
                 $m_{jkl}^c \leftarrow z_i^c$ 
                 $m_{jkl}^s \leftarrow z_i^s$ 
                Update metabolite copy numbers:  $m \leftarrow m + M_{\bullet i}$ 

```

C.3.5 DNA Repair

Biology

DNA is susceptible to several intrinsic and extrinsic modes of damage (see Section C.3.4). Because damaged DNA is deleterious to many processes including transcription and replication^{68,409–411}, organisms have evolved specialized machinery to detect and repair damaged DNA. Eisen and Hanawalt have shown that the *M. genitalium* genome contains the DNA damage sensor DisA and four DNA repair pathways: direct damage reversal (DDR), base excision repair (BER), global genomic nucleotide excision repair (GG-NER), and homologous recombination double strand break repair (HR-DSBR)¹¹². In addition, *M. genitalium* employs the type II restriction/modification (R/M) system MunI to selectively degrade foreign DNA.

Reconstruction

Damage Recognition

M. genitalium employs only one dedicated machine – DisA (MG105) – to recognize DNA damage. Bejerano-Sagie et al. have shown that *B. subtilis* DisA recognizes strand breaks, base damage, and cross links²¹. The mechanism of DisA damage recognition is not well understood. DisA is believed to signal DNA damage through the absence of the second messenger cyclic-di-AMP. In the absence of DNA damage, DisA cyclizes ATP to cyclic-di-AMP. In the presence of DNA damage, DisA is believed to bind DNA, adopt a catalytically inactive conformation, and via an unknown mechanism possibly involving cyclic-di-AMP and the transcriptional regulator Spo0A, delay sporulation. *M. genitalium*, however, does not have most of the downstream signaling machinery associated with *B. subtilis* DisA-dependent sporulation delay.

Direct Damage Reversal

DNA ligation is the only DDR pathway employed by *M. genitalium*¹¹². DNA ligation is catalyzed by DNA ligase LigA (MG254) by a NAD-dependent mechanism. DNA ligase repairs single strand breaks as well as ligates DNA synthesized during replication, BER, NER, and HR.

Base Excision Repair

The primary role of BER is to repair small patches of oxidized nucleobases^{280,369,370}. *M. genitalium* BER repairs single damaged nucleobases. *M. genitalium* does not undergo long patch BER because *M. genitalium* does not have a flap endonuclease.

M. genitalium BER repairs DNA via a four step mechanism. First, glycosylase Fpg (MG498) or Ung (MG097) hydrolytically cleaves the glycosidic bond between the damaged nucleobase and the DNA backbone, creating an abasic site. Second, two parallel subpathways cleave the phosphodiester bonds 3' and 5' to the abasic site, introducing a gap site. The first subpathway begins with cleavage of the 3' phosphodiester bond site by AP lyase Fpg and ends with cleavage of the 5' phosphodiester bond by 5'-deoxyribosephosphodiesterase Nfo. The second subpathway begins with cleavage of the 5' phosphodiester bond by 5'-AP endonuclease Nfo (MG235) and ends with cleavage of the 3' phosphodiester bond by 3'-(deoxyribose-5'-phosphate) lyase DnaN (MG001). Third, DNA polymerase DnaN restores the missing base using the template provided by the opposite strand. Finally, DNA ligase ligates the inserted base.

Nucleotide Excision Repair

The primary role of NER is to repair bulky distortions in the shape of the DNA helix of up to 12-13 bases in length caused by UV radiation and nitric oxide. *M. genitalium* NER provides global protection against small DNA lesions. *M. genitalium* does not undergo transcription-coupled NER because *M. genitalium* does have the transcription-repair coupling factor *mfd*. Compared to BER, NER employs less specific endonucleases and consequently has broader repair capacity.

M. genitalium NER repairs DNA via a four step mechanism. First, UvrABC (MG421, MG073, MG206) identifies a DNA lesion and cleaves the phosphodiester bonds 6-7 bases 3' and 5' to the lesion. Second, helicase PrcA excises the cleaved DNA. Third, DNA polymerase DnaN (MG001) restores the missing bases using the template provided by the opposite strand. Finally, DNA ligase ligates the inserted base.

Homologous Recombination Double Strand Break Repair

HR repairs double strand breaks caused by ionizing radiation and stalled replication forks. HR also repairs strand gaps generated by the interaction of replication forks with unrepaired lesions³⁶⁶. *M.*

genitalium has a very reduced complement of recombination repair proteins. Recombination repair is modeled as an eight step process:

1. Initiation: 5'-3' exonuclease removes dNMPs from the 5' ends of the strand break, leaving 3' overhangs of at least 8 bases⁸⁵. No traditional initiation gene has been identified in *M. genitalium*. We assumed that polI-like 5'-3' exonuclease (PolA, MG262) is the *M. genitalium* HR initiator.
2. Strand exchange: RecA (MG339) catalyzes the formation of a Holliday junction between one of the damaged 3' overhangs and the undamaged homologous chromosome.
3. Polymerization: DnaN (MG001) polymerizes DNA guided by the undamaged chromosome template.
4. Ligation: LigA (MG254) ligates the newly produced DNA.
5. Second strand exchange: RecA catalyzes the formation of a second Holliday junction.
6. Holliday junction migration: RuvA and RuvB widen the distance between the two strand cross over points by moving the Holliday junctions to the preferred sequence 5'-[AT]TTN[GC]-3'⁸⁹.
7. Resolution: RecU nicks the DNA at the cross over points, creating four single strand breaks.
8. Ligation: LigA ligates the four strand breaks.

Additionally, HR is believed to be critical for *M. genitalium* antigenic variation³⁶⁶.

Restriction/Modification

Organisms employ R/M systems to distinguish foreign from self DNA. These systems maintain self DNA in a fully methylated configuration by methylating self DNA during chromosomal replication and cleaving unmethylated foreign DNA which has not been exposed to self DNA methylases. *M. genitalium* contains a reduced R/M system consisting of the type I DNA recognition subunit EcoD (MG438) and the type II methylase MunI (MG184). EcoD is the DNA binding domain of a type I multimeric methylation and restriction complex which recognizes the sequence 5'-TCARTTC-3'. Because *M. genitalium* contains only the DNA recognition subunit and not the methylation and restriction subunits, we do not model type I R/M. MunI methylates the third base of both strands of the palindromic sequence 5'-CAATTG-3', producing N⁶-methyladenine. Because *M. genitalium* does not contain a separate type II endonuclease and because type II R/M systems are often monomeric, we assumed MunI also has restriction activity.

Metabolism of Damaged Nucleo-bases, -sides, and -tides

The metabolic byproducts of DNA repair are salvaged by oligonucleases. However, the mechanism of *M. genitalium* oligonucleotide salvage is not well understood. No gene has been identified which cleaves DNA oligonucleotides. Candidate oligonucleotide salvage genes include *pcrA* (MG244), *mgpA* (MG190), and *polA* (MG262). Of these genes, we assumed that *mgpA* is responsible oligonucleotide salvage. Individual damaged nucleotides are further metabolized and exported. See Section C.3.10 for further discussion.

Reaction Stoichiometry & Kinetics

The DNA specificity and geometry, stoichiometry, and kinetics of each DNA repair reaction were reconstructed based on an extensive review of the primary literature^{21,37,51,71,79,313,326,333,369,370,435,448,34,42,68,85,89,185,191,265,280,294,306,327,359,361,408,434,439}. Table S3O lists the reconstructed stoichiometry and kinetics of each reaction. Table S3D lists the reconstructed DNA specificity and geometry of each DNA repair reaction.

Computational Representation

Mathematical Model

This process models four DNA repair pathways and methylation and restriction of MunI R/M sites. Because DNA repair is not well understood on the genomic level, this process makes several simplifying assumptions. First, this process represents each step in DNA repair, methylation, and restriction as a separate reaction, and assumes that each reaction is independent. Consequently, the rate of each reaction is determined only by the configuration of the DNA. Second, this process assumes that the mean arrive rate, v_i , of each reaction i is independently limited by (1) the copy numbers of intracellular metabolites, m_j , and (2) the DNA repair enzyme copy numbers, e_j . Based on these assumptions, the function form of v_i is given by

$$v_i = \min \left(\left[\overbrace{\frac{m_j}{M_{ji}^s}}^{\text{metabolites}} \right], \left[\overbrace{\text{poissonRand} \left(\frac{e_j}{K_{ji}} \Delta t \right)}^{\text{enzymes}} \right] \right) \quad (\text{C.12})$$

where M_{ji} is the stoichiometry of metabolite j in reaction i , $M^s = \max(0, -M)$ is the negative part of M , K_{ji} is the experimentally observed catalytic rate of enzyme j in reaction i , and $\Delta t = 1\text{ s}$ is the simulation time step.

This process also stochastically models the binding of free DisA to DNA lesions.

Integration

The **Chromosome** state represents the copy number, modification status, and protein occupancy of each chromosome. The **Protein Monomer** and **Protein Complex** states represent the free and DNA-bound copy numbers of each protein species. The **Metabolite** state represents the copy number of each metabolite.

The **DNA Damage** process models spontaneous and chemical- and radiation-induced DNA modification. The **Metabolism** process models DisA diadenylate cyclase activity and modified nucleotide catabolism.

Initial Conditions

The **Chromosome** state initializes one chromosome, modified only at R/M sites. Several other processes including **Transcription** initialize the protein occupancy of the chromosome.

Dynamic Computation

Algorithm C.6 outlines the implementation of the DNA repair model.

Algorithm C.6. DNA repair simulation.

Repair, methylate, and restrict DNA

foreach reaction i in a random order **do**

foreach base j of strand k of chromosome l susceptible to reaction i in a random order **do**

if sufficient enzymatic and metabolic resources to support reaction i **then**

- Execute reaction i
- Update DNA configuration
- Update metabolite copy numbers
- Decrement enzymatic capacity

Bind DisA to damaged DNA

while there is free DisA and at least 1 DisA-accessible DNA lesion **do**

- Let $i, j, k \leftarrow$ represent the base, strand, and chromosome of a DisA-accessible DNA lesion
- Bind DisA to base i of strand j of chromosome k

C.3.6 DNA Supercoiling

Biology

DNA naturally exists at a certain level of helicity, and this level of helical density is important for the DNA's stability, its ability to fit in the cell, and its ability to bind proteins⁴²⁴. DNA supercoiling can also have an effect on RNA transcription rates as the helicity of the DNA may make given genes more or less accessible³⁰³.

Reconstruction

M. genitalium has three topoisomerase proteins: DNA gyrase, topoisomerase I, and topoisomerase IV. These proteins transiently break a DNA strand to wind (topoisomerase I) or unwind (topoisomerase IV, gyrase) the DNA. The opposing actions of the topoisomerase enzymes help maintain a stable level of DNA helicity³⁰⁰. This is especially important during replication because while the replication loops progress along the chromosome unwinding the DNA, the coils that previously existed in the DNA persist and accumulate downstream of the replication loop. This over-coiled region is relieved by the actions of topoisomerases⁴⁴⁶.

All of the enzymes and parameters used in the DNA Supercoiling process class are described in Table C.11 and Table C.12.

Table C.11. Enzymes and complexes used in the DNA Supercoiling process class.

Enzyme/Complex	Composition	Gene Name(s)	DNA Footprint (nt)
DNA gyrase	(2) MG003, (2) MG004	<i>gyrB</i> , <i>gyrA</i>	140
Topoisomerase IV	(2) MG203, (2) MG204	<i>parE</i> , <i>parC</i>	34
Topoisomerase I	(1) MG122	<i>topA</i>	19

Table C.12. Fixed parameters used in the DNA Supercoiling process class.

Parameter	Value	Symbol	Source
Supercoiling			
Bases per turn in relaxed DNA	10.5		
Equilibrium superhelical density	-0.06	σ_0	
Supercoils introduced by 1 gyrase event	-2	δ_{gyr}	284.
Supercoils introduced by 1 topo I event	1	δ_{topI}	91.
Supercoils introduced by 1 topo IV event	-2	δ_{topIV}	46.
Strand passing events rate per gyrase	1.2 s^{-1}	k_{gyr}	46.
Strand passing events rate per topo I	1 s^{-1}	k_{topI}	91.
Strand passing events rate per topo IV	2.5 s^{-1}	k_{topIV}	384.
ATP hydrolyzed per gyrase event	2	E_{gyr}	136.
ATP hydrolyzed per topo I event	0	E_{topI}	46.
ATP hydrolyzed per topo IV event	2	E_{topIV}	46.
Superhelical denisty threshold, gyrase	-0.1	T_{gyr}	46.
Superhelical denisty threshold, topo I	0	T_{topI}	91.
Superhelical density threshold, topo II	0	T_{topII}	46.
Gyrase logistic function steepness parameter	100	L_{gyr}	Fit
Topo I losgistic function steepness parameter	-100	L_{topI}	Fit
Number of bases in the chromosome	580076		
Enzyme DNA footprints	See Table C.11		284, 300, 424.
Processivity			
Mean time gyrase stays bound to the DNA	45 s		46.
Supercoiling's Effect on Gene Expression			
Slope of gyrase σ -gene expression data	42.8		303.
Slope of topo I σ -gene expression data	-7.4		303.
Slope of topo IV σ -gene expression data	1.1		303.
y-intercepts of gyrase σ -gene expression data	3.57		303.
y-intercepts of topo I σ -gene expression data	0.56		303.
y-intercepts of topo IV σ -gene expression data	1.07		303.
σ bounds for applying linear fit of gene expression	-0.08, 0.07	$\sigma_{lower}, \sigma_{upper}$	Fit
data			

Parameter Assignment

The logistic function parameters (L_{gyr} , L_{topI}) were fit such that the equilibrium superhelical density, σ_0 , could be maintained while still allowing both gyrase and topoisomerase I to act at the equilibrium σ_0 . Linear fits were approximated for gene expression data experimentally measured at varying σ . The upper and lower bounds of σ for applying the linear fit of the σ -gene expression data (σ_{lower} , σ_{upper}) were set in there range where a linear fit was appropriate for the data.

Computational Representation

The supercoiling of the DNA is quanitfied using a metric known as the DNA linking number, LK . The LK_{relaxed} is defined as (# of base pairs)/10.5, where 10.5 is the number of bases per turn in a relaxed double helix. The LK_{current} may deviate from the LK_{relaxed} due to the actions of topoisomerases and the progression of replication. The ΔLK is defined as the difference between the current level of DNA supercoiling and the relaxed level of DNA supercoiling. DNA gyrase and topoisomerase IV each require 2 ATP to induce 2 negative supercoils each time they act⁴¹⁷. Topoisomerase IV induces positive supercoils relatively rarely and its positive coiling activity is therefore not considered in our model⁴¹⁷. Topoisomerase I acts to induce positive supercoils⁹¹. Gyrase, topoisomerase IV, and topoisomerase I act at rates of 1.2, 2.5, and 1 strand passing events per second respectively^{46,91,384}.

The helicity of different regions on the DNA are modeled separately. Before replication, the entire chromosome is represented as one region. During replication, two additional regions are defined and accounted for: the replicated DNA upstream of the replication loops on each of the two chromosomes (see Figure C.7). During replication, the unreplicated region of DNA (region downstream of the two replication loops) is shrinking in terms of number of bases (see Figure C.7), and as a result, the LK_{relaxed} decreases. However, as the DNA replicative helicases unwind the DNA they push the existing coils into this region, and there are more coils for fewer bases. This over-coiled region has a LK_{current} that is too high, and must be brought down towards the LK_{relaxed} . Gyrases and topoisomerases help bring the DNA back to the relaxed state by inducing negative supercoils. It is assumed that the newly replicated DNA is made in the relaxed state: $LK_{\text{current}} = LK_{\text{relaxed}}$, but the supercoiling enzymes act in this region as well, helping maintain equilibrium helicity. It is essential that the ΔLK in the region downstream of the replication loops be brought down to 0 by the end of replication. After replication is complete, only the two replicated regions exist (2 separate chromosomes). While in general, DNA supercoiling involves the regions described above,

DNA damage and other processes can cause gaps in the DNA that result in additional regions.

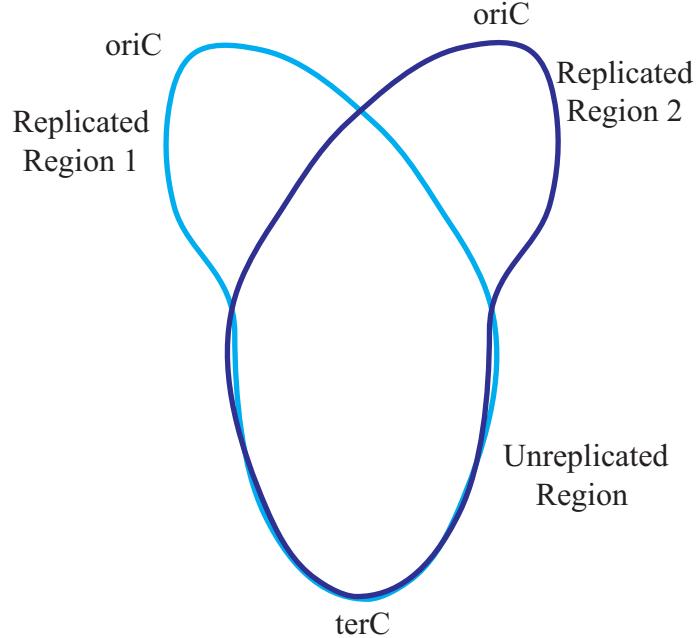


Figure C.7. Regions of varying superhelical density on the replicating chromosome.

The superhelical density of a DNA region, σ , is defined as the $(LK_{\text{current}} - LK_{\text{relaxed}})/LK_{\text{relaxed}}$. Experimentally it has been shown that the steady state superhelical density, σ_0 , is -0.06^{46} . The activity of topoisomerases depends on the σ of the DNA⁴⁶. Although there is likely a more complex relationship between the DNA supercoiling and enzyme activity, enzyme activity is modeled as a set of step functions and logistic functions (see Figure C.8). It is known that the topoisomerases are unable to act beyond certain σ thresholds (gyrase can only act above T_{gyr} , topoisomerase IV can only act above T_{topIV} , and topoisomerase I can only act below T_{topI}), so the activities of the topoisomerases outside of the thresholds are set to zero⁴⁶. However, the exact enzyme activity profiles within these bounds is unknown. Fitting a logistic function within the thresholds of gyrase and topoisomerase I allows the maintainance of σ approximately near σ_0 . Topoisomerase IV is not active near σ_0 and therefore acts more rarely than the other topoisomerases. As a result, there was no need to assume a logistic profile for its activity. The proportion of full activity for each topoisomerase was obtained from the profiles. This proportion was multiplied by the maximal activity to determine the number of strand passing events for each topoisomerase for a given timestep.

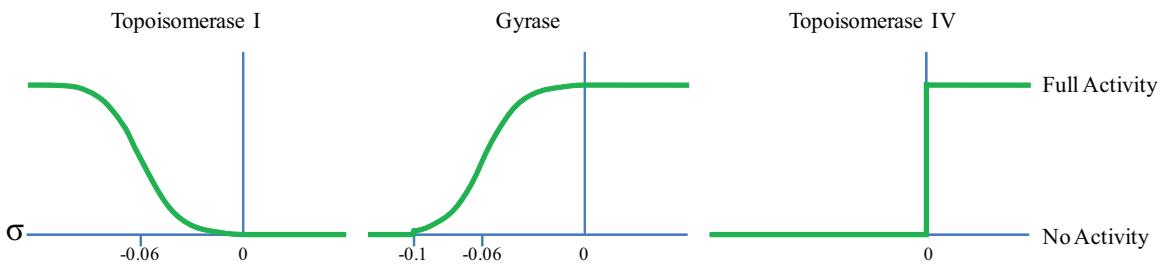


Figure C.8. Enzyme activity profiles.

Further, the model keeps track of enzyme processivity, or how long an enzyme can remain bound to the DNA and continue to act. Topoisomerase IV and DNA gyrase are highly processive and may perform several strand-passing events before falling off of the DNA. Topoisomerase I is not highly processive, it acts on the DNA and falls back off right away.

Finally, it has been shown that there is an effect of supercoiling on transcription rates, and experimental data in *E. coli* compares superhelical density to fold change of gene expression³⁰³. While expression fold changes at different superhelical densities have been calculated for many *E. coli* genes, none have been measured for *M. genitalium*, and therefore only the effects of supercoiling on the expression of the 5 supercoiling genes *gyrB*, *gyrA*, *parC*, *parE*, and *topA*, are included. These genes exist in 3 transcription units. Peter et al. have *E. coli* data for the fold change of expression (microarray data) of each of these genes at various values of σ (ranging from σ -0.06 to 0.02) at various experimental conditions³⁰³. Despite the large variation of data in a narrow σ range, we have decided to use linear fits of the data, as they fit the general trends in the data: increased helicity leads to decreased or increased gene expression. The linear fits are constrained such that the fold change is 1, at equilibrium σ , -0.06. The linear fits are extrapolated between the σ range of -0.08 to 0.07, where all the fold changes remain above 0. Outside of this range, a constant fold change of expression is estimated. While there is a separate set of data for each of the 5 genes, the **Transcription** process class requires a single probability of transcription for each transcription unit. The data for the first gene in each transcription unit is used (*gyrB* and *parE*). *topA* is transcribed with genes that are not supercoiling related. The expression of those genes (MG119, MG120, MG121) is also affected by supercoiling. The general trend is that gyrase and topoisomerase IV will have an increased expression when σ is higher than the equilibrium, and that topoisomerase I will have a higher expression when σ is lower than the equilibrium. The profiles of fold change in promoter binding probabilities for varying σ can be seen in Figure C.9.

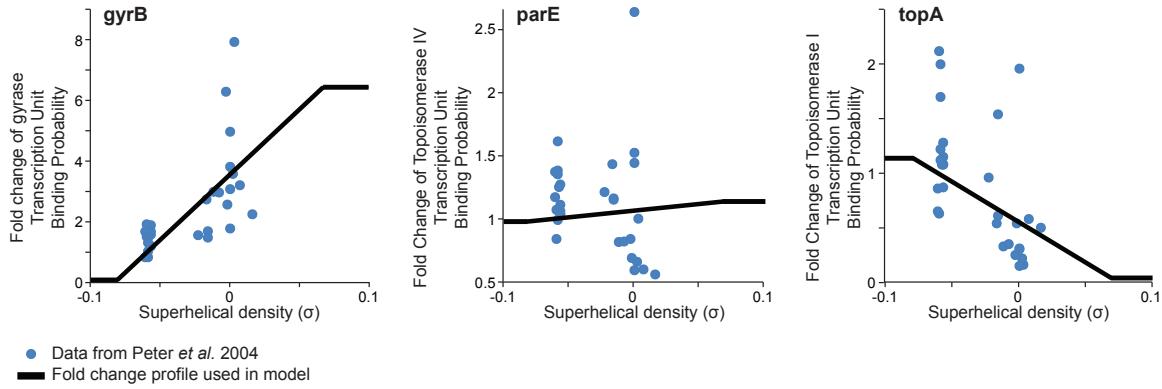


Figure C.9. Fold change in the probability of RNA polymerase-gene promoter binding for three transcription units.

Integration

Table C.13. State classes connected to the DNA Supercoiling process class.

Connected State	Read from state	Written to state
Chromosome	<ul style="list-style-type: none"> Accessible positions on the DNA for topoisomerases to bind Start positions of transcription units affected by supercoiling Linking number, length, and start sites of regions of DNA (considered independently in terms of superhelical density) 	<ul style="list-style-type: none"> Positions where topoisomerases are bound Fold change of transcription unit-RNA polymerase binding probabilities Updated linking number of regions of DNA

Initial Conditions

At the start of the simulation, the entire chromosome exists in a single double stranded region. The linking number of this region is set such that the superhelical density is at equilibrium, -0.06. All of the existing gyrase are bound to randomly designated positions around the chromosome. The transcription probabilities of the affected transcription units are adjusted according to the superhelical density.

Dynamic Computation

The activities of DNA gyrase, topoisomerase I, and topoisomerase IV, and the progress of replication loops can all affect the linking number (and superhelical density) of the DNA. The `DNA_Supercoiling` process class, evaluates the effects of all the free topoisomerases in random order, determines what regions they can bind in, and randomly binds them to large enough open positions on the DNA. The linking number is then adjusted based on the actions of all the bound enzymes, and the usage of ATP is determined. The processivity of gyrase and topoisomerase IV are also tracked so that they can be unbound from the DNA when appropriate.

Algorithm

1. Determine regions of DNA

The superhelicity of the chromosome is tracked separately for different each double stranded region. The exact base positions and length in base pairs, l , of each region is determined based on the progress of replication and positions of breaks in the DNA. If there are no breaks in the DNA, the following DNA regions exist:

Before Replication:

- Region 1: Unreplicated DNA

During Replication:

- Region 1: Replicated DNA, Chromosome 1 (upstream of replication loops, increasing in length over time)
- Region 2: Replicated DNA, Chromosome 2 (upstream of replication loops, increasing in length over time)
- Region 3: Unreplicated DNA (downstream of replication loops, decreasing in length over time)

After Replication:

- Region 1: Replicated DNA, Chromosome 1
- Region 2: Replicated DNA, Chromosome 2

2. Calculate σ at each DNA region

The enzymes may act on the chromosome affecting the LK_{current} , and experimentally it has been shown that the enzymes would obtain an LK_{current} such that the steady state $\sigma = -0.06^{446}$. For each region of DNA, the LK_{current} is stored across timesteps. The superhelical density, σ , of each region is calculated using the LK_{current} as follows:

$$LK_{\text{relaxed}} = \frac{l}{10.5}$$

$$\sigma = \frac{LK_{\text{current}} - LK_{\text{relaxed}}}{LK_{\text{relaxed}}}$$

3. Determine which topoisomerases can act in which DNA regions

Gyrase, topoisomerase I, and topoisomerase IV can only act on the DNA if the superhelical density (σ) is within a certain range. A combination of step functions and logistic functions were used to determine the activity of the enzymes for a given σ . The activity profiles can be seen in Figure C.8. An enzyme can act in a DNA region if its activity profile is greater than zero at the current σ .

4. Unbind Processive Enzymes

Topoisomerase IV and DNA gyrase are processive enzymes meaning that they may perform several strand-passing events before falling off of the DNA. Topoisomerase I is not highly processive, it acts on the DNA and immediately dissociates⁴⁶.

Topoisomerase IV is highly processive and will stay bound to the DNA as long as σ is greater than zero.

- For each region, if $\sigma \leq 0$,
 - unbind all Topoisomerase IVs

Gyrase will stay bound to the DNA for about 30-60 s. Gyrase processivity is modeled as a Poisson distribution with $\lambda = 45$ s.

- For each bound gyrase,
 - if $\text{rand} < 1/\lambda$, unbind gyrase.

5. Bind Enzymes to Each Region

Topoisomerases may bind to the DNA without acting on the DNA. Therefore, binding is only limited by the availability of free topoisomerases and space on the DNA.

- Randomize the order of binding each of the three topoisomerases, to fairly allocate space on the chromosome
 - For each of the three topoisomerases i ,
 - For each region j where the σ permits topoisomerase i binding,
 - Stochastically bind topoisomerases i in region j , up to:
- $$\max \begin{cases} \text{Available positions in DNA region}_j \text{ of length } > \text{footprint}_i \\ \text{Number of free topoisomerase}_i \end{cases}$$

6. Calculate new Linking Numbers and Perform Metabolite Accounting

The DNA linking number is affected by the bound topoisomerases, up to the limits of available resources and kinetic rates.

- For each DNA region j ,
 - For each of the three topoisomerases i that is able to bind region j given σ ,
 - Count the number of topoisomerase i bound in region j (C_{ij})
 - Use the logistic/step functions (See Figure C.8) to determine the Probability (P_{ij}) of topoisomerase i acting in region j at the current σ
 - Determine the number of strand passing events (S_{ij}) for enzyme i in region j , given the kinetic rate of topoisomerase i (k_i) and the energy requirement per strand passing event for topoisomerase i (E_i):

$$S_{ij} = \min \begin{cases} \text{round}(C_{ij}k_iP_{ij}) \\ \frac{\text{available ATP}}{E_i} \\ \frac{\text{available H}_2\text{O}}{E_i} \end{cases}$$

- Calculate the new linking number (LK_{current}) for region j , given topoisomerase i 's effect on the linking number (δ_i):

$$LK_{\text{current}} = LK_{\text{current}} + \delta_i S_{ij}$$

- Account for used metabolites, ATP and H_2O , and produced metabolites, ADP, Pi,

and H⁺

- Unbind all topoisomerase I, a non-processive enzyme

7. Determine the effects of supercoiling on transcription

Supercoiling effects the expression of the genes: *gyrB*, *gyrA*, *parC*, *parE*, and *topA*, which lie in three transcription units. The DNA Supercoiling process calculates the fold change in the probability of an RNA polymerase binding to the promoters of these transcription units.

- For each topoisomerase transcription unit *i*,
 - Find the double stranded region where transcription unit *i* lies and the σ of this region
 - Use the fold change profiles (See Figure C.9) to determine the fold change in probability of polymerase binding to transcription unit *i*

C.3.7 FtsZ Polymerization

Biology

Cell division in many bacterial species requires the assembly of an FtsZ ring at the cell membrane around the midplane of the cell⁷. FtsZ is a homologue of eukaryotic tubulin that assembles into long polymers. These polymers are typically localized to the center of the cell, forming a membrane-bound ring⁷. FtsZ is a GTPase, and GTP hydrolysis to GDP causes the FtsZ filaments to bend. This bending serves as one of the forces enabling cell division²¹⁶. (For more information about cell division, see the Cytokinesis process class.) GTP only hydrolyzes to GDP at an active site formed at the interface of two FtsZ molecules, meaning that FtsZ polymerization is required for GTPase activity²¹⁶. This process addresses the assembly FtsZ monomers into long polymers.

Reconstruction

This process addresses the assembly FtsZ (MG224) monomers into long polymers.

All of the parameters used in the FtsZ Polymerization process class are described in Table C.14.

Table C.14. Fixed parameters used in the FtsZPolymerization process class.

Parameter	Value	Symbol	Source
Maximum length of an FtsZ filament	9 subunits	n	216.
Activation rate	1.1 s^{-1}	k_{act1}	62.
Deactivation rate	0.01 s^{-1}	k_{act2}	62.
Exchange rate, GTP→GDP	$1 \times 10^4 \text{ M}^{-1} \text{ s}^{-1}$	k_{ex1}	388.
Exchange rate, GDP→GTP	$5 \times 10^3 \text{ M}^{-1} \text{ s}^{-1}$	k_{ex2}	388.
Nucleation rate	$4.2 \times 10^6 \text{ M}^{-1} \text{ s}^{-1}$	k_{nuc1}	62, 388.
Denucleation rate	40 s^{-1}	k_{nuc2}	62, 388.
Polymerization rate	$5.1 \times 10^6 \text{ M}^{-1} \text{ s}^{-1}$	k_{el1}	388.
Depolymerization rate	2.9 s^{-1}	k_{el2}	388.

Computational Representation

Typically, FtsZ polymers of variable length are able to bind the cell membrane to form contractile rings. The mean filament length in *E. coli* has been measured to be 100 nm (23 FtsZ monomers)⁷. The minimum fragment length observed in *C. crescentus* is 40 nm (9 FtsZ monomers)²¹⁶. Being such a small organism, we hypothesize that *M. genitalium* would have filaments on the lower end of what is sufficient for contractile force generation. For simplicity of calculating the **Cytokinesis** progress, we use polymers of a fixed length, 9 subunits²¹⁶.

This process class simulates polymerization of FtsZ-GTP monomers to 9mers. We use a differential equation model developed by Surovtsev et al. to assemble the polymers³⁸⁸. During polymerization, FtsZ-GTP can exist in any intermediate state: 1mer, 2mer, ... 8mer, 9mer. The set of differential equations models the following reactions:

Activation of FtsZ monomers (F) to FtsZ-GTP (F_T)

Deactivation of FtsZ-GTP (F_T) to FtsZ (F)

Exchange of FtsZ-GDP (F_D) to FtsZ-GTP (F_T)

Reverse exchange of FtsZ-GTP (F_T) to FtsZ-GDP (F_D)

Nucleation of two FtsZ-GTP molecules (F_T) to form a dimer (F_{T2})

Reverse nucleation of a dimer (F_{T2}) into two FtsZ-GTPs (F_T)

Polymer elongation adding an FtsZ-GTP (F_T) to an existing polymer of length 2-8 subunits ($F_{T2-} F_{T8}$)

Reverse elongation removing an FtsZ-GTP (F_T) from an existing polymer of length 3-9 subunits ($F_{T3-} F_{T9}$)

Integration

The count full-length activated FtsZ (9mer) filaments is used by the **Cytokinesis** process class for cell pinching.

Initial Conditions

The FtsZ polymerization differential equations are run a number of times (up to 50 iterations) before the simulation starts, to ensure starting at a steady state distribution of polymer lengths. Iterations are run until the change in the solution across consecutive iterations is less than a set tolerance (0.2% of the enzymatic counts).

Dynamic Computation

FtsZ can exist in one of multiple states: inactivated monomer, activated monomer (GTP bound), nucleated (dimer of two activated FtsZ molecules), elongated polymer of three or more GTP bound FtsZ molecules. The hydrolyzed FtsZ polymers are only considered during cell membrane pinching in the **Cytokinesis** process class. The FtsZ molecules can move to states of higher and lower polymerization at rates obtained from Chen et al. and Surovtsev et al.^{62,388}.

FtsZ polymerization is modeled using a set of differential equations modified from that described in Surovtsev et al., involving the activation, nucleation, and elongation of FtsZ polymers³⁸⁸. The main modifications were that the equations were simplified to not include annealing and cyclization of FtsZ polymers.

The following differential equation model is evaluated at each timestep:

$$\begin{aligned}
 \frac{dF}{dt} &= k_{act2}F_T - k_{act1}F \\
 \frac{dF_D}{dt} &= k_{ex2}F_T [GDP] - k_{ex1}F_D [GTP] \\
 \frac{dF_T}{dt} &= k_{act1}F - k_{act2}F_T + k_{ex1}F_D [GTP] - k_{ex2}F_T [GDP] - 2k_{nuc1}F_T^2 + \dots \\
 &\quad + \dots 2k_{nuc2}F_{T2} - k_{el1}F_T \left(\sum_{i=2}^8 F_{Ti} \right) + k_{el2} \left(\sum_{i=3}^9 F_{Ti} \right) \\
 \frac{dF_{T2}}{dt} &= k_{nuc1}F_T^2 - k_{nuc2}F_{T2} - k_{el1}F_T F_{T2} + k_{el2}F_{T3} \\
 \frac{dF_{Ti}}{dt} &= k_{el1}F_T F_{Ti-1} - k_{el2}F_{Ti} - k_{el1}F_T F_{Ti} + k_{el2}F_{Ti+1}, \text{ for } i \in 3..8 \\
 \frac{dF_{T9}}{dt} &= k_{el1}F_T F_{T8} - k_{el2}F_{T9}
 \end{aligned}$$

Solving these equations results in a real-value distribution of monomers and filament lengths at each timestep. The model framework requires the storage of the count of each molecular entity rather than the concentration. This process class discretizes the distribution at each time step for compatibility with the rest of the simulation.

C.3.8 Host Interaction

Biology

M. genitalium is a common extracellular urogenital epithelial parasite. Although *M. genitalium* is estimated to colonize the urogenital tract of 3% of all women, most infections are asymptomatic, and most researchers regard *M. genitalium* as an “ideal parasite” which lives in harmony with its human host³²⁵. Only a small fraction of *M. genitalium* infections manifest clinically as urethritis, cervicitis, or pelvic inflammatory disease.

Reconstruction

Although *M. genitalium* is a common urogenital pathogen, the pathophysiology of *M. genitalium* is only beginning to be elucidated. Razin and colleagues have studied the *Mycoplasma* terminal organelle, and have shown that several lipoproteins (MG191, MG192, MG217, MG318, and MG386) are involved in host adhesion and activation of the host immune response^{324,325}. Shimizu et al. and

McGown et al. have identified 3 additional lipoproteins – MG149, MG309, and MG412 – which activate the host immune response by stimulating Toll-like (TLR) receptors 1, 2, and 6, which in turn activate the host transcriptional regulator NF- κ B^{244,357,358}. Additionally, Duffy et al. have reported that MG075 is immunoreactive¹⁰¹. Razin and colleagues have suggested that *Mycoplasma* also stimulates the host immune response by secreting hydrogen peroxide and superoxide radicals which cause membrane oxidative damage and trigger inflammation³²⁵.

Computational Representation

Mathematical Model

Because the interaction between *M. genitalium* and its human host is not well understood, this process implements a Boolean model of the effect of the *M. genitalium* terminal organelle and accessory lipoproteins on six properties of the host human urogenital epithelium:

- Adherence – Mycoplasma adheres to its host if its terminal organelle is properly formed and all of its adhesion proteins are expressed.
- TLR 1, 2, and 6 activation – Host TLR 1 is activated if the bacterium is adherent and MG149 or MG412 is expressed. TLR 2 is activated if the bacterium is adherent and MG149, MG309, or MG412 is expressed. TLR 6 is activated if the bacterium is adherent and MG309 is expressed.
- NF- κ B activation – NF- κ B is activated if TLRs 2 and 1 or 6 are active.
- Inflammatory response activation – The host inflammatory response is activated if either NF- κ B is active or the bacterium is adherent and at least one of the MG075, MG149, MG309, or MG412 antigens is expressed.

Integration

The **Host** state uses six Boolean variables to represent the status of the human urogenital epithelium. The **Protein Monomer** state represents the copy number of each terminal organelle and accessory lipoprotein synthesized and matured by the protein maturation pathway (see Section C.2.10) and translocated into the terminal organelle (see **Terminal Organelle Assembly** process).

Initial Conditions

After the **Protein Monomer** and **Terminal Organelle Assembly** states initialize the copy number of each protein, the **Host** state is initialized to the steady-state of the **Host Interaction** Boolean

model. Because the Boolean model is acyclic, the model converges in one iteration.

Dynamic Computation

At each time step, this process evaluates the six Boolean rules outlined in Algorithm C.7 until convergence. Because the Boolean rules are acyclic, the model converges in one iteration.

Algorithm C.7. Host-parasite interaction simulation.

```

adherent ← organelle proteins MG191, MG192, MG217, MG218, MG312, MG317, MG318, and MG386
expressed
tlr1 ← adherent AND (lipoproteins MG149 and MG412 expressed)
tlr2 ← adherent AND (lipoproteins MG309 expressed)
tlr6 ← adherent AND (lipoproteins MG149, MG309, and MG412 expressed)
nfkb ← tlr2 AND (tlr1 OR tlr6)
inflammation ← nfkb OR (adherent AND MG075, MG149, MG309, or MG412 antigen expressed)

```

C.3.9 Macromolecular Complexation

Biology

Many enzymes are functional as multimeric proteins or ribonucleoproteins. Macromolecular complexes are believed to form quickly ($k_{on} \approx 10^3 - 10^6 \text{ M}^{-1} \text{ s}^{-1}$), energetically favorably ($\Delta G \approx -12 \text{ kcal mol}^{-1}$), and stably ($K_D \approx 10^{-4} - 10^{-10} \text{ M}$). This process models the formation of all macromolecular complexes except the 30S and 50S ribosomal particles, the 70S ribosome, the FtsZ ring, and the oriC DnaA complex. The **Ribosome Assembly**, **Translation**, **FtsZ Polymerization**, and **Replication Initiation** processes model the more complex and better characterized formation of these complexes.

Reconstruction

An extensive review of the primary literature and several databases (see Table S3AI) suggested that *M. genitalium* forms 201 distinct macromolecular complexes containing 269 protein and 5 RNA gene products^{59,71,109,188}. Table S3N lists the reconstructed composition of each *M. genitalium* macromolecular complex.

Computational Representation

Mathematical Model

This process models the formation of macromolecular complexes as a stochastic process. Because macromolecular complexation is poorly understood, this process makes several simplifying assumptions. First, this process assumes that macromolecular complexes form spontaneously, uncoupled to other chemical reactions and without assistance from chaperones or other proteins. The contribution of chaperones to the three-dimensional folding of individual protein monomers is modeled by the **Protein Folding** process. Second, this process assumes that macromolecular complexation is fast and proceeds to completion within the 1 s time step of the simulation. Third, this process makes the simplifying assumption that each macromolecule complex forms with the same specific rate. Fourth, this process assumes that complexes form by simultaneous collision of each subunit. Together these assumptions imply that the relative formation rate, r_i of each complex, i , is described by mass-action kinetics,

$$r_i = \prod_j \left(\frac{m_j}{V} \right)^{S_{ij}}, \quad (\text{C.13})$$

where m_j is the copy number of gene product j , V is the cell volume, and S_{ij} is the stoichiometry of subunit j in complex i . This process models complex formation by (1) calculating the relative formation rate of each complex, r_i , (2) randomly forming one complex according to a multinomial distribution defined by the relative formation rates, (3) updating the copy numbers of RNA and protein subunits and complexes, and (4) repeating until insufficient protein and RNA subunits are available to form additional complexes. Importantly, this algorithm resolves the order of macromolecular complex formation before several subunits participate in multiple complexes.

Integration

The **Rna** and **Protein Monomer** states represent the copy number of each RNA and protein subunit synthesized by the RNA and protein synthesis pathways (see Section C.2.12 and C.2.10). The **Protein Complex** state represents the copy number of each complex, c_i . The **Geometry** state represents the cell volume.

Initial Conditions

After the cell volume and the total copy number of each RNA and protein gene product are initialized (see **Geometry**, **Rna**, and **Protein Monomer** states), the copy number of each macromolecular complex is initialized by iteratively evaluating the dynamic model until convergence, or more specifically until insufficient subunits are available to form additional complexes.

Dynamic Computation

Algorithm C.8 outlines the implementation of the macromolecular complexation model.

Algorithm C.8. Macromolecular complexation simulation. See Mathematical Model above for mathematical notation.

```

repeat
  foreach protein complex  $i$  do
    Calculate relative formation rate,  $r_i \leftarrow \prod_j \left( \frac{m_j}{V} \right)^{S_{ij}}$ 
    Select a complex  $k$  to form according to multinomialRand(1, ri / ∑i ri)
    Increment copy number of complex  $k$ ,  $c_k \leftarrow c_k + 1$ 
    Decrement copy numbers of complex  $k$  subunits,  $m_j \leftarrow m_j - S_{k,j}$ 
  until Insufficient subunits to form additional complexes

```

C.3.10 Metabolism

Biology

To grow and replicate cells transform external nutrients into cellular mass. The first step in this process is to import nutrients from the external environment and metabolize those nutrients primarily into macromolecule building blocks. Furthermore, cells must recycle and/or export byproducts. This process models the import of extracellular nutrients and their conversion into macromolecule building blocks.

M. genitalium is believed to have adapted to the rich environment provided by its host human urogenital epithelium by massive degenerative evolution from low G+C content Gram positive bacteria, eliminating non-essential genes involved in oxidative phosphorylation, ATP synthesis via the pentose phosphate pathway, and amino acid, nucleotide, lipid, and cofactor biosynthesis^{104,122,233}. Several studies have also shown that *M. genitalium* has evolved metabolic enzymes with relaxed substrate specificity^{74,267,314,365}. *M. genitalium* is generally cultured on SP-4 medium, a complex and undefined medium based on CMRL-1066⁴¹⁵.

Reconstruction

Extracellular Medium

The *in silico* medium was based on SP-4 medium with supplementary metabolites added to facilitate *in silico* growth (see Table S3BI). First, the chemical composition of SP-4 medium was estimated from the characterized composition of each medium component (see Section C.2.7 and Table S3BI). Second, additional metabolites were added until the flux-balance analysis (FBA) metabolic model predicted non-zero growth. Finally, the concentrations of several metabolites were increased to support sustained cell growth.

Cellular Composition

The *in silico* chemical composition of *M. genitalium* was based on its reconstructed cellular mass composition (see Table S3AR-S3BG), and fit to match the 27 other modeled cellular processes (see Section C.2.5 and C.1.3).

Energy

In addition to metabolic building blocks such as nucleic and amino acids, cellular growth and maintenance requires energy. The *in silico* *M. genitalium* growth-associated maintenance energy (GAM) was based on the characterized energetic requirements of *E. coli*¹²⁰.

Reactions

The *M. genitalium* metabolic network illustrated in Figure S1B was reconstructed as previously described¹¹⁹ with guidance from the *M. genitalium* FBA metabolic model recently reported by Suthers et al.³⁸⁹. Briefly, the metabolic network was reconstructed based on homology to previously modeled organisms¹²⁰ and an extensive review of the primary literature. In particular, the metabolic network was reconstructed to support the metabolic demands of all 27 other modeled cellular processes. Table S3O lists the reconstructed metabolic reactions.

Kinetics

Reaction kinetics were curated primarily from the proteomic databases SABIO-RK⁴³⁶, BRENDA⁵⁹, and BioCyc¹⁸⁸ (see Table S3AL). The maximum exchange rates of carbon- and non-carbon-containing

metabolites were set to 12 and 20 mmol gDCW⁻¹ h⁻¹, comparable to recent FBA models of *M. genitalium*³⁸⁹ and *E. coli*¹²⁰ metabolism.

Computational Representation

Mathematical Model

This process models several critical cellular functions including (1) the uptake of external nutrients, (2) the synthesis of the metabolic building blocks and intermediate energy carriers, (3) lipid assembly, membrane insertion, and maturation, and (4) the recycling and export of the metabolic byproducts of other cellular processes. This process makes two key assumptions and uses FBA^{287,402} to model *M. genitalium* metabolism. In particular, this process calculates the flux, v , of each metabolic reaction. First, this process assumes that the internal dynamics of the metabolome are fast compared to the 1 s time step of the *M. genitalium* simulation, or equivalently, that the metabolic network can be considered to be at steady-state on the 1 s simulation time scale. Second, this process assumes that the *M. genitalium* metabolic network maximizes cellular metabolite production given the available extracellular nutrients and metabolic enzymes. See Orth et al.²⁸⁷ for further discussion of the assumptions and mathematical formalism of FBA.

To be compatible with the 27 other modeled cellular processes, the metabolic sub-model has four key differences from most FBA metabolic models. First, the metabolic network, S , and the cellular mass production pseudoreaction are expanded to produce all of the metabolites required by the 27 other processes, where S_{ij} is the stoichiometry of metabolite i in reaction j and S_{ib} is the stoichiometry of metabolite i in the cellular mass production pseudoreaction, b . Second, internal exchange reactions are added to recycle the metabolic byproducts of the other processes (e.g. recycle ADP and Pi to ATP), that is to exchange metabolites between the cytosol and the metabolic network. Third, the optimization objective, g , is expanded to include the construction of new cell mass as well as the recycling and export of the metabolic byproducts of the 27 other processes. g represents the relative fitness gains of intracellular metabolite recycling and novel cell mass production, that is g represents how heavily the metabolic network favors novel cell mass production over metabolite recycling. g was set assuming that the production/exchange of each molecule contributes equally to cellular fitness. Specifically g was set such that the production/exchange of each individual molecule is weighted equally. Fourth, the flux bounds, v_l are v_u , are functions of the cell dry mass, m , the copy number and maximum exchange rate of each extracellular metabolite, n_m and $k_{\pm ex}$, the thermodynamics,

ΔE , of each reaction, and the copy number and maximum catalytic rate of each metabolic enzyme, n_e and $k_{\pm cat}$. Specifically, external exchange flux bounds are functions of the cell dry mass and the extracellular copy numbers and maximum exchange rates of the exchanged metabolites; internal exchange flux bounds are functions of the intracellular copy numbers of the exchanged metabolites; transport and chemical reaction flux bounds are functions of the copy numbers and kinetics of the enzymes and the reaction thermodynamics. Fifth, because the average energy used by the 27 other cellular processes is significantly less than the observed GAM, this process also represents the turnover of the otherwise unmodeled energy consumption. Figure C.10 and Eq. C.14 summarize the FBA metabolic model.

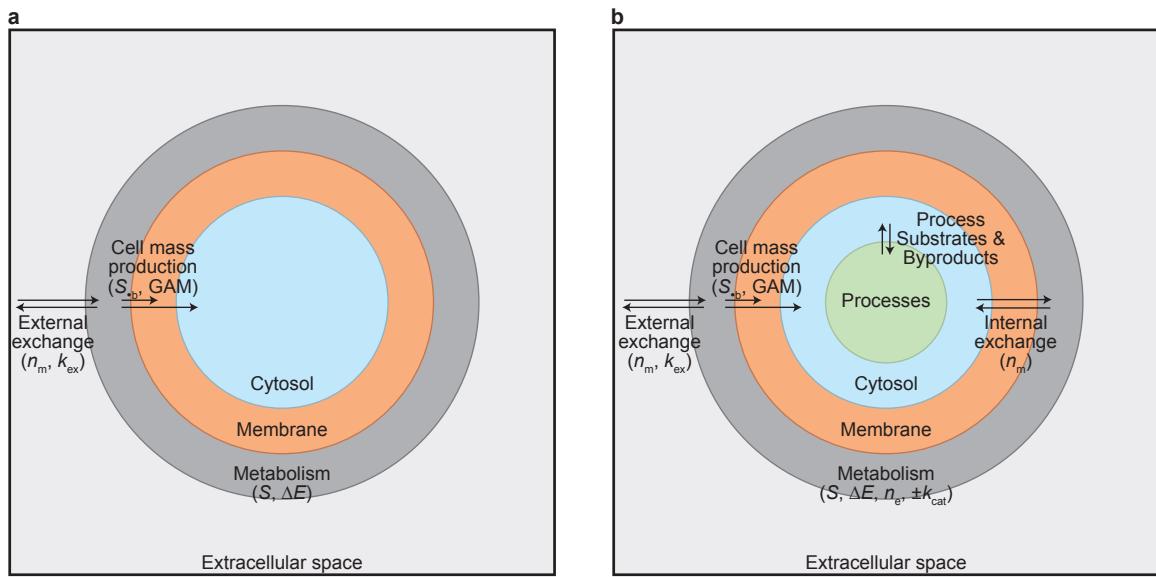


Figure C.10. Metabolite perspective of the flux-balance analysis (FBA) metabolic model. **a**, Conventional FBA metabolic model. **b**, Integrated FBA metabolic model.

$$v^* = \underset{v}{\operatorname{argmax}} g^\top v \text{ subject to} \quad (\text{C.14})$$

$Sv = 0$ and

$v_l \leq v \leq v_u$ where

$$g_i = \begin{cases} \sum_{j \neq bm} S_{j,b} & \text{pseudoreaction } i = b \text{ represents the synthesis of biomass (metabolite } bm) \\ -1 & \text{reaction } i \text{ represents an internal exchange} \\ 0 & \text{otherwise} \end{cases}$$

$$v_l = f_l(m, n_m, k_{-ex}, -\Delta E, n_e, k_{-cat})$$

$$v_u = f_u(m, n_m, k_{-ex}, \Delta E, n_e, k_{-cat})$$

Algorithm C.9 describes the functional forms of f_l and f_u in detail.

Integration

The **Metabolism** process imports extracellular nutrients and converts them into the building blocks of macromolecular synthesis. The **Metabolite** state represents both the extracellular and cellular copy numbers of each metabolite. The rates of extracellular nutrient exchange are functions of the total cell mass represented by the **Mass** state and the extracellular metabolite copy numbers. Cytosolic and membrane metabolite copy numbers limit the rates of internal exchange/recycling. The rates of chemical and transport reactions are limited by the copy numbers and kinetics of metabolic enzymes synthesized by the protein synthesis and maturation pathway (see Section C.2.10) and represented by the **Protein Monomer** and **Protein Complex** states. The **Metabolic Reaction** state records the calculated flux of each metabolic reaction. The **Simulation** object allocates metabolites produced by the **Metabolism** process to the other 27 cellular processes to support several functions including DNA, RNA, and protein synthesis and maturation. Additionally, the other 27 processes generate byproducts which the **Metabolism** process either recycles or exports from the cell.

Initial Conditions

The **Metabolite** state initializes the copy number of each metabolite. The **Protein Monomer** state initializes the total copy number of each protein monomer. The **Macromolecular Complexation**

process initializes the copy number of each macromolecular complex. This process initializes the cellular growth rate and reaction fluxes to the steady-state of the metabolic network using the same FBA simulation used at each time step.

Dynamic Computation

Algorithm C.9 outlines the implementation of the FBA metabolic model.

Fitting

The metabolic model was fit to match the observed *M. genitalium* mass doubling time, $\tau = 9\text{ h}$. Specifically, metabolic enzyme expression was fit using a modified version of minimization of metabolic adjustment (MOMA)³⁵². Let μ_o, v_o be the FBA solution. Let $\mu^* = \ln 2/\tau$ be the target growth rate. First, find a neighboring flux distribution consistent with the target growth rate. If $\mu^* < \mu_o$ find a neighboring flux distribution within the current flux bounds v_l and v_u ,

$$\begin{aligned} v^* &= \operatorname{argmin}_v \left| \frac{v - v_o}{v_o} \right| \text{ subject to} \\ Sv &= 0 \\ v_l &\leq v \leq v_u \\ v_b &= \mu^*. \end{aligned} \tag{C.15}$$

If $\mu^* > \mu_o$ find a neighboring flux distribution by relaxing the flux bounds by μ^*/μ_o ,

$$\begin{aligned} v^* &= \operatorname{argmin}_v \left| \frac{v - v_o}{v_o} \right| \text{ subject to} \\ Sv &= 0 \\ v_l \frac{\mu^*}{\mu_o} &\leq v \leq v_u \frac{\mu^*}{\mu_o} \\ v_b &= \mu^*. \end{aligned} \tag{C.16}$$

Second, invert f_l and f_u to calculate a set of enzyme expression consistent with v^* ,

$$n_m = \max(f_l^{-1}(v^*), f_u^{-1}(v^*)). \tag{C.17}$$

Algorithm C.9. Metabolism FBA simulation. See Mathematical Model section above for definition of the mathematical notation.

Calculate flux bounds

begin

```

Initialize bounds:  $v_{l,i} \leftarrow -\infty$ ,  $v_{u,i} \leftarrow +\infty$ 
foreach thermodynamically irreversible reaction  $i$  do
    | Constrain reverse flux to zero:  $v_{l,i} \leftarrow 0$ 
foreach chemically catalyzed reaction  $i$  do
    |  $j \leftarrow$  index of enzyme which catalyzes reaction  $i$ 
    | if  $k_{-,i}$  is known then
        | | bound flux by enzyme kinetics and expression:  $v_{l,i} \leftarrow \max(v_{l,i}, k_{-cat,i} n_{e,j})$ 
    | else bound flux by enzyme expression:  $v_{l,i} \leftarrow v_{l,i} \cdot (n_{e,j} > 0)$ 
    | if  $k_{+,i}$  is known then
        | | bound flux by enzyme kinetics and expression:  $v_{u,i} \leftarrow \min(v_{u,i}, k_{+cat,i} n_{e,j})$ 
    | else bound flux by enzyme expression:  $v_{u,i} \leftarrow v_{u,i} \cdot (n_{e,j} > 0)$ 
foreach chemical reaction  $i$  do
    | foreach protein substrate  $j$  of reaction  $i$  do
        | | if protein substrate  $j$  is not expressed then constrain flux to zero:  $v_{l,i} \leftarrow 0$ ,  $v_{u,i} \leftarrow 0$ 
foreach internal exchange reaction  $i$  do
    | |  $j \leftarrow$  index of metabolite exchanged by reaction  $i$ 
    | | Bound internal metabolite exchange by copy number:  $v_{l,i} \leftarrow \max(v_{l,i}, -n_{m,j})$ 
foreach external exchange reaction  $i$  do
    | |  $j \leftarrow$  index of metabolite exchanged by reaction  $i$ 
    | | Bound external metabolite exchange by copy number and maximum exchange rate:
        | |  $v_{l,i} \leftarrow \max(v_{l,i}, m k_{-ex,i}), v_{u,i} \leftarrow \min(v_{u,i}, m k_{+ex,i}, n_{m,j})$ 

```

Calculate the growth rate and flux of each reaction

begin

```

 $v^* \leftarrow \underset{v}{\operatorname{argmax}} \mu = v_b$  subject to  $Sv = 0$  and  $v_l \leq v \leq v_u$ 
 $\mu^* \leftarrow v_b^*$ 

```

Update the copy number of each metabolite species

begin

```

foreach extracellular metabolite  $i$  do
    | |  $j \leftarrow$  index of reaction which exchanges metabolite  $i$ 
    | |  $n_{m,i} \leftarrow n_{m,i} + v_j^*$ 
foreach intracellular metabolite  $i$  do
    | |  $j \leftarrow$  index of reaction which exchanges/recycle metabolite  $i$ 
    | |  $n_{m,i} \leftarrow n_{m,i} + v_j^*$ 
foreach metabolic objective component  $i$  do
    | |  $n_{m,i} \leftarrow n_{m,i} - S_{i,b} \mu^*$ 

```

Turnover extra energy produced beyond the demands of the 27 other modeled cellular processes,

$\Delta GAM = GAM - \text{used energy}$

```

 $n_{m,ATP} \leftarrow n_{m,ATP} - \Delta GAM \mu^*$ 
 $n_{m,ADP} \leftarrow n_{m,ADP} + \Delta GAM \mu^*$ 
 $n_{m,Pi} \leftarrow n_{m,Pi} + \Delta GAM \mu^*$ 
 $n_{m,H_2O} \leftarrow n_{m,H_2O} - \Delta GAM \mu^*$ 
 $n_{m,H^+} \leftarrow n_{m,H^+} + \Delta GAM \mu^*$ 

```

The upper and lower bound of the expression of each enzyme consistent with the target growth rate was calculated using a similar procedure.

C.3.11 Protein Activation

Biology

The activity of mature protein monomers and complexes is not fixed, but rather can be modulated by small molecules, DNA, RNA, and other proteins, as well as by temperature and pH. Furthermore, cells often purposefully modulate protein activity to respond to external signals and maintain homeostasis.

Reconstruction

Protein chemical regulation was reconstructed based on extensive review of the primary literature and several databases, with particular emphasis on antibiotics (see Table S3Q and S3AQ). The reconstructed protein chemical regulation network contains 16 metabolites, three Boolean-valued pseudometabolites or stimuli, and temperature which regulate 10 proteins including four transcription factors, topoisomerases II and IV, the 30S and 50S ribosomal particles. The 10 putative chemically regulated proteins are critical for transcription, supercoiling, and translation. The effects of physical properties such as temperature and pH on protein activity were curated from BREND⁵⁹, BioCyc¹⁸⁸, and UniProt⁷¹ (see Table S3AO and S3AP). The prosthetic groups and coenzymes required for maturation and catalysis were also curated (see Table S3AM).

Computational Representation

Mathematical Model

Because the exact effects of small molecules, temperature, and pH on the functional activity of proteins are not well characterized, this process implements a Boolean model of their effects on the functional state – enzymatically competent or incompetent – of mature proteins. Less well characterized effects of other physical properties such as pH are not modeled. The functional state, I_i , of every copy of protein species i is governed by a single independent Boolean function, f_i (see

Table S3Q),

$$I_i = f_i \left(\frac{\vec{m}}{V}, \vec{s}, T \right), \quad (\text{C.18})$$

where m_j is the concentration of metabolite j , s_j is the value of stimulus j , and T is the temperature. Proteins with no known chemical regulation were assumed to be constitutively competent.

The **Protein Folding** process separately models the role of chaperones in protein folding and accounts for the small molecule prosthetic groups required for protein folding. Several processes including **Metabolism** model the coenzymes required for catalytic activity.

Integration

The **Stimulus** state represents the value of each pseudometabolite and temperature. The **Metabolite** state represents the copy number of each metabolite. The **Geometry** state represents the cell volume. The **Protein Monomer** and **Protein Complex** states represent the competent and incompetent copy numbers of each protein species. Proteins regulated by this process are synthesized and matured by several processes (see Section C.2.10).

Initial Conditions

After the temperature, cell volume, pseudometabolite values, and metabolite and total protein copy numbers are initialized, the competency state of each of the regulated proteins is initialized to the steady state of the regulatory network. Because the regulatory network is acyclic, the network converges in one iteration.

Dynamic Computation

Algorithm C.10 outlines the implementation of the protein activation model.

Algorithm C.10. Protein activation simulation.

```

Input:  $p_i^c$  competent copy number of protein  $i$ 
Input:  $p_i^i$  incompetent copy number of protein  $i$ 
foreach regulated protein  $i$  do
    if regulatory rule  $f_i(\vec{m}/V, \vec{s}, T)$  then
        | Mark all copies of protein  $i$  competent
        |  $p_i^c = p_i^c + p_i^i$ 
        |  $p_i^i = 0$ 
    else
        | Mark all copies of protein  $i$  incompetent
        |  $p_i^i = p_i^i + p_i^c$ 
        |  $p_i^c = 0$ 

```

C.3.12 Protein Decay

Biology

Protein degradation serves two critically important functions. First, protein degradation eliminates abnormal and potentially toxic proteins including prematurely aborted polypeptides tagged with SsrA degradation signals, and salvages their valuable metabolic resources to support protein synthesis. Second, degradation controls protein expression, enabling cells to direct valuable amino acids away from ineffective or even harmful proteins toward productive proteins, regulate protein activity, and respond to external signals. Tobias et al. have shown that the degradation rates of *E. coli* proteins are correlated with their N-terminal residue, suggesting that cells target proteins for degradation by manipulating their N-termini⁴⁰⁴. This relationship is known as the N-end rule.

Protein refolding is an efficient mechanism for eliminating abnormally folded proteins, requiring less energy than protein degradation and avoiding the large cost of resynthesis. The *M. genitalium* chromosome contains one protein chaperone – the cytosol-localized and ATP-dependent protease ClpB (MG355) – dedicated to protein refolding^{223,304}. The kinetics and energetics of protein refolding are not well characterized.

This process models the degradation of protein monomers, macromolecular complexes, cleaved signal sequences, and prematurely aborted polypeptides as well as the misfolding and refolding of protein monomers and complexes.

Reconstruction

Protein Half-Lives

The half-life of each protein monomer was predicted using the N-end rule reported by Tobias et al.⁴⁰⁴. The half-life of each macromolecular complex was set equal to the weighted mean half-life of its RNA and protein subunits. The half-lives of the damaged and misfolded configurations of each protein species, cleaved signal sequences, and prematurely aborted polypeptides were set to zero to reflect their rapid degradation. Section C.2.12 and Table S3Y describe how RNA half-lives were reconstructed.

Proteolytic Machinery

The *M. genitalium* genome contains homologs of proteases FtsH (MG457) and La (MG239) and six peptidases: aminopeptidase (MG324), cytosol aminopeptidase (MG391), glycoprotease (MG208), metalloendopeptidase (MG046), oligoendopeptidase F (MG183), and proline iminopeptidase (MG020)^{304,380}. The membrane-localized protease FtsH is believed to cleave prematurely aborted polypeptides tagged with an N-terminal SsrA tag into approximately 15 amino long fragments by an ATP-dependent mechanism^{161,380}. Bruckner et al. reported the kinetics and energetics of FtsH proteolysis⁴⁴. The cytosol-localized protease La is believed to processively cleave most other proteins into fragments 10-20 amino acids in length by an ATP-dependent mechanism²¹². Lee and Suzuki reported the kinetics and energetics of La proteolysis²¹². Peptidases are believed to hydrolytically degrade the polypeptide fragments produced by peptidases FtsH and La. The specific function of each of the six *M. genitalium* peptidases is unknown.

Computational Representation

Mathematical Model

Under conditions of excess proteases, peptidases, and ATP, this process models the configuration of each protein as a five-state system – aborted, nascent, mature, misfolded, and damaged – and models the degradation of protein i as Poisson process with rate parameter $k_{d,i} = \ln 2/\tau_{1/2,i}$ where $\tau_{1/2,i} = 20\text{ h}$ for the nascent and mature states and zero for the aborted, misfolded, and damaged states. Under conditions of limited proteases, peptidases, or ATP the model stochastically degrades

proteins as a function of the copy number and catalytic rate of each enzyme and the number of cytosolic ATP molecules. This model assumes (1) cytosol-localized nascent, mature, misfolded, and damaged proteins are degraded by protease La and six peptidases, (2) membrane and extracellular-localized proteins are not vulnerable to degradation by the cytosol localized protease La, (3) aborted polypeptides are degraded by protease FtsH and six peptidases, (4) protein degradation is kinetically fast and therefore this model doesn't represent intermediate degradation states, and (5) aborted, misfolded, and damaged proteins are immediately degraded if proteases are expressed and energy is available. Algorithm C.11 outlines the implementation of the protein degradation model.

This process models protein unfolding as a Poisson process with rate parameter k_u set to the nominal rate $10^{-6} \text{ s}^{-1} \text{ molecule}^{-1}$. Cytosol-localized protein refolding is modeled as a single chemical event catalyzed by the cytosol-localized chaperone ClpB and driven by the hydrolysis of a single ATP molecule. Because protein refolding is not well characterized, this process assumes that if ClpB is expressed and excess ATP is available, protein refolding proceeds to completion within the 1 s simulation time step. If ClpB is expressed, but ATP is not present in excess, the model stochastically refolds proteins equal to the intracellular copy number of ATP.

This model makes several simplifying assumptions. First, the model assumes that all protein species misfold and refold at the same rate. Second, the model assumes that the kinetics of protein misfolding and refolding are fast and therefore intermediate misfolded states can be ignored.

Integration

The **Protein Monomer** and **Protein Complex** states represent the copy number of each protein monomer and macromolecular complex including the signal sequence, misfolded, and damaged copy numbers of each protein species and the copy number of each proteolytic enzyme in each of five compartments: cytosol, membrane, terminal organelle cytosol, terminal organelle membrane, and extracellular space. The **Polypeptide** state represents the amino acid sequence of each prematurely aborted polypeptide. The **Chromosome**, **FtsZ Ring**, **Ribosome**, and **RNA Polymerase** states represents the detailed configurations of DNA-bound proteins, the FtsZ septal ring, RNA polymerases, and ribosomes. These detailed configurations are updated when DNA-bound proteins, FtsZ, RNA polymerase, or 70S ribosomes are degraded to reflect decreased enzyme copy numbers. The **Transcript** and **Polypeptide** states are also updated upon RNA polymerase or 70S ribosome degradation to reflect transcript and polypeptide abortion. The **Rna** state represents the damaged copy

number of each RNA species. See below for further discussion. The **Metabolite** state represents the copy number of each intracellular metabolite.

This process marks RNA subunits of degraded macromolecules as “damaged” for immediate degradation by the **RNA Decay** process.

Initial Conditions

The misfolded and damaged configurations of each RNA and protein species are initialized with zero copy number because the typical occupancies these configurations are small compared to that of the mature configuration. Similarly, the cell is initialized with no prematurely aborted polypeptides. The **Rna**, **Protein Monomer**, and **Protein Complex** states initialize the total copy numbers of each RNA and protein species.

Dynamic Computation

Algorithm C.11 outlines the implementation of the protein decay model.

Fitting

The expression of the proteases and peptidases was fit to provide sufficient enzymes to quickly degrade damaged proteins, or more specifically to prevent sustained accumulation of damaged proteins. See Section C.1.3 for further discussion.

Algorithm C.11. Protein decay simulation.

Input: $k_u = 10^{-6}$ s protein unfolding rate
Input: $k_{d,i}^p = \ln 2/\tau_{1/2,i}$ degradation rate of protein monomer species i
Input: $k_{d,i}^c = \ln 2/\tau_{1/2,i}$ degradation rate of macromolecular complex species i
Input: $k_{\text{Lon}}, k_{\text{FtsH}}$ protease kinetic rates in amino acids per second
Input: m_i copy number of metabolite i
Input: r_i^d damaged copy number of RNA species i
Input: p_i^a, p_i^u, p_i^d mature, unfolded, and damaged copy numbers of protein monomer species i
Input: c_i^a, c_i^u, c_i^d mature, unfolded, and damaged copy numbers of macromolecular complex species i
Input: $p \leftarrow \{p_i^a, p_i^u, p_i^d\}$ copy number of all forms of protein monomer species i
Input: $c \leftarrow \{c_i^a, c_i^u, c_i^d\}$ copy number of all forms of macromolecular complex species i
Input: $e_{\text{Lon}}, e_{\text{FtsH}}$ copy numbers of proteases La and FtsH
Input: s_i sequence of prematurely aborted polypeptide i
Input: l_i^p length of protein monomer species i
Input: l_i^s length of prematurely aborted polypeptide i
Input: P_{ij}^m reaction stoichiometry of metabolite i in the degradation of protein monomer j including ATP hydrolysis
Input: C_{ij}^m stoichiometry of metabolite species i in macromolecular complex j
Input: C_{ij}^r subunit stoichiometry of mature RNA species i in macromolecular complex j
Input: C_{ij}^p subunit stoichiometry of mature protein monomer i in macromolecular complex j
Input: S_{ij}^m reaction stoichiometry of metabolite i in the degradation of polypeptide j including ATP hydrolysis
Input: $\Delta t = 1$ s is the simulation time step

Misfold proteins (see Algorithm C.13)
Refold proteins (see Algorithm C.12)
Degrade macromolecular complexes (see Algorithm C.14)
Degrade prematurely aborted polypeptides (see Algorithm C.15)
Degrade protein monomers (see Algorithm C.16)

Algorithm C.12. Protein refolding simulation. See Algorithm C.11 for mathematical notation.

```

if ClpB chaperone is expressed ( $e_{\text{ClpB}} > 0$ ) then
    repeat
        Select protein species  $i \sim \text{multinomialRand}(1, \{p_i^u, c_i^u\}) / (\sum_j p_j^u + \sum_j c_j^u)$ 
        Update protein copy numbers
        if species  $i$  is a protein monomer then  $p_i^u \leftarrow p_i^u - 1, p_i^a \leftarrow p_i^a + 1$ 
        else  $c_i^u \leftarrow c_i^u - 1, c_i^a \leftarrow c_i^a + 1$ 
        Hydrolyze ATP
         $m_{\text{ATP}} \leftarrow m_{\text{ATP}} - 1$ 
         $m_{\text{ADP}} \leftarrow m_{\text{ADP}} + 1$ 
         $m_{\text{Pi}} \leftarrow m_{\text{Pi}} + 1$ 
         $m_{\text{H}_2\text{O}} \leftarrow m_{\text{H}_2\text{O}} - 1$ 
         $m_{\text{H}^+} \leftarrow m_{\text{H}^+} + 1$ 
    until ( $p_i^u = 0 \forall i$  and  $c_i^u = 0 \forall i$ ) or  $m_{\text{ATP}} = 0$ 

```

Algorithm C.13. Protein unfolding simulation. See Algorithm C.11 for mathematical notation.

```

foreach protein monomer  $i$  do
    Calculate protein unfolding extent:  $\Delta p_i^u \leftarrow \text{poissonRand}(k_u p_i^a)$ 
    Update protein copy numbers:  $p_i^a \leftarrow p_i^a - \Delta p_i^u, p_i^u \leftarrow p_i^u + \Delta p_i^u$ 

foreach macromolecular complex  $i$  do
    Calculate protein unfolding extent:  $\Delta c_i^u \leftarrow \text{poissonRand}(k_u c_i^a)$ 
    Update protein copy numbers:  $c_i^a \leftarrow c_i^a - \Delta c_i^u, c_i^u \leftarrow c_i^u + \Delta c_i^u$ 

```

Algorithm C.14. Macromolecular complex degradation simulation. See Algorithm C.11 for mathematical notation.

```

Calculate rates of protein degradation
foreach macromolecular complex  $i$  do
     $\Delta c_i \leftarrow \text{poissonRand}(k_{d,i}^c c_i)$ 

Degrade specific complexes
repeat
    Select protein species  $i \sim \text{multinomialRand}(\Delta c_i / \sum_j \Delta c_j)$ 
    if sufficient metabolic resources available ( $m_j \geq -C_{ji}^m \forall j$ ) then
        Breakdown complex into metabolites and damaged RNA and protein subunits
         $\Delta c_i \leftarrow \Delta c_i - 1$ 
         $c_i \leftarrow c_i - 1$ 
         $p^d \leftarrow p_i^d + C_{\bullet i}^p$ 
         $r^d \leftarrow r_i^d + C_{\bullet i}^r$ 
         $m \leftarrow m C_{\bullet i}^m$ 
    else
         $\Delta c_i \leftarrow 0$ 
    until  $\Delta c_i = 0 \forall i$ 

```

Algorithm C.15. Aborted polypeptide degradation simulation. See Algorithm C.11 for mathematical notation.

```

Let  $z \leftarrow \text{poissonRand}(e_{FtsH}/k_{FtsH}\Delta t)$  be the FtsH enzymatic capacity
while all peptidases expressed ( $e_i \forall \text{peptidases}_i$ ) do
    Select an aborted polypeptides  $i \sim \text{multinomialRand}(1, 1/n_{pep})$ 
    if insufficient resources to degrade polypeptide ( $\exists j. s.t -S_{ji}^m > m_j \text{ and } z \geq \text{length}(l_i^s) \geq 1$ ) then
        break
    Degrade polypeptide:  $s_i \leftarrow \emptyset, m \leftarrow m + S_{\bullet i}^m, z \leftarrow z - \text{length}(l_i^s)$ 

```

Algorithm C.16. Protein monomer degradation simulation. See Algorithm C.11 for mathematical notation.

```

if all peptidases expressed ( $e_i \forall$  peptidases $_i$ ) then
    Let  $z \leftarrow \text{poissonRand}(e_{\text{Lon}}/k_{\text{Lon}}\Delta t)$  be the Lon enzymatic capacity
    Calculate rates of protein degradation
    foreach protein monomer  $i$  do
         $\Delta p_i \leftarrow \text{poissonRand}(k_{d,i}^p p_i)$ 
    Degrade specific monomers
    repeat
        Select protein species  $i \sim \text{multinomialRand}(\Delta p_i / \sum_j \Delta p_j)$ 
        if sufficient metabolic resources available ( $m_j \geq -P_{ji}^m \forall j$  and  $z \geq l_i^p \geq 1$ ) then
            Breakdown protein into metabolites
             $\Delta p_i \leftarrow \Delta p_i - 1$ 
             $p_i \leftarrow p_i - 1$ 
             $m \leftarrow m P_{\bullet i}^m$ 
             $z \leftarrow z - l_i^p$ 
        else
             $\Delta p_i \leftarrow 0$ 
        until  $\Delta p_i = 0 \forall i$ 

```

C.3.13 Protein Folding

Biology

Proteins are synthesized as long, catalytically inactive linear chains of amino acids (see Translation). Subsequently, proteins fold into energetically favorable, compact, and enzymatically competent three-dimensional structures. While some protein species are believed to fold spontaneously, other proteins are believed to require helper chaperone proteins to properly fold^{194,280}. In addition, some protein species require metal ions and other small molecule prosthetic groups to fold. This process models chaperone-mediated protein folding.

M. genitalium is believed to employ three chaperones to assist protein folding. First, trigger factor (Tig, MG238) co-translationally binds all nascent polypeptides at the ribosome exit site (L23) and assists in early protein folding^{95,181,194,223,280,291,321}.

Second, chaperones GroEL (MG392) and DnaK (MG305) and their co-chaperones are believed to assist in late folding. GroEL and DnaK are believed to fold 10-15% and 5-18% of all proteins, respectively. GroEL and its co-chaperone GroES (MG393) are believed to help fold intermediate sized proteins (20-60 kDa)^{223,291}. GroEL is believed to bind each protein for 30 s to 10 min, and to couple folding to ATP hydrolysis^{223,291}.

DnaK and its co-chaperones DnaJ (MG019) and GrpE (MG201) are believed to help fold large proteins ($> 30 \text{ kDa}$)^{223,291}. DnaK is a monomeric protein which transiently ($< 2 \text{ min}$) binds to the backbones of short, linear, unfolded peptide segments. GrpE is believed to couple peptide release to ATP hydrolysis. DnaJ is believed to regulate the activity of DnaK, as well bind the side chains of hydrophobic and aromatic residues and directly assist protein folding.

Finally, FtsH and the lipids phosphatidyl ethanolamine and phosphatidyl glycerol have been suggested to assist membrane protein folding²²³. However, the role of FtsH as a molecular chaperone is not well established, and FtsH has been associated with several additional functions. Consequently, we chose not to model FtsH as a chaperone. The contributions of phosphatidyl ethanolamine and phosphatidyl glycerol to protein folding are also poorly understood, and are not modeled.

Reconstruction

M. genitalium GroEL substrates were reconstructed based on two proteome-scale studies of *E. coli* and *B. subtilis* (see Table S3AG)^{113,187}. *M. genitalium* DnaK substrates were reconstructed based on a proteome-scale study of *E. coli* (see Table S3AG)⁹⁵. The prosthetic group requirements for protein folding were reconstructed based on an extensive review of the literature and several databases (see Table S3AM). Table S3M lists the chaperones required to fold each protein monomer and the prosthetic group stoichiometry of each protein monomer.

Computational Representation

Mathematical Model

This process implements a model of the chaperone-mediated folding of processed, translocated polypeptides. Because the kinetics and energetics of protein folding are not well understood, this process represents the three-dimensional configuration of each protein as a two-state – folded, unfolded – Boolean variable. Furthermore, this process makes the simplifying assumption that the folding rate, r_i , of protein species i is a Boolean function of the copy numbers of metabolites and chaperones,

$$r_i = \min \left(\overbrace{p_i^u}^{\text{protein}}, \overbrace{\left[\min_j \frac{m_j}{M_{ji}} \right]}^{\text{metabolites}}, \overbrace{\min_j \frac{e_j}{C_{ji}}}^{\text{enzymes}} \right), \quad (\text{C.19})$$

where p_i^u is the unfolded copy number of protein species i , m_i is the copy number of metabolite i , e_i is the copy number of chaperone i , M_{ji} is the stoichiometry of prosthetic group j in protein i , and C_{ji} is true if protein species i requires chaperone j to fold. p_i^f is the folded copy number of protein species i .

Integration

The **Metabolite** state represents the copy number of each metabolite species. The **Protein Monomer** and **Protein Complex** states represent the copy numbers of unfolded and folded proteins. Protein folding is one of the last steps in protein maturation following protein processing and translocation (see **Protein Processing I** and **Protein Translocation** processes) and preceding protein modification (see **Protein Modification** process). See Section C.2.10 for further discussion.

Initial Conditions

The **Protein Monomer** and **Protein Complex** states initialize the total copy number of each protein species, and set all protein monomers and complexes to their mature – folded and modified – configuration.

Dynamic Computation

Algorithm C.17 outlines the implementation of the protein folding model.

Algorithm C.17. Protein folding simulation.

```

repeat
  foreach protein species  $i$  do
    Calculate the relative Boolean-valued folding rate,  $r_i$ , of each protein species
     $r_i \leftarrow \min \{p_i^u, m_j/M_{ij}, c_k \geq C_{ik}\}$ 
    Select a protein species  $i$  to fold according to multinomialRand(1,  $\vec{r}$ )
    Increment the folded copy number of protein species  $i$ ,  $p_i^f \leftarrow p_i^f + 1$ 
    Decrement the unfolded copy number of protein species  $i$ ,  $p_i^u \leftarrow p_i^u - 1$ 
    Decrement the copy numbers of the prosthetic groups of protein species  $i$ ,  $m_j \leftarrow m_j - M_{ij}$ 
  until no additional proteins can fold ( $\vec{r} = 0$ )

```

Dynamic Computation

Chaperone expression was fit to provide sufficient enzymes to quickly fold all newly synthesized proteins, or more precisely, to prevent accumulation of unfolded proteins. See Section C.1.3 for further discussion.

C.3.14 Protein Modification

Biology

Post-translational protein modification serves several important functions^{355,423}. First, post-translation modification can increase the structural and chemical diversity of the proteome by stabilizing alternative conformations and providing catalytic cofactors. Second, post-translational modification, and in particular phosphorylation, provides a mechanism to regulate protein activity. Third, post-translational modification can be used to regulate protein expression through proteasome recruitment. This process models protein covalent modification including phosphorylation, lipoyl transfer, and α -glutamate ligation.

Reconstruction

The *M. genitalium* protein modification complement was reconstructed in three steps. First, protein modifications that have been observed in *M. genitalium*³⁸⁶, *M. pneumoniae*^{93,163,200,231,386}, or other related bacteria^{71,145,152,164,298} or which have been computationally predicted^{96,214,252} were curated. Second, curated protein modifications were mapped to *M. genitalium* homologs using sequence alignment. Third, based on the genome annotation³⁰⁴, *M. genitalium* was determined to have three protein modification pathways: (1) phosphorylation catalyzed by serine/threonine kinase PrkC (MG109), (2) lipoyl transfer to lysine catalyzed by LplA (MG270), and (3) C-terminal α -glutamate ligation catalyzed by RimK (MG012). Fourth, modifications were rejected which do not belong to one of these three pathways, or for which a specific locus was not reported. Table S3AH outlines the reconstruction process and lists the reconstructed protein modifications. The reconstructed protein modification network contains one kinase that modifies 16 proteins, one lipoyl transferase that modifies the E2 subunit of pyruvate dehydrogenase (PdhC, MG272), providing an important organosulfur cofactor which contains a catalytic disulfide bond, and one α -glutamate ligase that modifies 50S ribosomal protein L6 (RplF, MG166). The stoichiometry and kinetics of all three modeled protein modification reactions were based on a review of the primary literature^{61,117,137,241,381} (see Table S3O). Catalytic disulfide bonds were separately reconstructed and are modeled by several chemical reactions in the Metabolism process (see Table S3O).

Computational Representation

Mathematical Model

This process models protein covalent modification, the fifth step of post-translational processing. In particular, this process models protein phosphorylation, lipoyl transfer, and α -glutamate ligation. Because the mechanisms of protein modification are not well characterized on the genomic scale, this process make several simplifying assumptions. First, this process assumes that each protein is fully modified in a single time step, collapses the modification of each protein into a single reaction, and only represents unmodified and fully modified proteins. Intermediate protein configurations are not represented. Second, this process assumes that the mean arrival rate, v_i of modification events of each protein species i is independently limited by (1) the copy number of unmodified protein, p_i^u , (2) the copy numbers of intracellular metabolites, m_j , and (3) the copy numbers of the protein modification enzymes, e_j . Based on these assumptions, the functional form of v_i is given by

$$v_i = \min \left(\overbrace{p_i^u}^{\text{Protein}}, \overbrace{\left[\min_j \frac{m_j}{M_{ji}^s} \right]}^{\text{metabolites}}, \overbrace{\text{poissonRand} \left(\min_j \frac{e_j}{K_{ji}} \Delta t \right)}^{\text{enzymes}} \right), \quad (\text{C.20})$$

where M_{ji} is the stoichiometry of metabolite j in the modification of protein species i , $M^s = \max(0, -M)$ is the negative part of M , K_{ji} is the experimentally observed catalytic rate of enzyme j in the modification of protein species i , and $\Delta t = 1\text{s}$ is the simulation time step.

This process implements a stochastic model of the arrival of protein monomer and macromolecular complex modification events with relative rates v_i . Until protein, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, v_i , of each modification event, (2) selects a single modification event to execute according to a multinomial distribution parameterized by v_i , and (3) executes the selected modification reaction, updating the copy numbers of proteins and metabolites and decrementing the available enzymatic capacity. Algorithm C.18 outlines the implementation of the protein modification model.

Integration

The **Protein Monomer** and **Protein Complex** states represent the copy number of each protein modification enzyme. The **Protein Monomer** state also represents the unmodified and modified copy numbers of each protein monomers species. The **Metabolite** state represents the copy number of each intracellular metabolite.

Proteins are synthesized and matured in six steps (see Section C.2.10). This process models the fifth step in protein synthesis following protein folding (see **Protein Folding** process) and preceding macromolecule assembly (see **Macromolecular Complexation**, **Ribosome Assembly**, **Protein Folding**, and **Protein Modification** processes).

Initial Conditions

Section C.1.4 outlines the cell state initialization algorithm. Briefly, after the **Mass** state initializes the total cell mass, the **Protein Monomer** state initializes the total copy number of each protein monomer species and initializes all monomers to their mature – processed, folded, modified, and localized – configuration. Second, the **Macromolecular Complexation** and **Ribosome Assembly** processes initialize the copy number of each ribonucleoprotein complex, and set all initialized complexes to their mature – folded and modified – configuration. Finally, several processes including **Transcription** and **Translation** initialize the detailed configurations of RNA polymerases, 70S ribosomes, DNA-binding proteins, and FtsZ.

The unmodified configurations of each protein monomer species is initialized with zero copy number because the typical occupancy this configuration is small compared to that of the modified/mature configuration.

Dynamic Computation

Algorithm C.18 outlines the implementation of the protein modification model.

Fitting

The expression of the protein modification enzymes was fit to provide sufficient enzymes to quickly modify newly synthesized proteins, or more specifically to prevent sustained accumulation of unmodified proteins. See Section C.1.3 for further discussion.

Algorithm C.18. Protein modification simulation. See Mathematical Model section above for definition of the mathematical notation.

Input: p_i^m copy number of modified protein species i

Let $k_i \leftarrow e_i \Delta t$ be the capacity of enzyme i for protein modification

repeat

 Calculate modification rates

foreach protein species i **do**

 Calculate v_i according to Eq. C.20

 Select protein species $i \sim \text{multinomialRand}(1, v_i / \sum_j v_j)$

 Update protein copy numbers: $p_i^u \leftarrow p_i^u - 1, p_i^m \leftarrow p_i^m + 1$

 Update metabolites: $m \leftarrow m - M_{\bullet i}$

 Update enzyme catalytic capacity: $k \leftarrow k - K_{\bullet i}$

until no further modification possible ($v_i = 0 \forall i$)

C.3.15 Protein Processing I

Biology

Bacterial translation is initiated by the formation of the 70S ribosome and the recruitment of the initiator tRNA^{fMet} to the ribosomal P site. This process models N-terminal formylmethionine deformylation and N-terminal methionine cleavage, the first steps in post-translational processing.

The exact function of the initiator tRNA^{fMet} is unknown. Several authors suggest that bacteria employ a separate tRNA for translation initiation in order to control the rate of protein synthesis through the expression of initiator tRNA^{105,139}. Others believe that cells use a separate initiator tRNA to differentiate between initiation factor 2-dependent recruitment of the first tRNA to the ribosomal P site and EF-Tu-dependent recruitment of subsequent tRNA to the A site^{105,240}. The unique formyl group of tRNA^{fMet} has been suggested to enable initiation factor 2 to discriminate between initiator and other tRNA¹⁰⁵. The role of formylmethionine as the starting amino acid is also not well understood^{240,247}. Some authors believe that bacteria employ methionine as the first amino acid because it is the most expensive to synthesize and therefore couples translation to general cell health²⁴⁷. Furthermore, neither the formyl group of tRNA^{fMet} nor methionine is essential for translation initiation^{240,247}. Several studies have shown that other amino acids are able to support translation initiation^{240,247}.

Following translation, bacteria deformylate the N-terminal methionine of most proteins and cleave the N-terminal methionine of approximately 7% of proteins. The function of peptide deformylation

and methionine cleavage is not well understood. Some authors believe that bacteria employ these reactions to recycle formylmethionine which is a metabolically expensive amino acid²⁴⁷. Other authors, citing the N-end rule⁴⁰⁴, believe that bacteria cleave the N-terminal methionine of specific proteins in order to regulate protein half-lives by exposing the second-most N-terminal amino acid²⁴⁷.

Reconstruction

M. genitalium peptide deformylase (MG106) was assumed to deformylate all protein monomers. The protein substrates of *M. genitalium* methionine aminopeptidase (MG172) were reconstructed based on specific N-terminal methionine cleavages of *M. genitalium* homologs observed in *Shewanella oneidensis* MR-1¹⁴⁵. The kinetics of peptide deformylation and N-terminal methionine cleavage were reconstructed from the primary literature^{144,248}. Table S3O lists the stoichiometry and kinetics of the deformylation and cleavage reactions.

Computational Representation

Mathematical Model

This process models N-terminal formylmethionine deformylation and N-terminal methionine cleavage, the first step of post-translational processing. Because the mechanisms of early protein processing are not well characterized on the genomic scale, this process make several simplifying assumptions. First, this process assumes that each protein is both deformylated and cleaved in a single time step. Consequently, this process collapses the processing of each protein into a single reaction, and only represents nascent and fully processed proteins. Intermediate protein configurations are not represented. Second, this process assumes that the mean arrival rate, v_i of processing events of each protein species i is independently limited by (1) the copy number of unprocessed protein, p_i^u , (2) the copy numbers of intracellular metabolites, m_j , and (3) the copy numbers of processing enzymes, e_j . Based on these assumptions, the functional form of v_i is given by

$$v_i = \min \left(\overbrace{p_i^u}^{\text{Protein}}, \overbrace{\left[\min_j \frac{m_j}{M_{ji}^s} \right]}^{\text{metabolites}}, \overbrace{\text{poissonRand} \left(\min_j \frac{e_j}{K_{ji}} \Delta t \right)}^{\text{enzymes}} \right), \quad (\text{C.21})$$

where M_{ji} is the stoichiometry of metabolite j in the processing of protein species i , $M^s = \max(0, -M)$ is the negative part of M , K_{ji} is the experimentally observed catalytic rate of enzyme j in the processing of protein species i , and $\Delta t = 1\text{ s}$ is the simulation time step.

The **Protein Processing I** process implements a stochastic model of the arrival of early protein monomer processing events with relative rates v_i . Until protein, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, v_i , of each processing event, (2) selects a single processing event to execute according to a multinomial distribution parameterized by v_i , and (3) executes the selected processing reaction, updating the copy numbers of protein monomers and metabolites and decrementing the available enzymatic capacity. Algorithm C.19 outlines the implementation of the early protein processing model.

Integration

The **Protein Monomer** and **Protein Complex** states represent the copy number of each protein processing enzyme. The **Protein Monomer** state also represents the nascent and processed copy numbers of each protein monomers species. The **Metabolite** state represents the copy number of each intracellular metabolite. Proteins are synthesized and matured in six steps (see Section C.2.10). This process models the first step in post-translational processing following translation (see **Translation** process) and preceding membrane and extracellular protein translocation and cytosolic protein folding (see **Protein Translocation** and **Protein Folding** processes).

Initial Conditions

Section C.1.4 outlines the cell state initialization algorithm. Briefly, after the **Mass** state initializes the total cell mass, the **Protein Monomer** state initializes the total copy number of each protein monomer species and initializes all monomers to their mature – processed, folded, modified, and localized – configuration. Second, the **Macromolecular Complexation** and **Ribosome Assembly** processes initialize the copy number of each ribonucleoprotein complex and set all initialized complexes to their mature – folded and modified – configuration. Finally, several processes including **Transcription** and **Translation** initialize the detailed configurations of RNA polymerases, 70S ribosomes, DNA-binding proteins, and FtsZ.

The nascent and processed configurations of each protein monomer species are initialized with zero copy number because the typical occupancy these configurations is small compared to that of the

mature configuration.

Dynamic Computation

Algorithm C.19 outlines the implementation of the protein processing (I) model.

Algorithm C.19. Protein processing (I) simulation. See Mathematical Model section above for definition of the mathematical notation.

Input: p_i^p copy number of processing protein monomer species i
 Let $k_i \leftarrow e_i \Delta t$ be the capacity of enzyme i for protein processing
repeat
 Calculate processing rates
 foreach protein monomer species i **do**
 Calculate v_i according to Eq. C.21
 Select protein monomer species $i \sim \text{multinomialRand}(1, v_i / \sum_j v_j)$
 Update protein monomer copy numbers: $p_i^u \leftarrow p_i^u - 1, p_i^p \leftarrow p_i^p + 1$
 Update metabolites: $m \leftarrow m - M_{\bullet i}$
 Update enzyme catalytic capacity: $k \leftarrow k - K_{\bullet i}$
until no further processing possible ($v_i = 0 \forall i$)

Fitting

The expression of the protein processing enzymes was fit to provide sufficient enzymes to quickly process proteins, or more specifically to prevent sustained accumulation of unprocessed proteins. See Section C.1.3 for further discussion.

C.3.16 Protein Processing II

Biology

This process models the third step of post-translational processing: lipoprotein diacylglycerol addition and lipoprotein and secreted protein signal peptide cleavage.

Lipoproteins

As discussed in Section C.3.17, *M. genitalium* lipoproteins are translated in the cytosol and subsequently targeted to the plasma membrane by type II N-terminal signal sequences. Following membrane insertion, *M. genitalium* lipoproteins are first anchored to the outer leaflet of the plasma membrane. This is achieved via covalent addition of diacylglycerol to the sulfhydryl group of

the lipobox cysteine by diacylglyceryl transferase (MG086)^{56,157,342}. Many bacteria further anchor lipoproteins through phospholipidation. However, *M. genitalium* does not have an apolipoprotein transacylase, and is not believed to phospholipidate lipoproteins^{57,263,311}. Second, lipoprotein N-terminal signal sequences are cleaved immediately C-terminal to the lipobox cysteine by type II signal peptidase (MG210)⁵⁶. See Section C.3.17 for further discussion of the structure of type II signal sequences.

Secreted Proteins

Extracellular *M. genitalium* proteins are transcribed in the cytosol and targeted to the plasma membrane by type II signal sequences. In contrast to lipoproteins, extracellular proteins do not undergo diacylglyceryl transfer, and instead are cleaved immediately C-terminal to the cysteines of their lipoboxes by type II signal peptidase, releasing the resultant protein and signal peptide into the extracellular space.

Integral Membrane Proteins

M. genitalium does not have a type I signal sequence protease³⁸⁰ and does not cleave the signal sequences of integral membrane proteins. Integral membrane protein signal peptides are believed to help anchor proteins to the membrane⁸⁴.

Reconstruction

The localization and N-terminal signal sequence length of each protein monomer was reconstructed based on an extensive review of the primary literature, several proteomic database, and several computational predictions. See Section C.3.17 for further discussion. The stoichiometry and kinetics of each protein processing reaction was reconstructed from the primary literature^{33,56,126,342,354}. Table S3O lists the reconstructed stoichiometry and kinetics of each reaction.

Computational Representation

Mathematical Model

This process models lipoprotein diacylglyceryl adduction and lipoprotein and secreted protein signal peptide cleavage, the third step of post-translational processing. Because the mechanisms of protein

processing are not well characterized on the genomic scale, this process make several simplifying assumptions. First, this process assumes that each protein is both covalently modified and cleaved in a single time step. Consequently, this process collapses the processing of each protein into a single reaction and only represents unprocessed and fully processed proteins. Intermediate protein configurations are not represented. Second, this process assumes that the mean arrival rate, v_i of processing events of each protein species i is independently limited by (1) the copy number of unprocessed protein, p_i^u , (2) the copy numbers of intracellular metabolites, m_j , and (3) the copy numbers of processing enzymes, e_j . Based on these assumptions, the functional form of v_i is given by

$$v_i = \min \left(\underbrace{p_i^u}_{\text{Protein}}, \underbrace{\left[\min_j \frac{m_j}{M_{ji}^s} \right]}_{\text{metabolites}}, \underbrace{\text{poissonRand} \left(\min_j \frac{e_j}{K_{ji}} \Delta t \right)}_{\text{enzymes}} \right), \quad (\text{C.22})$$

where M_{ji} is the stoichiometry of metabolite j in the processing of protein species i , $M^s = \max(0, -M)$ is the negative part of M , K_{ji} is the experimentally observed catalytic rate of enzyme j in the processing of protein species i , and $\Delta t = 1\text{s}$ is the simulation time step.

This process implements a stochastic model of the arrival of protein monomer processing events with relative rates v_i . Until protein, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, v_i , of each processing event, (2) selects a single processing event to execute according to a multinomial distribution parameterized by v_i , and (3) executes the selected processing reaction, updating the copy numbers of protein monomers and metabolites and decrementing the available enzymatic capacity. Algorithm C.20 outlines the implementation of the protein processing model.

Integration

The **Protein Monomer** and **Protein Complex** states represent the copy number of each protein processing enzyme. The **Protein Monomer** state also represents the unprocessed and processed copy numbers of each protein monomers species. The **Metabolite** state represents the copy number of each intracellular metabolite.

Protein are synthesized and matured in six steps (see Section C.2.10). This process models lipoprotein diacylglycerol transfer and lipoprotein and secreted protein signal sequence cleavage following translocation (see **Protein Translocation** process) and preceding folding and modification (see

Protein Folding and Protein Modification processes).

Initial Conditions

Section C.1.4 outlines the cell state initialization algorithm. Briefly, after the `Mass` state initializes the total cell mass, the `Protein Monomer` state initializes the total copy number of each protein monomer species and initializes all monomers to their mature – processed, folded, modified, and localized – configuration. Second, the `Macromolecular Complexation` and `Ribosome Assembly` processes initialize the copy number of each ribonucleoprotein complex, and set all initialized complexes to their mature – folded and modified – configuration. Finally, several processes including `Transcription` and `Translation` initialize the detailed configurations of RNA polymerases, 70S ribosomes, DNA-binding proteins, and FtsZ.

The unprocessed and processed configurations of each protein monomer species are initialized with zero copy number because the typical occupancy these configurations is small compared to that of the mature configuration.

Dynamic Computation

Algorithm C.20 outlines the implementation of the protein processing (II) model.

Algorithm C.20. Protein processing (II) simulation. See Mathematical Model section above for definition of the mathematical notation.

Input: p_i^p copy number of processed protein monomer species i
Input: p_i^s copy number of cleaved signal sequence of protein monomer species i

Let $k_i \leftarrow e_i \Delta t$ be the capacity of enzyme i for protein processing

repeat

- Calculate processing rates
- foreach** protein monomer species i **do**
- Calculate v_i according to Eq. C.22
-
- Select protein monomer species $i \sim \text{multinomialRand}(1, v_i / \sum_j v_j)$
- Update protein monomer copy numbers: $p_i^u \leftarrow p_i^u - 1, p_i^p \leftarrow p_i^p + 1, p_i^s \leftarrow p_i^s + 1$
- Update metabolites: $m \leftarrow m - M_{\bullet i}$
- Update enzyme catalytic capacity: $k \leftarrow k - K_{\bullet i}$

until no further processing possible ($v_i = 0 \forall i$)

Fitting

The expression of the protein processing enzymes was fit to provide sufficient enzymes to quickly process proteins, or more specifically to prevent sustained accumulation of unprocessed proteins. See Section C.1.3 for further discussion.

C.3.17 Protein Translocation

Biology

Bacterial proteins are translated in the cytosol. However, many functionally important proteins including nutrient transporters, ATP synthase, metabolic enzymes, adhesins, receptors, transducers, virulence factors, and the protein translocation machinery itself must be embedded in the cell membrane or secreted in order to properly function³⁶⁸. Cells employ short N-terminal and C-terminal signal sequences to selectively translocate proteins. This process models membrane and extracellular protein localization, the second step in post-translational processing.

Reconstruction

The subcellular localization – cytosol, integral membrane, lipoprotein, terminal organelle cytosol, terminal organelle lipoprotein, or extracellular space – of each protein monomer and the N-terminal signal sequence of each lipoprotein and secreted protein was reconstructed based on an extensive review of the primary literature^{17,18,54,60,196–198,317,324,329,400}, several proteomic databases^{59,65,71,109,128,143,258}, and several computational predictions^{22,23,118,151,154,160,182,282,334,397}. Table S3AC-S3AE describe the reconstruction process in detail and list the reconstructed localization and signal sequence of each protein monomer.

M. genitalium employs two convergent SecA-dependent pathways – co-translational and post-translational – to translocate approximately 35-45% of all protein monomers into the plasma membrane⁸⁴. *M. genitalium* does not contain a type I Tat transporter or sortase²⁹². The co-translational SecA pathway translocates integral membrane proteins into the cell membrane. First, GTP-dependent signal recognition particles (SRP; MG0001, MG048) and molecular chaperones co-translationally recognize the type I signal sequence of nascent integral membrane proteins⁸⁴. Second, SRPs deliver nascent proteins to their cognate receptor FtsY (MG297) which is associated with the SecA type II preprotein translocon⁸⁴. Finally, the preprotein translocase SecA (MG072) iteratively pushes

nascent proteins through the SecYEG (MG170, MG055, MG476) translocase pore by an ATP-dependent, step-wise mechanism⁸⁴. Several studies have shown that SecDF (MG277)¹⁸⁸ and YidC (MG464)^{103,122} also participate in the translocase pore. SecDF and YidC are believed to increase the efficiency of protein translocation⁸⁴. The exact functions of SecDF and YidC are not known.

The *M. genitalium* post-translational SecA pathway translocates lipoproteins and secreted proteins. First, nascent proteins complete translation. Second, SecA translocons directly recognize the type II signal sequence of nascent proteins. Type II signal sequences are short, positively charged N-terminal sequences⁸⁴. Type II signal sequences are composed of three regions – n, h, and c. The n region is composed of 1-5 positively charged amino acids. The h region is composed of 7-15 hydrophobic amino acids and forms an α -helix. The c region is composed of 3-7 polar amino acids and forms a β -strand. Signal peptidase II cleaves type II signal sequences at lipoboxes distinguished by the sequence L[ASI][GA]C inside the c region. Finally, similar to the co-translational pathway, SecA iteratively translocates nascent proteins into the plasma membrane.

Following translocation *M. genitalium* anchors lipoproteins to the outer leaflet through the covalent ligation of diacylglycerol by diacylglycerol transferase and extracellularly cleaves the type II signal sequence of each lipoprotein and secreted protein at conserved lipoboxes (see **Protein Processing II**). *M. genitalium* does not have a I signal sequence protease³⁸⁰ and does not cleave the signal sequences of integral membrane proteins. Integral membrane protein signal peptides are believed to help anchor proteins to the membrane⁸⁴.

Following insertion into the plasma membrane and signal peptide cleavage, terminal organelle-localized lipoproteins are recruited into the terminal organelle. See the **Terminal Organelle Assembly** process for further discussion.

The stoichiometry, energetics, and kinetics of each protein translocation reaction were reconstructed from the primary literature^{99,299,407} (see Table S3D and S3O). Tomkiewicz et al. reported that SecA translocates 270 pmol amino acid min⁻¹⁴⁰⁷. Doyle et al. reported that SecA translocates 20-30 amino acids per ATP⁹⁹.

Computational Representation

Mathematical Model

This process models integral membrane, lipoprotein, and secreted protein translocation into the cell membrane by the SecA preprotein translocase, the second step of post-translational processing. Because the mechanisms of protein translocation are not well characterized on the genomic scale, this process makes several simplifying assumptions. First, this process decouples translation, N-terminal formylmethionine processing, and translocation by assuming that translocation does not begin until after translation termination and N-terminal formylmethionine processing. Second, this process assumes that each protein is fully translocated in a single time step. Consequently, this process collapses the translocation of each protein into a single reaction, and only represents untranslocated and fully translocated proteins. Intermediate protein localizations are not represented. Third, this process assumes that the mean arrival rate, v_i of translocation events of each protein species i is independently limited by (1) the copy number of untranslocated protein, p_i^u , (2) the copy numbers of intracellular metabolites, m_j , and (3) the copy numbers of the translocation enzymes, e_j . Based on these assumptions, the functional form of v_i is given by

$$v_i = \min \left(\overbrace{p_i^u}^{\text{Protein}}, \overbrace{\left[\min_j \frac{m_j}{M_{ji}^s} \right]}^{\text{metabolites}}, \overbrace{\text{poissonRand} \left(\min_j \frac{e_j}{K_{ji}} \Delta t \right)}^{\text{enzymes}} \right), \quad (\text{C.23})$$

where M_{ji} is the stoichiometry of metabolite j in the translocation of protein species i including ATP and GTP hydrolysis, $M^s = \max(0, -M)$ is the negative part of M , K_{ji} is the experimentally observed catalytic rate of enzyme j in the translocation of protein species i , and $\Delta t = 1\text{s}$ is the simulation time step.

This process implements a stochastic model of the arrival of protein monomer translocation events with relative rates v_i . Until protein, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, v_i , of each translocation event, (2) selects a single translocation event to execute according to a multinomial distribution parameterized by v_i , and (3) executes the selected translocation reaction, updating the copy numbers of protein monomers and metabolites and decrementing the available enzymatic capacity. Algorithm C.21 outlines the implementation of the protein translocation model.

Integration

The **Protein Monomer** and **Protein Complex** states represent the copy number of each protein translocation enzyme. The **Protein Monomer** state also represents the cytosolic- and membrane-localized copy numbers of each protein monomers species. The **Metabolite** state represents the copy number of each intracellular metabolite.

Proteins are synthesized and matured in six steps (see Section C.2.10). This process models protein translocation following deformylation and N-terminal methionine cleavage (see **Protein Processing I** process) and preceding lipoprotein diacylglycerol transfer and lipoprotein and secreted protein signal sequence cleavage (see **Protein Processing II** process).

Initial Conditions

Section C.1.4 outlines the cell state initialization algorithm. Briefly, after the **Mass** state initializes the total cell mass, the **Protein Monomer** state initializes the total copy number of each protein monomer species and initializes all monomers to their mature – processed, folded, modified, and localized – configuration. Second, the **Macromolecular Complexation** and **Ribosome Assembly** processes initialize the copy number of each ribonucleoprotein complex, and set all initialized complexes to their mature – folded and modified – configuration. Finally, several processes including **Transcription** and **Translation** initialize the detailed configurations of RNA polymerases, 70S ribosomes, DNA-binding proteins, and FtsZ.

The untranslocated and translocated configurations of each protein monomer species are initialized with zero copy number because the typical occupancy these configurations is small compared to that of the mature configuration.

Dynamic Computation

Algorithm C.21 outlines the implementation of the protein translocation model.

Fitting

The expression of the protein translocation enzymes was fit to provide sufficient enzymes to quickly translocate proteins, or more specifically to prevent sustained accumulation of untranslocated proteins. See Section C.1.3 for further discussion.

Algorithm C.21. Protein translocation simulation. See Mathematical Model section above for definition of the mathematical notation.

Input: p_i^t copy number of translocated (membrane- or extracellular-localized) protein monomer species i

Let $k_i \leftarrow e_i \Delta t$ be the capacity of enzyme i for protein translocation

repeat

 Calculate translocation rates

foreach protein monomer species i **do**

 Calculate v_i according to Eq. C.23

 Select protein monomer species $i \sim \text{multinomialRand}(1, v_i / \sum_j v_j)$

 Update protein monomer copy numbers: $p_i^u \leftarrow p_i^u - 1, p_i^t \leftarrow p_i^t + 1$

 Update metabolites: $m \leftarrow m - M_{\bullet i}$

 Update enzyme catalytic capacity: $k \leftarrow k - K_{\bullet i}$

until no further translocation possible ($v_i = 0 \forall i$)

C.3.18 Replication

Biology

DNA replication is an integral part of the cell cycle which produces a complete chromosome for each daughter cell. DNA replication is initiated by the formation of a large multimeric DnaA complex at the origin of replication (oriC), which enables the recruiting of the DNA replication machinery to the oriC. Replication proceeds bidirectionally from the oriC to the terminus (terC), half way around the circular chromosome. The chromosome has two strands of base pairing DNA called the leading and lagging strands. Replication of the leading strand proceeds continuously the 5' to 3' direction. Replication of the lagging strand occurs in short Okazaki fragments, also in the 5' to 3' direction¹⁹⁵.

Reconstruction

The DNA replication machinery consists of multiple proteins (See Figure C.11), and there is one set of machinery for each of the two replication bubbles (on each side of the oriC). The replicative DNA helicase serves to unwind the coiled DNA⁴⁷. The DNA primase makes short primers that help initiate the polymerization of a long stretch of DNA. Primers are made once at the origin for the replication of the leading strand, and one primer is made to start each Okazaki fragment⁴⁴². DNA polymerization is carried out by DNA polymerase III molecules, consisting of two core subunits, a gamma complex, and beta-clamps²⁰⁷. One core resides on each of the leading and lagging strands, and is made up of two alpha subunits. The two cores are held together by a gamma complex, consisting of delta, delta prime, gamma, and tau subunits. The core is also bound to a sliding

beta-clamp which helps anchor the polymerase to the DNA and maintain processivity. On the lagging strand, beta-clamps are swapped out each time a new DNA loop is created to make a new Okazaki fragment (See Figure C.11). A “back-up” beta-clamp binds downstream of the lagging DNA loop to facilitate this switch and formation of the next loop²⁰⁷. (The lagging stand polymerization requires the formation of DNA loops, so that the DNA can be polymerized in the opposite, 5' to 3', direction.) DNA ligases connect the separate polymerized Okazaki fragments together³⁵⁹. Finally, single-stranded binding proteins (SSBs) stabilize and protect single-stranded DNA, which is created as the DNA is unwound to make room for the DNA machinery^{66,206}. For example, the lagging stand DNA loop often has long stretches of unwound DNA waiting to be polymerized. All of the replication proteins are described in Table C.15, and all of the parameters used in replication are described in Table C.16.

Table C.15. Enzymes and complexes used in the Replication process class.

Enzymes/Complexes	Composition	Gene Name(s)	DNA Footprints (nt)
DNA helicase	(6) MG094	dnaB	20
DNA primase	(1) MG250	dnaG	14
β -clamp	(2) MG001	dnaN	25
DNA polymerase core	(1) MG031, (1) MG261	polC, polC-2	24
γ -complex	(1) MG007, (1) MG351, (4) MG419	holB, holA, dnaX	26
DNA ligase	(1) MG254	ligA	19
Single stranded binding protein (SSB) 8mer	(8) MG091	ssb	145

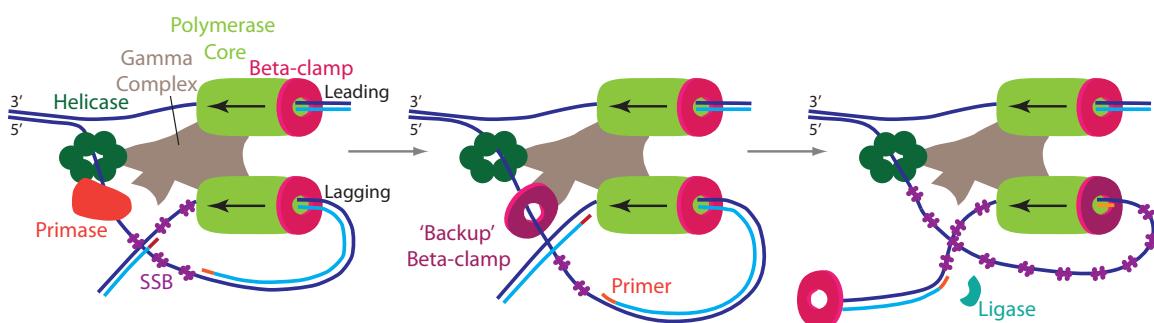


Figure C.11. Schematic of DNA replication.

Computational Representation

Upon replication initiation (the binding of 29 DnaA-ATP molecules near the oriC by the **Replication Initiation** process class), the **Replication** process class tracks the progression of the

Table C.16. Fixed parameters used in the Replication process class.

Parameter	Value	Symbol	Source
OriC position	Base: 1	oriC	304.
TerC position	Base: 290038	terC	304.
Primer length	11 nt	l_{prim}	442.
DNA polymerase elongation rate	100 nt s ⁻¹	k_{el}	323.
SSB complex spacing	30 nt	s_{ssb}	66.
Okazaki fragment mean length	1500 nt	l_{OF}	78, 195, 438.
Ligase kinetic rate	0.04 s ⁻¹	k_{lig}	369.
Current Okazaki fragment length before “back-up” beta-clamp can bind	750 nucleotide	l_{beta}	Set as half l_{OF}
Length of lagging strand loop at the start of Okazaki fragment polymerization	50 nt	l_{loop}	Set to be larger than polymerase footprint
DNA polymerase stall time upon anti-direction collision with RNA polymerase	1.7 s	t_{stall}	220.
SSB dissociation rate	0.3 s ⁻¹	k_{dssb}	206.
Chromosome length	580076		304.
Chromosome sequence	See 304.		304.
DNA footprints of Proteins	See ‘Enzymes/Complexes’		47, 207, 327, 359.
ATP cost of beta-clamp binding	1 ATP molecule	e_{beta}	47.
ATP cost of 1 base pair unwinding	1 ATP molecule	e_{hel}	207.
NAD requirement per ligation event	1 NAD molecule	e_{lig}	359.

replication proteins on the known chromosome sequences. Some bacterial species are known to have multiple replication initiation events during their life cycles. This has never been demonstrated in *M. genitalium* and we model up to 1 chromosome duplication event per cell cycle. Further, the exact mechanism of replication initiation in *M. genitalium* is unknown. *M. genitalium* does not include a DnaC homolog, which in other species is an essential cue for the binding of the replication machinery to the oriC. Here, the binding of 29 DnaA-ATP molecules to the oriC is the cue for replication initiation.

To duplicate the chromosome, the exact position of each replisome protein is tracked over time on both the leading and lagging strands. The polymerization of Okazaki fragments is explicitly modeled. Since the exact primer binding sites for *M. genitalium* Okazaki fragments are uncharacterized, the Okazaki fragment lengths are randomly determined based on a Poisson distribution centered around the mean fragment length (l_{OF}). Polymerization of both the leading and lagging strands is limited by the average rate of polymerization (k_{el}) in Mycoplasmas³²³, the availability of nucleotides and energy, and the available protein binding sites on the chromosome. Further, polymerization of the leading strand is prevented if additional unwinding will lead to too many unprotected single-stranded

bases on the lagging stand, or if the leading polymerase will progress over two Okazaki fragment length distances beyond the lagging strand polymerase. Leading strands are not polymerized past the terC, and the lagging strands are not polymerized past the ends of Okazaki fragments.

This process outputs the progress of replication, which has a big impact on many other processes in the system. For example, the copy number of each gene plays a role in RNA polymerase binding in the Transcription process, and affects the cell’s gene expression. The duplicated regions of the chromosome have to be wound appropriately by the DNA supercoiling process, and the completion of replication triggers other cell cycle events such as the decatenation of the chromosomes by the Chromosome Segregation process.

Collisions of the replication machinery with the transcription machinery (as determined by the **Chromosome** state class) are also handled by the **Replication** process class. A helicase may collide with a RNA polymerase in two ways, head-on (RNA polymerase is traveling in a direction opposite to the helicase) or co-directionally (RNA polymerase is traveling in the same direction as the helicase)²⁵⁷. There is some debate in the literature regarding the pausing of the replication loop upon collisions and whether RNA polymerases are displaced upon collisions^{243,255–257,338}. Based on these studies as well as modeling considerations, we assume that the RNA polymerase is always displaced upon collisions, and the replication loop is only stalled upon a head-on collision. If the helicase hits an RNA polymerase, polymerization pauses, the RNA polymerase falls off, and its transcript is degraded. If it is a head-on collision (the helicase and RNA polymerase were traveling in opposite directions on the DNA), the impact will stall progression of the replication bubble for some amount of time (t_{stall}). If it is a codirectional collision, polymerization will continue at full speed the following time step^{255–257}.

Integration

Table C.17 outlines the states integrated with the **Replication** process class.

Initial Conditions

At the beginning of the simulation, we choose Okazaki fragment lengths randomly based on a Poisson distribution around the mean length of 1500 nucleotides (l_{OF}). From this we obtain the start position of each Okazaki fragment, to be used to determine primer, beta-clamp, and polymerase core binding during the simulation.

Table C.17. State classes connected to the Replication process class.

Connected States	Read from state	Written to state
Chromosome	<ul style="list-style-type: none"> • Whether a DnaA complex has formed at oriC • DNA-bound protein locations • Superhelicity • DNA strand breaks to be ligated • DNA sequence • DNA footprints of proteins • Chromosome regions accessible for protein binding • Damaged DNA bases • OriC position • TerC position • Sequence Length 	<ul style="list-style-type: none"> • Polymerized regions of DNA • DNA-bound protein locations • Unwound bases (Effect on superhelicity) • DNA strand breaks to be ligated

Dynamic Computation

The **Replication** process class is built up of 8 subfunctions which move the replication proteins along the chromosomes, and evolve the DNA copy number of the cell from 1 to 2. These subfunctions are evaluated in random order, such that shared resources across subfunctions, such as energy, are allocated fairly.

1. Initiate Replication

- If a 29-mer DnaA-ATP complex exists at the oriC,
 - If sufficient proteins and metabolites exist to make 2 replisomes,
 - Unwind enough DNA bases ($b_{unwound}$) to bind 2 sets of replication machinery. One set binds on either side of the oriC. All proteins are bound to the leading strand of DNA. Each set includes:
 - 1 helicase
 - 1 primase
 - 2 polymerase cores
 - 1 gamma complex
 - 1 beta-clamp
 - Account for total ATP usage (as well as H₂O usage, and ADP, Pi, and H⁺ production):

$$\text{ATP usage} = 2e_{\text{beta}} + e_{\text{hel}}b_{\text{unwound}}$$

- Note: following the initiation of replication, the dissociation of the DnaA complex is handled by the **Chromosome** state class.

2. Advance replisomes

- If an Okazaki fragment is starting at the current timestep,
 - Bind a primase and a polymerase core to the lagging strand
- Model the unwinding and polymerizing of the DNA. The leading and lagging polymerases and helicases are advanced only so far as the following conditions are met:
 Let N = the number of polymerized/unwound bases on a given strand in the given timestep.
 Let P be the last polymerized position. P varies from 0 (at the oriC) to 290038 (at the terC).
 - $N \leq l_{\text{prim}}$ if a primer is being synthesized
 - $N \leq k_{\text{el}}$
 - Available dNTPs $\leq dNTP$ requirements
 - Available ATP $\leq Ne_{\text{hel}}$
 - # Bound SSBs = $\frac{\text{Length of lagging single strand region}}{\text{SSB 8mer footprint} \times s_{\text{ssb}}}$
 - Upstream DNA regions accessible to replisome proteins (no DNA damage, non-displacable proteins)
 - Leading strand $P \leq terC$, Lagging strand $P \leq Okazaki fragment length$
 - Leading strand $P \leq 2L_{\text{OF}} +$ lagging strand P
 - $P \leq$ position of colliding RNA polymerase

3. Bind and Unbind SSBs

SSBs nucleate as a tetramer, binding single-stranded DNA around them. These tetramers generally bind in sets of two.

- For all single-stranded stretches of DNA,
 - Randomly bind SSBs (as 8mers) to deterministically selected position (with fixed spacing: s_{ssb})
- Randomly release SSB 8mers according to the dissociation rate (k_{dssb})
- Dissociate all free 8mer SSBs into two tetramer SSBs

4. Bind “Back-up” Beta-Clamp

The formation of a new Okazaki fragment requires the formation of a new lagging strand DNA loop. To prepare for this event, while the current Okazaki fragment is being polymerized, a “back-up” beta-clamp is bound just downstream of the start position of the next Okazaki fragment. The beta-clamp assembles as a dimer on the chromosome, and the binding event requires hydrolysis of one ATP molecule to ADP (e_{beta}).

- If $\begin{cases} \# \text{ available beta-clamp monomers} > 2 \\ \# \text{ available } ATP ; e_{\text{beta}} \\ \text{Position on the DNA is accessible} \\ \text{Helicase has passed beta-clamp binding site} \\ \text{Okazaki fragment length} > l_{\text{beta}} \end{cases}$
- Bind “back-up” beta-clamp
- Decrement ATP, H₂O, and increment ADP, Pi, and H⁺

5. Terminate Okazaki fragments

When an Okazaki fragment has been completely polymerized, its beta-clamp is released, and the end of the Okazaki fragment is marked as having a single-strand break to be ligated by DNA ligase. The lagging strand primase and polymerase associate with the “back-up” beta clamp that was previously bound. This forms a new lagging strand DNA loop.

- If $\begin{cases} \text{Okazaki fragment has been completely polymerized} \\ \text{Number of bound SSBs} = \frac{\text{Length of lagging single strand region}}{\text{SSB 8mer footprint} \times s_{\text{sss}}} \\ \text{A “back-up” clamp has been bound on the lagging strand} \\ \text{The replication machinery on the leading strand has advanced beyond the Okazaki start site} \end{cases}$
- Release beta-clamp
- Mark Okazaki fragment as having a single-strand break to be ligated by DNA ligase
- Associate lagging strand primase and polymerase with the “back-up” beta clamp that was previously bound (This forms a new lagging strand DNA loop.)

6. Ligate DNA

- If single-strand breaks exist (usually between polymerized Okazaki fragments),
 - Ligate these breaks in random order up to the limits of:
 - DNA ligase availability
 - Ligase kinetics (k_{lig})
 - NAD availability
 - Decrement NAD, and increment AMP, NMM, and H^+

7. Terminate Replication

- If leading and lagging strands to both sides of the terC have been completely polymerized,
 - Release all replication machinery from the chromosome
 - Mark terC as having a single-strand break to be ligated by DNA ligase

Representation of Replication Machinery on the Chromosome

The **Replication** process class involves keeping track of the specific positions of the replication proteins on the chromosome. We define where on the chromosome the proteins bind based on the following rules:

1. The Helicase is centered on the boundary between double stranded DNA (wound) and single-stranded DNA (unwound). The position over which it is centered is the next position to be unwound.
2. There is no gap between the helicase and leading strand DNA polymerase core or between the leading strand polymerase core and the leading strand beta-clamp. Therefore, movement of the helicase controls the movement of the leading strand polymerase core.
3. There is no gap between the lagging strand DNA polymerase core and the beta-clamp on the current Okazaki fragment.
4. The positions over which the polymerase cores are centered are the next position to be polymerized.
5. “Back-up” beta-clamps bind slightly upstream of the start site of the next Okazaki fragment to be polymerized such that when the next Okazaki fragment starts polymerizing, there will be no gap between the polymerase and beta-clamp, and the polymerase core will be centered on the Okazaki fragment start site.
6. At replication initiation, the mother strands are separated such that the leading polymerase cores are centered at $\text{oriC} \pm 1$ base and the helicases are 11 nucleotides (l_{prim}) ahead

7. During replication initiation (and the final step of replication after the last Okazaki fragment has completed), the lagging polymerase, beta-clamp, and primase are accounted for as part of a complex on the leading strand (containing also the helicase, leading polymerase, gamma complex, and leading beta-clamp). At all other times, the lagging polymerase, lagging beta-clamp, and primase are accounted for as a complex on a different strand. This allows for separate tracking of the leading and lagging polymerase positions.
8. SSBs are bound at a fixed spacing (s_{ssb}) in the single-stranded regions of DNA.

C.3.19 Replication Initiation

Biology

The **Replication Initiation** process determines when during the cell cycle chromosome duplication begins. This replication initiation time is therefore very important in determining the cell's division time.

Reconstruction

The mechanism of replication initiation used here is modeled after that described for *E. coli* by Messer involving the protein DnaA (MG469)²⁴⁹. Chromosome **Replication** begins when a complex of 29 DnaA-ATP molecules assembles near the replication origin, OriC, at specific DNA motifs called R1-R5. The assembly of this complex is rare because the DnaA molecules are titrated out by approximately 2000 additional binding sites, or "DnaA boxes" that exist all around the chromosome¹⁴⁸. DnaA needs to be bound to ATP or ADP to bind to these sites, and binds on and off these sites throughout the cell cycle. Binding of the DnaA-ATP at the R1-R5 sites is cooperative, enabling the large complex to form at this specific location on the chromosome.

All of the parameters used in the **Replication Initiation** process class are described in Table C.18.

Parameter Assignment

All of the rate constants used in this process class were obtained from previous models of replication initiation^{11,43}. The site and state cooperativity constants were fit according to the cell cycle length. The site and state cooperativity constants directly affect the time required for replication initiation.

Table C.18. Fixed parameters used in the Replication Initiation process class.

Parameter	Value	Symbol	Source
Positions of all the DnaA binding sites on the chromosome(s)	(see Table S3L)		Calculated from motif in 73.
Factor by which DnaA-ATP to oriC site binding probability increases when other sites are bound	85.6909	C_{site}	See Parameter Fitting
Factor by which DnaA-ATP to oriC site binding probability increases when x^4 sized DnaA complex has formed at oriC	1	C_{state}	See Parameter Fitting
Rate for DnaA-ATP binding high affinity DnaA boxes	$25 \text{ nM}^{-1} \text{ h}^{-1}$	kb1ATP	43.
Rate for DnaA-ATP binding medium affinity DnaA boxes	$0.6 \text{ nM}^{-1} \text{ h}^{-1}$	kb2ATP	43.
Rate for DnaA-ATP dissociating from the DNA	20 h^{-1}	kd1ATP	43.
Rate for DnaA-ADP binding high affinity DnaA boxes	$2.5 \text{ nM}^{-1} \text{ h}^{-1}$	kb1ADP	43.
Rate for DnaA-ADP binding medium affinity DnaA boxes	$0.61 \text{ nM}^{-1} \text{ h}^{-1}$	kb2ADP	43.
Rate for DnaA-ADP dissociating from the DNA	20 h^{-1}	kd1ADP	43.
Rate for DnaA-ADP to DnaA-ATP regeneration	2.3 h^{-1}	k_Regen	11.
Rate for DnaA-ADP to DnaA-ATP regeneration catalyzed by membrane lipids	0.018 g L^{-1}	K_Regen_P4	11.

The total cell cycle length has been experimentally measured, and the time required for **Cytokinesis** and **Chromosome Replication** can be estimated from their process classes. Thus:

$$\text{time}_{\text{ReplicationInitiation}} = \text{time}_{\text{CellCycle}} - \text{time}_{\text{Cytokinesis}} - \text{time}_{\text{Replication}} \quad (\text{C.24})$$

The site and state cooperativity constants were fit such that on average the simulated cell divides in the experimentally measured amount of time.

Computational Representation

The methods we use to model replication initiation are based on existing models by Atlas et al. and Browning et al.^{11,43}. First, DnaA rapidly associates with ATP, and in some cases DnaA-ATP can be converted to DnaA-ADP. DnaA-ATP and DnaA-ADP molecules bind to and release from the binding sites on the chromosome. The number of binding/unbinding events that occur in a given

period of time is determined by the number of available DnaA-ATP and DnaA-ADP molecules and binding/unbinding rates calculated as in Browning et al.⁴³.

Binding Sites

In addition to the R1-R5 sites at the origin, there 2227 DnaA binding sites around the chromosome. All of the DnaA binding sites are based on the *M. genitalium* motifs described by Cordova et al.⁷³. The motifs are 9 bases long. A high affinity site is defined as one that exactly matches the motif or its reverse complement (148 sites). A medium affinity motif is defined as one that matches 8 of the 9 bases in the motif or its reverse complement (2079 sites), and a low affinity site is one that only matches 7 of 9 bases. Our model allows binding to all of the high and medium affinity sites outside of the oriC region.

We recognize 5 DnaA binding sites in the OriC region: one high affinity (R4), three medium affinity (R1-R3), and one low affinity (R5), to mimic *E. coli*'s 5 R-sites²³⁴ and the 5 R-sites predicted for *M. genitalium*⁷³. These sites reside between the genes MG469 and MG470 (bases: 578581-579224). There are 9 high and medium affinity motifs in this region, but we only recognize 4 to match the *E. coli* pattern, and therefore ignore the sites at genome positions 578837, 578855, 578881, 578966, and 579139. R5 matches 7 of 9 bases of the motif, so it is a very weak binder of DnaA. Since we do not know its exact mechanism/purpose we bind this site after a 28-mer DnaA-ATP initiator complex at R1-R4 is formed, and the R5 binding triggers initiation. This is the only low affinity binding site included in our model.

7 States

The 28-mer initiator complex consists of 7 DnaA-ATP molecules bound to each of the R1-R4 sites²³⁴. This binding is split up into 7 serial states, and in each state one additional DnaA-ATP molecule binds to each of the R1-R4 sites. All four sites must be bound before transition to the next state.

Cooperative Binding

DnaA-ATP binding at the origin is cooperative. There are two forms of cooperativity in our model of replication initiation. The first type of cooperativity is state cooperativity, which helps increase the probability of transitioning into higher states. Once the binding within a state is completed, the cooperativity to transition into the next state is calculated as:

$$\text{State Cooperativity Factor} = C_{\text{state}}(\text{State} - 1) \quad (\text{C.25})$$

where C_{state} is a fixed cooperativity constant. This calculation is described in Figure C.12.

The second type of cooperativity is site cooperativity, which describes the effects of the binding of one or more of the R1-R4 sites, on the probability of binding the other sites. The site cooperativity factor of a given site is 1 unless certain other R-sites are already bound, in which case the cooperativity factor changes to C_{site} . R4 is a high affinity site, so it generally is the first to bind and has the highest cooperative effect on the other sites. The site cooperativity rules can be seen in Figure C.13).

R5 is also bound by cooperativity, given the presence of a complete 28-mer DnaA-ATP initiator complex at the R1-R4 sites. The resulting DnaA-ATP 29-mer is the trigger to begin the **Replication** of the chromosome.

Post-Replication Initiation

Following initiation, the DnaA-ATP complex at the origin dissociates and hydrolyzes. DnaA molecules can continue to bind around the chromosome. The formation of a second complete initiation complex is rare because the time left in the cell cycle is generally less than that needed for an initiation event and because DnaA molecules are further titrated with a 2nd set of binding sites on the 2nd chromosome. For simplicity, we do not model a second DNA replication event.

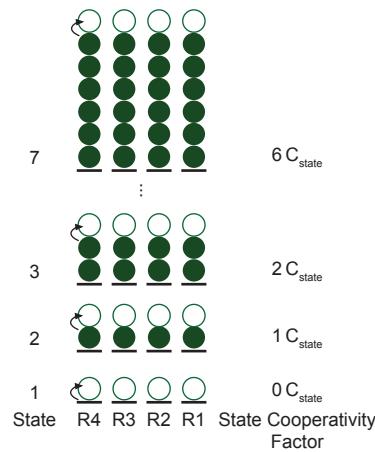


Figure C.12. Calculation of the state cooperativity factor for DnaA-ATP binding at the origin of replication.

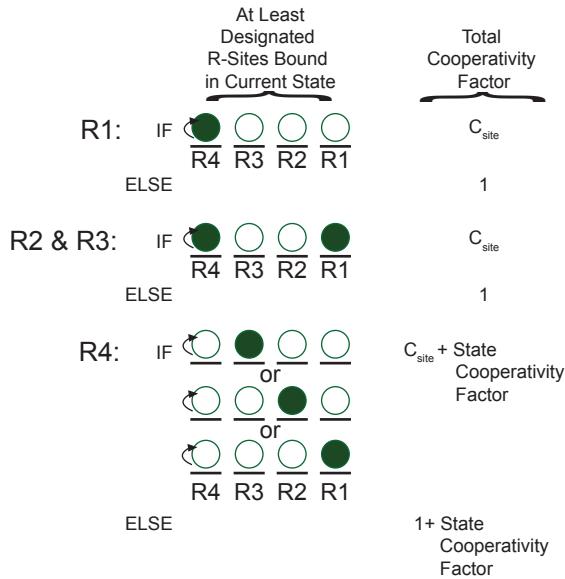


Figure C.13. Calculation of the total cooperativity factor for DnaA-ATP binding at the origin of replication.

Integration

Table C.19. State classes connected to the Replication Initiation process class.

Connected States	Read from State	Written to State
Chromosome	<ul style="list-style-type: none"> • Positions on chromosome(s) where DnaA molecules are bound • Accessible DnaA binding sites • Copy numbers of DnaA binding sites 	<ul style="list-style-type: none"> • Updated positions on chromosome(s) where DnaA molecules are bound
Mass	<ul style="list-style-type: none"> • Membrane mass (used to calculate membrane concentration) 	
Geometry	<ul style="list-style-type: none"> • Cell volume (used to calculate membrane concentration) 	

Initial Conditions

This process is initialized to a steady state. The steady state amounts of free, medium affinity site bound, and high affinity site bound DnaA-ATPs and DnaA-ADPs are found using non-linear constrained optimization where we try to identify a state which is a stable point and which maximizes

the amount of high affinity site bound DnaA-ATP. All DnaA proteins are either ADP or ATP bound at the initial timestep. There are also no DnaA polymers at the R1-R4 sites at the start of the simulation.

Dynamic Computation

At each timestep, we perform the following algorithm:

1. DnaA activation

- Deterministically form DnaA-ATP complexes up to the limit of available DnaA and ATP. The kinetics of DnaA activation are not known, and are not modeled.

$$\text{Number of activation events} = \min \begin{cases} \text{Number of available ATP molecules} \\ \text{Number of free DnaA molecules} \end{cases}$$

- Update the counts of free DnaA, DnaA-ATP, and ATP

2. DnaA-ATP complex dissociation

A complex of multiple bound DnaA-ATPs is only found near the oriC. This complex can be released from the chromosome by the DNA polymerase at the start of DNA replication. The resulting DnaA-ADP molecules are able to re-bind to the chromosome or reactivate to DnaA-ATP.

- If 29-mer DnaA-ATP complex is disrupted by the replisome machinery,
 - Dissociate the complex to individual DnaA-ATP molecules
 - Hydrolyze the ATP to form free DnaA-ADP molecules
 - Decrement the counts of DnaA-ATP complexes and H₂O molecules
 - Increment the counts of free DnaA-ADP, H⁺, and Pi

3. Binding DnaA-ATP and DnaA-ADP

Up to 7 DnaA-ATPs can polymerize at each of the origin sites and only one DnaA-ATP can bind at each site outside of the origin. Only one DnaA-ADP can bind at each site.

- If the chromosomes are sufficiently supercoiled,
 - Stochastically select both high and low affinity sites at the origin and around the chromosome to bind DnaA-ATP and DnaA-ADP molecules. Binding proceeds such that the rate of binding each site is as follows (where the cooperativity factor, depends on the

polymerization status of the R1-R4 boxes):

$$\begin{aligned} \text{rate of DnaA-ATP binding R4 (high affinity)} &= \frac{\text{kb1ATP} \times \text{numFreeDnaAATP}}{\text{cell volume} \times \text{cooperativity factor}} \\ \text{rate of DnaA-ATP binding R1-R3 (medium affinity)} &= \frac{\text{kb2ATP} \times \text{numFreeDnaAATP}}{\text{cell volume} \times \text{cooperativity factor}} \\ \text{rate of DnaA-ATP binding non-origin high affinity sites} &= \frac{\text{kb1ATP} \times \text{numFreeDnaAATP}}{\text{cell volume}} \\ \text{rate of DnaA-ATP binding non-origin medium affinity sites} &= \frac{\text{kb2ATP} \times \text{numFreeDnaAATP}}{\text{cell volume}} \end{aligned}$$

The rate equations for DnaA-ADP binding are the same as above, except that the rate constants are kb1ADP and kb2ADP.

4. Cooperativity

The binding of DnaA-ATP to the R1-R4 sites is cooperative at two levels. Site cooperativity increases the probability of binding successive sites within a state. State cooperativity increases the probability of entering successive states. The cooperativity factors for sites R1-R4 are calculated as follows:

- Calculate the State Cooperativity Factor as described in Figure C.12:

$$\text{State Cooperativity Factor} = C_{\text{state}}(\text{State} - 1)$$

- Calculate the Site Cooperativity as described in Figure C.13. The Site Cooperativity Factor for a given site is 1 or C_{site} depending on the occupancy of the other R-sites
- Assign the total Cooperativity Factor for sites R1, R2, and R3 as their respective Site Cooperativity Factors
- Assign the total Cooperativity Factor for R4 as its Site Cooperativity Factor + State Cooperativity Factor

The cooperativity factors calculated here will be used in the rate calculation in Step 3.

5. Displacing DnaA-ATP and DnaA-ADP

We model the release of DnaA-ATP and DnaA-ADP molecules from their binding sites throughout the cell cycle at the following rates:

- Stochastically select DnaA-ATP and DnaA-ADP molecules to unbind from both high and low affinity sites at the origin and around the chromosome, such that the number of unbinding

events is in agreement with the following rates:

$$\text{rate of DnaA-ATP displacement} = kd1ATP$$

$$\text{rate of DnaA-ADP displacement} = kd1ADP$$

6. Reactivation of DnaA-ADP to DnaA-ATP

The rejuvenation of DnaA-ADP to DnaA-ATP is incorporated similarly to that described by Atlas et al.¹¹. This reaction is promoted by the acidic phospholipids cardiolipin and phosphatidylglycerol.

- Deterministically reactivate free DnaA-ADP at a rate of:

$$\text{rate of reactivation} = \text{numFreeDnaAADP} \frac{k_{\text{Regen}} \text{membraneConcentration}}{K_{\text{Regen}} P4 + \text{membraneConcentration}}$$

where the membrane concentration is the grams of membrane in the cell divided by the volume of the cell. These values are read from the **Mass** and **Geometry** state classes.

7. Replication-dependent bound DnaA-ATP release

Atlas et al. included a global term for the effect of active beta-clamps (a part of the DNA polymerase machinery) on bound DnaA-ATP inactivation¹¹. Our model predicts the exact position of the active beta-clamps on the chromosome. Therefore, we do not need to use a global term, and instead can model the release of a DnaA-ATP molecule exactly when the beta-clamp encounters it on the DNA. We however cannot distinguish between free DnaA-ATP in the cytosol and DnaA-ATP that has been recently released by a beta-clamp. Therefore, we model the release in the form of DnaA-ATP and not the hydrolysis to DnaA-ADP. This beta-clamp-dependent release is handled by the DNA **Replication** process class and **Chromosome** state class.

C.3.20 Ribosome Assembly

Biology

Ribosomes are large ribonucleoproteins which synthesize polypeptides. The *M. genitalium* 70S ribosome is composed of two subunits – the 30S and 50S ribosomal particles – which assemble at

the mRNA Shine-Dalgarno sequence with assistance from initiation factors 1-3^{278,283}. This process models the enzyme-catalyzed formation of 30S and 50S ribosomal particles.

Reconstruction

Table S3N lists the observed subunit composition of the 30S and 50S particles¹⁸⁸. The 30S particle is composed of 1 RNA and 20 protein monomer subunits. The 50S particle is composed of 2 RNA and 32 protein monomer subunits. The 30S and 50S particles have both been shown to assemble in stereotyped patterns with assistance from several GTPases^{36,41,81,189,283,298,343,416}. Era (MG387) and RbfA (MG143) have been associated with 30S particle formation^{41,298}. EngA (MG329), EngB (MG335), Obg (MG384), and RbgA (MG442) have been associated with 50S particle formation^{41,298,343,416}. The exact functions, kinetics, and energetics of the six GTPases are unknown.

Computational Representation

Mathematical Model

Because 30S and 50S ribosomal assembly are not well characterized, this process makes several simplifying assumptions to model ribosomal particle assembly. First, the process only represents individual rRNA transcripts and protein monomers and fully formed ribosomal particles. Intermediate states of ribosomal particle formation are not represented. Second, this process assumes that ribosomal particle formation is kinetically fast and energetically favorable such that ribosomal particle formation proceeds to completion within the 1 s simulation time step. Finally, the process assumes that each GTPase hydrolyzes 1 GTP molecule per ribosomal particle. With these assumptions, the rate of formation of ribosomal particle i is given by

$$\Delta c_i = \begin{cases} \min \left(\underbrace{\min_j \frac{r_j}{R_{ji}}}_{\text{rRNA}}, \underbrace{\min_j \frac{p_j}{P_{ji}}}_{\text{protein}}, \underbrace{\frac{m_{GTP}}{\eta_i}}_{\text{GTP}} \right) & \overbrace{\left(\min_j \frac{e_j}{E_{ji}} \right)}^{\text{GTPases}} > 0 \\ 0 & \text{otherwise} \end{cases} \quad (\text{C.26})$$

where m , r , p , c , and e represent the copy number of each metabolite, rRNA transcript, protein monomer, ribosomal particle, and GTPase, R_{ij} and P_{ij} represent the stoichiometry of rRNA transcript or protein monomer i in ribosomal particle j , E_{ij} is true if ribosomal particle j formation requires GTPase i and false otherwise, and $\eta_i = \sum_j E_{ij}$ is the GTP cost of forming ribosomal

particle i .

This process implements a stochastic model of the arrival of ribosomal particle assembly events with relative rates Δc_i . Until RNA, protein, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, Δc_i , of each assembly event, (2) selects an event to execute according to a binomial distribution parameterized by Δc_i , and (3) executes the selected assembly reaction, updating the copy numbers of RNA, protein, and metabolites and decrementing the available enzymatic capacity. Algorithm C.22 outlines the implementation of the Boolean ribosomal particle assembly model.

Integration

The **Rna** and **Protein Monomer** states represents the free copy numbers of each RNA and protein monomer species. The **Protein Complex** state represents the copy numbers of 30S and 50S ribosomal particles and the 70S ribosome. The **Ribosome** state represents the (t)mRNA location of each 70S ribosome.

Several processes including **Transcription** and **Translation** model the synthesis and maturation of rRNA transcripts and protein monomers (see Section C.2.12 and C.2.10). The **Translation** process models (1) translation initiation: 70S ribosome formation at mRNA Shine-Dalgarno sequences, (2) translation elongation: 70S ribosome-catalyzed polypeptide synthesis, and (3) translation termination: mRNA release and 70S ribosome disassembly. The **Translation** process also models ribosome stalling whereby the mRNA template is replaced by a tmRNA and the 70S ribosome synthesizes a SsrA degradation tag at the polypeptide C-terminus, marking the nascent polypeptide for degradation. The **Protein Activation** process models the effect of antibiotics on the catalytic activity of 30S and 50S ribosomal particles.

Initial Conditions

After the **Mass** state initializes the total cell mass, the **Rna** and **Protein Monomer** states initialize the total copy number of each rRNA and protein monomer species. Next, the **Ribosome Assembly** process initializes the total copy numbers of 30S and 50S ribosomal particles to a steady-state of the ribosome assembly model by evaluating Algorithm C.22 with excess GTP. Finally, the **Translation** process initializes the copy number of 70S ribosomes, and randomly positions each 70S ribosome on mRNA weighted by the expressed copy number of each codon (see Algorithm C.4).

Dynamic Computation

Algorithm C.22 outlines the implementation of the Boolean ribosomal particle assembly simulation.

Algorithm C.22. Ribosome assembly simulation. See Mathematical Model section above for definition of the mathematical notation.

```

foreach ribosomal particle  $i$  in random order do
    Calculate the extent of ribosomal particle formation,  $\Delta c_i$ , using Eq. C.26
    Form ribosomal subunits and decrement the copy numbers of rRNA transcripts and protein monomers.
     $r \leftarrow r - R_{\bullet i} \Delta c_i$ 
     $p \leftarrow p - P_{\bullet i} \Delta c_i$ 
     $c_i \leftarrow c_i + \Delta c_i$ 
    Hydrolyze GTP
     $m_{GTP} \leftarrow m_{GTP} - \eta_i \Delta c_i$ 
     $m_{GDP} \leftarrow m_{GDP} - \eta_i \Delta c_i$ 
     $m_{Pi} \leftarrow m_{Pi} - \eta_i \Delta c_i$ 
     $m_{H_2O} \leftarrow m_{H_2O} - \eta_i \Delta c_i$ 
     $m_{H^+} \leftarrow m_{H^+} - \eta_i \Delta c_i$ 

```

Fitting

The expression of each ribosomal RNA and protein monomer was fit to provide sufficient ribosomes for translation and prevent sustained amino acid accumulation. See Section C.1.3 for further discussion.

C.3.21 RNA Decay

Biology

In presence of ribonucleases such as ribonuclease R, RNAs have relatively short half-lives compared to that of other macromolecules (eg. protein, DNA) and the *M. genitalium* doubling time. The relatively short half-lives of RNAs enables the small *M. genitalium* with its very small pool of RNAs and particularly mRNAs to sample a broader range of configurations of the RNA pool over a shorter period that would be possible with longer half-lives. This helps the cell more finely tune the expression of proteins, more efficiently execute cell-cycle dependent events, and respond to the external environment. However, this enhanced fitness due to short RNA half-lives comes at a large energetic cost. In addition to ribonucleases, aminoacylated RNAs require peptidyl tRNA hydrolase to release their conjugated amino acids. This process decays all species of RNA, and at all maturation states including aminoacylated states.

Reconstruction

RNA decay is modeled as requiring ribonuclease R (MG104). The decay of tRNAs also requires peptidyl tRNA hydrolase (Pth: MG083). Given the availability of the necessary decay enzymes, RNAs are randomly selected for degradation by a Poisson probability distribution based on the RNA half lives. All of the parameters used in the Replication Initiation process class are described in Table C.20.

Table C.20. Fixed parameters used in the RNA Decay process class.

Parameter	Value	Symbol	Source
Rate of peptidyl tRNA hydrolase activity	0.7 s^{-1}	k_{hyd}	248.
Half lives of all RNA species	See Table S3Y	$t_{1/2i}$	29.
Decay rates of all RNA species		$k_{\text{decay } i}$	Derived from RNA half lives. See Transcription Process Class .
Reactants and products of decay reactions of all RNAs		M_{decay}	Computed from RNA sequences

Computational Representation

Half-lives of all the RNA species are largely based on experimental measurements of homologous *E. coli* genes. The *E. coli* genes are mapped to the *M. genitalium* genes by homology. (Refer to the **Transcription Process Class** for additional details regarding the determination of RNA half-lives.) RNA species are randomly selected to decay from a Poisson distribution based on the half-life of each RNA species. This process class contains no intermediate representation of RNA degradation. RNA degradation is treated as an all-or-nothing event that proceeds to completion in a single timestep. Upon degradation, water is used to break the nucleotide-nucleotide bonds and the nucleotides are recycled. All aborted transcripts (due to stalled RNA polymerases or RNA polymerases that have been knock off of the DNA) are also degraded by this process.

Integration

Table C.21. State classes connected to the RNA Decay process class.

Connected State	Read from state	Written to state
Rna	<ul style="list-style-type: none"> • Count of each RNA species • Decay Rates of all RNAs 	<ul style="list-style-type: none"> • Updated count of each RNA species
Transcript	<ul style="list-style-type: none"> • Aborted transcript sequences 	

Initial Conditions

No initialization steps are required for this process.

Dynamic Computation

1. Determine the counts of free ribonuclease R (N_R) and peptidyl tRNA hydrolase (N_T) from the **Protein Monomer** State Class
2. Determine the limits on the number of RNA decay events, D_R :
(The number of tRNA decay events is designated D_T)
 - (a) If $N_R = 0$, then $D_R = 0$
(All RNAs require ribonuclease R in order to decay, but ribonuclease R has a high rate of activity, so as long as $N_R > 0$, RNA decay can occur)
 - (b) $D_T = N_T k_{\text{hyd}}$
(The tRNA decay limit is dependent on the peptidyl tRNA hydrolase availability and kinetic rate)
3. Considering each RNA independently, decide whether to decay an RNA by random number selection from a Poisson distribution with a rate parameter, λ calculated as:

$$\lambda = RNA_i k_{\text{decay}} i \quad (\text{C.27})$$

where RNA_i is the count of a given RNA species i

4. If aborted transcripts exist in the cell due to RNA polymerase stalling or RNA polymerase displacement on the DNA, decay all aborted sequences
5. Calculate the total number of metabolites used and produced from the decay of the given RNA

species, using the matrix M_{decay} , update the counts of existing RNAs and metabolites.

C.3.22 RNA Modification

Biology

Bacteria employ post-transcriptional base modification in order to degenerately encode approximately 61 triplet codes using far fewer tRNA species than possible using only Watson-Crick base pairing⁴. Modification of wobble position 34 is believed to be most important for improving codon recognition⁴. Bacteria modify several positions in addition to position 34⁴. Modifications distant from the anticodon are believed to help tRNAs properly fold and stabilize their catalytically active structures⁴. Bacteria also post-transcriptionally modify rRNA²⁹⁸. rRNA modifications are believed to help stabilize rRNA, participate in protein synthesis, and confer resistance to ribosomal inhibitors²⁹⁸. The exact role of rRNA modification is unknown²⁹⁸. This process models tRNA and rRNA modification.

Reconstruction

The *M. genitalium* tRNA modification complement was reconstructed based on the observed complement of *E. coli* modifications. Table S3AA describes the reconstruction process in detail. First, the *E. coli* modification complement was reconstructed^{31,32,49,111,210,224,341}. Second, modifications situated in conserved motifs were transferred to *M. genitalium*. The *M. genitalium* rRNA modification complement was similarly reconstructed (see Table S3Z). The stoichiometry and kinetics of each RNA modification reaction were reconstructed based on an extensive review of the literature (see Table S3O, S3Z, and S3AB).

Computational Representation

Mathematical Model

This process models non-coding RNA modification. Because the kinetics of RNA modification are not well characterized, this process make several simplifying assumptions. First, this process assumes that each RNA is fully modified in a single time step. Consequently, this process collapses the modification of each RNA into a single reaction, and only represents unmodified and fully modified RNA. Intermediate modification configurations are not represented. Second, this process assumes

that the mean arrival rate, v_i of modification events of each RNA species i is independently limited by (1) the copy number of unmodified RNA, r_i^u , (2) the copy numbers of intracellular metabolites, m_j , and (3) the copy numbers of RNA modification enzymes, e_j . Based on these assumptions, the functional form of v_i is given by

$$v_i = \min \left(\overbrace{r_i^u}^{\text{RNA}}, \overbrace{\left[\min_j \frac{m_j}{M_{ji}^s} \right]}^{\text{metabolites}}, \overbrace{\text{poissonRand} \left(\min_j \frac{e_j}{K_{ji}} \Delta t \right)}^{\text{enzymes}} \right), \quad (\text{C.28})$$

where M_{ji} is the stoichiometry of metabolite j in the modification of RNA species i , $M^s = \max(0, -M)$ is the negative part of M , K_{ji} is the experimentally observed catalytic rate of enzyme j in the modification of RNA species i , and $\Delta t = 1\text{s}$ is the simulation time step.

This process implements a stochastic model of the arrival of RNA modification events with relative rates v_i . Until RNA, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, v_i , of each modification event, (2) selects a single modification to execute according to a multinomial distribution parameterized by v_i , and (3) executes the selected modification reaction, updating the copy numbers of RNA and metabolites and decrementing the available enzymatic capacity. Algorithm C.23 outlines the implementation of the RNA modification model.

Integration

The **Rna** state represents the unmodified and modified copy numbers of each RNA species. The **Metabolite** state represents the copy number of each intracellular metabolite. The **Protein Monomer** and **Protein Complex** states represent the copy number of each RNA modification enzyme.

RNA are synthesized and matured in four steps (see Section C.2.12). This process models the modification of individual non-coding RNA following RNA processing (see **RNA Processing** process). The **Macromolecular Complexation** and **Ribosome Assembly** processes model the formation of macromolecular complexes, including the 30S and 50S ribosomal particles. The **Translation** process models the function of m-, r-, s-, and tRNA in translation.

Initial Conditions

Section C.1.4 outlines the cell state initialization algorithm. Briefly, after the **Mass** state initializes the total cell mass, the **Rna** state initializes the total copy number of each RNA species and initializes all RNA to their mature – processed and modified – configuration. Second, the **tRNA Aminoacylation** process initializes tRNA to the aminoacylated configuration. Next, the **Macromolecular Complexation** and **Ribosome Assembly** processes initialize the copy number of each ribonucleoprotein complex. Finally, the **Translation** process initializes the mRNA location of each 70S ribosome.

Dynamic Computation

Algorithm C.23 outlines the implementation of the RNA modification model.

Algorithm C.23. RNA modification simulation. See Mathematical Model section above for definition of the mathematical notation.

Input: r_i^p copy number of modified RNA species i
 Let $k_i \leftarrow e_i \Delta t$ be the capacity of enzyme i for RNA modification
repeat
 Calculate modification rates
foreach non-coding RNA species i **do**
 Calculate v_i according to Eq. C.28
 Select non-coding RNA species $i \sim \text{multinomialRand}(1, v_i / \sum_j v_j)$
 Update RNA copy numbers: $r_i^u \leftarrow r_i^u - 1, r_i^m \leftarrow r_i^m + 1$
 Update metabolites: $m \leftarrow m - M_{\bullet i}$
 Update enzyme catalytic capacity: $k \leftarrow k - K_{\bullet i}$
until no further modification possible ($v_i = 0 \forall i$)

Fitting

The expression of the RNA modification enzymes was fit to provide sufficient enzymes to quickly modify RNA, or more specifically to prevent sustained accumulation of unmodified RNA. See Section C.1.3 for further discussion.

C.3.23 RNA Processing

Biology

Bacteria transcribe genes both individually as well in groups referred to as transcription units or operons. Operonic transcription has several advantages compared to single-gene transcription. First,

operonic transcription allows cells to minimize the number of transcriptional regulators required to control gene expression. Second, operonic transcription increases the likelihood that groups of gene products have similar stoichiometry. At the same time, operonic transcription carries the costs of synthesizing intercistronic RNA and cleaving operonic non-coding transcripts into individual RNAs. Additionally, operonic transcription doesn't provide separate control of the expression of each gene. This process models operonic RNA cleavage into individual RNA gene products.

Reconstruction

Transcription Unit Structure

The transcription unit organization of the *M. genitalium* chromosome was reconstructed by mapping the “suboperon” structure of *M. pneumoniae* chromosome defined by Güell et al.¹⁴⁰ on to that of *M. genitalium* (see Table S3U) with two modifications. First, non-coding RNA were organized into transcription units according to their “reference operons”¹⁴⁰. Second, all mRNAs either not associated with suboperons, without *M. pneumoniae* homologs, or which have been rearranged since divergence from *M. pneumoniae*, were assigned to their own transcription units. The *in silico* *M. genitalium* chromosome is organized into 335 transcription units containing 525 genes.

Leader Sequences

Bacteria also transcribe 3' and 5' leader sequences before and after each non-coding RNA gene. These leader sequences must be cleaved to produce functional non-coding RNA.

mRNA Cleavage

M. genitalium mRNA cleavage is not well described and is not modeled. The model assumes *M. genitalium* polycistronic mRNA are not cleaved.

rRNA Cleavage

E. coli rRNA have been shown to be transcribed as a single 30S transcript which is cleaved into individual 5S, 16S, and 23S rRNA transcripts by the action of several ribonucleases²⁹⁸. The *E. coli* rRNA cleavage scheme was adapted and simplified for the reduced ribonuclease complement of *M. genitalium* by removing cleavage reactions catalyzed by non-homologous enzymes. Figure C.14

illustrates the reconstructed *M. genitalium* rRNA cleavage scheme. First, ribonuclease III (MG367) hydrolytically cleaves the 30S rRNA into 5S, 16S, and 23S rRNA precursors. Second, hydrolytic ribonuclease J (MG139) and phosphorolytic ribonucleases RsgA (MG110) and DeaD (MG425) cleave the 3' and 5' ends of the 5S and 16S rRNA precursors. *M. genitalium* doesn't contain a homolog of the *E. coli* 9S RNA cleavage enzyme ribonuclease E. The mechanisms of 3' and 5' cleavage of the 5S rRNA precursor are unknown. 3' and 5' cleavage of the 5S rRNA precursor are modeled as a spontaneous process.

sRNA Cleavage

The reconstructed *M. genitalium* small non-coding RNA (sRNA) precursor cleavage scheme illustrated in Figure C.14 was based on that of *E. coli*^{271,281}. ffs (MG0001) is cleaved at both its 3' and 5' ends by ribonuclease III. ribonuclease P cleaves the 5' end of rpnB (MG0003) and ssrA (MG0004).

tRNA Cleavage

The reconstructed *M. genitalium* tRNA precursor cleavage scheme illustrated in Figure C.14 was based on that of *E. coli*^{271,281}. Ribonucleases III (MG367) and P (MG0003, MG465) hydrolytically cleave the 3' and 5' ends of each pre-tRNA, removing the 3' and 5' leader regions and intercistronic regions to produce individual tRNA.

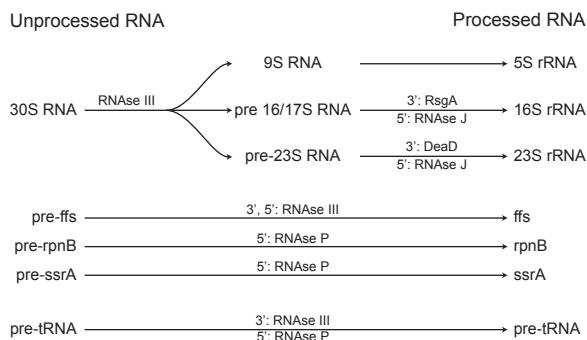


Figure C.14. Non-coding RNA cleavage.

Cleavage Reactions

The stoichiometry, kinetics, and energetics of each non-coding RNA cleavage reaction were reconstructed based on extensive review of the primary literature (see Table S3D and S3O).

Computational Representation

Mathematical Model

This process models the cleavage of operonic non-coding RNA into individual gene products and intercistronic RNA fragments. Because the kinetics of RNA cleavage are not well characterized, this process makes several simplifying assumptions. First, this process assumes that each RNA is fully cleaved in a single time step. Consequently, this process collapses the cleavage of each RNA into a single reaction, and only represents uncleaved and fully cleaved RNA. Intermediate cleavage configurations are not represented. Second, this process assumes that the mean cleavage rate, v_i of each RNA species i is independently limited by (1) the copy number of unprocessed RNA, r_i^u , (2) the copy numbers of intracellular metabolites, m_j , and (3) the copy numbers of RNA processing enzymes, e_j . Based on these assumptions, the functional form of v_i is given by

$$v_i = \min \left(\overbrace{r_i^u}^{\text{RNA}}, \overbrace{\left[\min_j \frac{m_j}{M_{ji}^s} \right]}^{\text{metabolites}}, \overbrace{\text{poissonRand} \left(\min_j \frac{e_j}{K_{ji}} \Delta t \right)}^{\text{enzymes}} \right), \quad (\text{C.29})$$

where M_{ji} is the stoichiometry of metabolite j in the processing of RNA species i including NTP hydrolysis coupled to phosphorolytic cleavage, $M^s = \max(0, -M)$ is the negative part of M , K_{ji} is the experimentally observed catalytic rate of enzyme j in the processing of RNA species i , and $\Delta t = 1 \text{ s}$ is the simulation time step.

This process implements a stochastic model of the arrival of RNA processing events with relative rates v_i . Until RNA, metabolic, and/or enzymatic resources are exhausted, the model iteratively (1) computes the arrival rate, v_i , of each processing event, (2) selects a single processing event to execute according to a multinomial distribution parameterized by v_i , and (3) executes the selected processing reaction, updating the copy numbers of RNA and metabolites and decrementing the available enzymatic capacity. Algorithm C.24 outlines the implementation of the RNA processing model.

Integration

The **Rna** state represents the copy numbers of unprocessed, processed, and intercistronic RNA. The **Metabolite** state represents the copy number of each intracellular metabolite. The **Protein Monomer** and **Protein Complex** states represent the copy number of each RNA processing enzyme.

RNA are synthesized and matured in four steps (see Section C.2.12). This process models the cleavage of RNA transcripts produced by the **Transcription** process. The **RNA Modification** process models the modification of transcripts cleaved by this process. The **Macromolecular Complexation** and **Ribosome Assembly** processes model the formation of macromolecular complexes, including the 30S and 50S ribosomal particles. The **Translation** process models the function of m-, r-, s-, and tRNA in translation. The **RNA Decay** process models the degradation of cleaved intercistronic RNA fragments.

Initial Conditions

Section C.1.4 outlines the cell state initialization algorithm. Briefly, after the **Mass** state initializes the total cell mass, the **Rna** state initializes the total copy number of each RNA species and initializes all RNA to their mature – processed and modified – configuration. Second, the **tRNA Aminoacylation** process initializes tRNA to the aminoacylated configuration. Next, the **Macromolecular Complexation** and **Ribosome Assembly** processes initialize the copy number of each ribonucleoprotein complex. Finally, the **Translation** process initializes the mRNA codon location of each 70S ribosome.

Dynamic Computation

Algorithm C.24 outlines the implementation of the RNA processing model.

Fitting

The expression of the RNA processing enzymes was fit to provide sufficient enzymes to quickly process RNA, or more specifically to prevent sustained accumulation of unprocessed RNA. See Section C.1.3 for further discussion.

Algorithm C.24. RNA processing simulation. See Mathematical Model section above for definition of the mathematical notation.

Input: r_i^p copy number of processed RNA species i
Input: r_i^i copy number of intercistronic fragment i
Input: R_{ji}^g is one if operonic RNA i contains gene j , and zero otherwise
Input: R_{ji}^i is one if operonic RNA i contains intercistronic fragment j , and zero otherwise

Let $k_i \leftarrow e_i \Delta t$ be the capacity of enzyme i for RNA processing

repeat

- Calculate cleavage rates
- foreach** operonic non-coding RNA species i **do**
- | Calculate v_i according to Eq. C.29
- | Select operonic non-coding RNA species $i \sim \text{multinomialRand}(1, v_i / \sum_j v_j)$
- | Update RNA copy numbers: $r_i^u \leftarrow r_i^u - 1, r^p \leftarrow r^p + R_{\bullet i}^g, r^i \leftarrow r^i + R_{\bullet i}^i$
- | Update metabolites: $m \leftarrow m - M_{\bullet i}$
- | Update enzyme catalytic capacity: $k \leftarrow k - K_{\bullet i}$

until no further cleavage possible ($v_i = 0 \forall i$)

C.3.24 Terminal Organelle Assembly

Biology

Balish and Krause have shown that *M. genitalium* maintains a flask shape with a single 300×80 nm membrane-bound bleb or terminal attachment organelle throughout most of its life cycle¹⁸. Krause and Balish have also shown that the *M. genitalium* terminal organelle divides during cell division, producing a daughter organelle which subsequently migrates to the opposite pole¹⁹⁷. The *M. genitalium* terminal organelle been associated with several cellular processes including motility, adhesion, replication, and cytokinesis^{17,18,60,197,198,308–310,324}. This process models the assembly of the protein content of the terminal organelle.

Reconstruction

Krause and Balish have shown that the terminal organelle is composed of eight proteins – high molecular weight cytadherence accessory proteins (HMW) 1-3 (MG312, MG218, MG317), adhesins MgPa (MG191), P32 (MG318) and P65 (MG217), and proteins P110 (MG192) and P200 (MG386)^{17,60,197,198}. The terminal organelle protein content is believed to assemble in the stereotyped hierarchical pattern illustrated in Figure C.15^{197,198}. First, HMW1 and HMW2 mutually recruit each other into the terminal organelle. Second, HMW1 recruits MgPa, HMW3, and P200 into the terminal organelle. Third, HMW3 recruits P32 which in turn recruits P65. P110 independently localizes to the terminal

organelle. The kinetics of terminal organelle assembly are unknown.

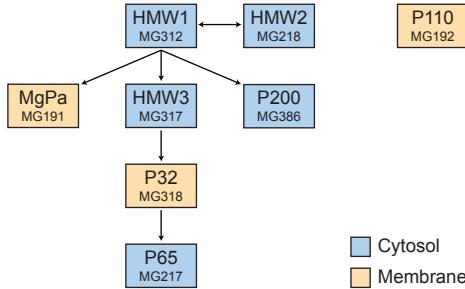


Figure C.15. Hierarchical assembly of the *M. genitalium* terminal organelle.

Computational Representation

Mathematical Model

This process implements a Boolean model of the observed hierarchical assembly of the protein content of the *M. genitalium* terminal organelle. Figure C.15 outlines the Boolean assembly model. Algorithm C.25 describes the model implementation in detail.

Integration

The **Protein Monomer** state represents the copy number of each protein species in each of four compartments: cytosol (c), membrane (m), and terminal organelle cytosol (tc) and membrane (tm). The **Host** state represents the status of the host urogenital epithelium to which the *M. genitalium* terminal organelle attaches.

Several processes including **Translation** model the synthesis and maturation of protein monomers (see Section C.2.10). The **Host Interaction** process models the role of the terminal organelle in host attachment and immune activation.

Initial Conditions

After the total cell mass is initialized, the **Protein Monomer** state initializes the terminal organelle copy number of each terminal organelle protein.

Dynamic Computation

Algorithm C.25 outlines the implementation of the Boolean terminal organelle assembly model.

Algorithm C.25. Terminal organelle assembly simulation.

```

if  $HMW1$  and  $HMW2$  expressed ( $HMW1_c + HMW1_{tc} > 0$  and  $HMW2_c + HMW2_{tc} > 0$ ) then
    Localize  $HMW1$  and  $HMW2$  to the terminal organelle
     $HMW1_{tc} \leftarrow HMW1_{tc} + HMW1_c$ 
     $HMW2_{tc} \leftarrow HMW2_{tc} + HMW2_c$ 
     $HMW1_c \leftarrow 0$ 
     $HMW2_c \leftarrow 0$ 

if  $HMW1$  localized to terminal organelle ( $HMW1_{tc} > 0$ ) then
    Localize  $HMW3$ ,  $MgPa$ ,  $P200$  to terminal organelle
     $HMW3_{tc} \leftarrow HMW3_{tc} + HMW3_c$ 
     $MgPa_{tm} \leftarrow MgPa_{tm} + MgPa_m$ 
     $P200_{tc} \leftarrow P200_{tc} + P200_c$ 
     $HMW3_c \leftarrow 0$ 
     $MgPa_m \leftarrow 0$ 
     $P200_c \leftarrow 0$ 

if  $HMW3$  localized to terminal organelle ( $HMW3_{tc} > 0$ ) then
    Localize  $P32$  to terminal organelle
     $P32_{tm} \leftarrow P32_{tm} + P32_m$ 
     $P32_m \leftarrow 0$ 

if  $P32$  localized to terminal organelle ( $P32_{tm} > 0$ ) then
    Localize  $P65$  to terminal organelle
     $P65_{tc} \leftarrow P65_{tc} + P65_c$ 
     $P65_c \leftarrow 0$ 

Localize  $P110$  to terminal organelle
 $P110_{tm} \leftarrow P110_{tm} + P110_m$ 
 $P110_m \leftarrow 0$ 

```

C.3.25 Transcription

Biology

Transcription is the first step in the synthesis of functional gene products where RNA polymerase and several accessory enzymes translate transcription units (regions of the DNA containing 1 or more genes) into RNA molecules. An RNA polymerase can bind on and off of the DNA either specifically to a gene promoter or non-specifically to a non-promoter site²⁴². Transcription begins with the recruitment of RNA polymerase to a gene promoter with the help of a sigma initiation factor and possibly transcription factors. Next elongation factors are recruited, RNA begins to be polymerized, and the sigma factor is released. In this stage, the RNA polymerase is said to be in

the actively transcribing state. Finally the RNA polymerase reaches a terminator at the end of the transcription unit, and with the help of termination factors releases the polymerized RNA and dissociates from the DNA.

Termination of transcription in some bacteria can require the hexameric ATP-dependent helicase Rho⁶⁷. However, Rho is not essential in *B. subtilis* (gram positive) or *M. genitalium*⁴²⁷, and therefore we chose to only include Rho-independent termination in our model. Rho-independent termination occurs via the intrinsic properties of RNA which disrupt RNA polymerase-DNA binding.

As soon as the RNA begins to polymerize, even prior to termination, the mRNA transcripts may be bound by ribosomes and translated to polypeptides. For simplicity, our model doesn't represent this phenomenon, allowing translation only of completed mRNAs.

Reconstruction

Enzymes

All of the enzymes required for transcription are detailed in Table C.22.

Table C.22. Enzymes and complexes used in the Transcription process class.

Enzymes/Complexes	Composition	Gene	DNA
		Name(s)	Footprint (bases)
RNA polymerase sigma factor	(1) MG249	rpoD	58
Transcription elongation factor	(1) MG282	greA	
Transcription termination factor	(1) MG141	nusA	
Transcription termination/antitermination protein	(1) MG027	nusB	
DNA-directed RNA Polymerase	(1) MG022, (2) MG177, (1) MG340, (1) MG341	rpoE, rpoA, rpoC, rpoB	75

Parameters

The parameters required for transcription were derived from multiples sources including microarray gene expression data and RNA half life data. The data used and calculations performed pre-fitting, are described below:

mRNA Gene expression data

Gene expression data for *M. genitalium* was unavailable, so we used data (measured at 37°C) from *Mycoplasma pneumoniae*, the closest phylogenetic relative of *M. genitalium*⁴³¹. The microarray data by Weiner III et al. was presented in normalized log form. Since the exact method of normalization was unknown to us, to re-derive expression levels we simply calculated $2^{\wedge}(\text{presented value})$. The *M. genitalium* genome is contained in the larger *M. pneumoniae* genome and we were able to map a *M. pneumoniae* gene to all but 6 *M. genitalium* genes²⁶. The genes were mapped based on matching gene names, matching gene descriptions, and Bio-cyc's Align Multi-Genome Browser¹⁸⁸.

tRNA expression data

We also require tRNA expression estimates analogous to the mRNA microarray data. Since no microarray data is available, we use cell composition data to approximate tRNA expression²⁶¹. We approximate the total tRNA expression as the total mRNA expression multiplied by the ratio of the tRNA to mRNA weight fractions of the cell. Next we need to approximate the expression of each individual tRNA. We use measurements of the relative abundances of each amino acid in the cell. A tRNA's expression is the total tRNA expression multiplied by the fraction of the total amino acid weight represented by its amino acid^{304,364}. For example, alanine accounts for about 8% of the total amino acid mass. Then the expression of tRNA for alanine, MG471, is 8% of the total tRNA expression. In cases of degeneracy, where multiple tRNAs bind to the same amino acid, the amino acid weight fractions is split evenly between the tRNAs.

rRNA and sRNA expression data

Similar to the tRNA expression calculations, the total rRNA expression is calculated as the total mRNA expression multiplied by the ratio of the rRNA to mRNA weight fractions of the cell. This total rRNA expression is split between the three rRNA species (23S, 16S, 5S) based on their relative abundances in the cell. sRNA expression is also calculated in the same way, such that expression is proportional to RNA abundance in the cell.

Half-life data

We obtained mRNA half-lives from measurements in *E. coli* performed at 30°C in M9 minimal media²⁹. Additional sets of *E. coli* half-life data are available, but we chose the set with the most comprehensive list of genes mapping to homologous *M. genitalium* genes, and the set whose average half-life was closest to the reported bulk *E. coli* mRNA half-life^{328,353}. The gene mapping between *E.*

coli and *M. genitalium* was based on common gene names and annotations, Bio-Cyc’s Align Multi-Genome Browser, and UniProt^{71,188,304}. This have us a half-life estimate for 274 *M. genitalium* genes. For the remaining genes, we assigned the average half-life, 4.425 minutes. We used separate sources to acquire the half-lives of tRNAs (45 minutes), rRNAs (150 minutes), and sRNAs (89 minutes)^{15,184}.

Gene assignment to transcription units

The transcription unit structure (sets of genes that are transcribed together following a single promoter binding event) was compiled from several sources including primary reports^{101,199,266,268,363,394,405,430} and databases^{301,304} of cotranscribed genes, and a computational model¹⁵⁸that predicts promoter and transcription unit start sites. We also predicted transcription units using information on the conservation of gene order across multiple species, the related functions of adjacent genes, the similar expression levels of adjacent genes as measured by microarrays, and the strandedness (transcription direction) of adjacent genes⁴³¹. We transcribe the 525 *M. genitalium* genes in 355 transcription units. 294 genes fall in transcription units of more than one gene. The longest transcription unit contains 4 genes.

Rate at which each transcript is produced

The rate of change of the concentration of an RNA_i , is its synthesis net its degradation:

$$\frac{d[RNA_i]}{dt} = \underbrace{k_{synth,i}}_{\text{synthesis}} - \underbrace{k_{deg,i}[RNA_i]}_{\text{degradation}}, \quad (\text{C.30})$$

where $k_{synth,i}$ is the rate of synthesis and $k_{deg,i}$ is the rate of degradation of RNA_i

The degradation rate is described by the first-order degradation constant of RNA_i with half-life, h_i (obtained as described above).

$$k_{deg,i} = \frac{\ln(2)}{h_i} \quad (\text{C.31})$$

At steady state RNA concentration, $\frac{d[RNA]}{dt} = 0$:

$$k_{synth,i} = \frac{\ln(2)}{h_i} [RNA_i]_{SS} \quad (\text{C.32})$$

The relative expression, E_i , as described above, was obtained from microarray and cell composition

data.

Substituting $[RNA_i]_{SS}$ with $E_i[RNA]$, where the relative expression, E_i , as described above, was obtained from microarray and cell composition data, and $[RNA]$ is the total RNA concentration in the cell, we get the synthesis rate of RNA_i :

$$k_{synth,i} = \frac{\ln(2)}{h_i} E_i [RNA] \quad (C.33)$$

Rate at which each transcript is produced (assuming exponential growth)

We adjust the above derivation for growth and dilution. Assuming exponential growth of the cell:

$$k_{synth,i} = k_{synth,iinit} e^{\ln(2) \frac{t}{\tau}} \quad (C.34)$$

$$[RNA_i] = [RNA_i]_{init} e^{\ln(2) \frac{t}{\tau}} \quad (C.35)$$

where τ is the cell cycle length,

$$\frac{d[RNA_i]}{dt} = \underbrace{k_{synth,iinit} e^{\ln(2) \frac{t}{\tau}}}_{\text{degradation}} - \underbrace{k_{deg,i} [RNA_i]_{init} e^{\ln(2) \frac{t}{\tau}}}_{\text{degradation}} \quad (C.36)$$

Assuming that mother and daughter cells are identically distributed:

$$[RNA_i]_{t=\tau} = 2[RNA_i]_{init} \quad (C.37)$$

$$[RNA_i]_{init} = \int_0^\tau \frac{d[RNA_i]}{dt} \quad (C.38)$$

$$= (k_{synth,iinit} - k_{deg,i} [RNA_i]_{init}) \frac{\ln(2)}{\tau} \quad (C.39)$$

$$k_{synth,iinit} = [RNA_i]_{init} \left(\frac{\tau}{\ln(2)} + k_{deg,i} \right) \quad (C.40)$$

Again substituting $[RNA_i]$ with $E_i[RNA]$:

$$k_{synth,iinit} = E_i[RNA]_{init} \left(\frac{\tau}{\ln(2)} + k_{deg,i} \right) \quad (C.41)$$

Synthesis Rate Adjustment for Transcription Units

Genes that are transcribed together in a transcription unit must have the same synthesis rate. This is because we do not model incomplete transcription of a transcription unit, where the RNA polymerase can fall off the unit, after transcribing a subset of its genes. For half-lives that have not been measured in *E. coli* and were set to an average 4.425 minutes, we adjusted the half-lives to even out the synthesis rates within a transcription unit. For the other cases, we set the synthesis rate of the transcription unit to be the average of the rates within the transcription unit.

Determining the probabilities of making transcripts

The probability of an RNA polymerase binding a given transcription unit, i , is based on the synthesis rate of transcription unit i relative to the synthesis rate of all the other transcription units:

$$\text{Probability(transcribing } i) = P_{tu,i} \quad (C.42)$$

$$= \frac{k_{synth,i}}{\sum_{j=1:335} k_{synth,ij}} \quad (C.43)$$

Substituting in the synthesis rate from above:

$$P_{tu,i} = \frac{E_i[RNA]_{init}(\frac{\tau}{\ln(2)} + k_{deg,i})}{\sum_{j=1:335} E_i[RNA]_{init}(\frac{\tau}{\ln(2)} + k_{deg,i})} \quad (C.44)$$

$$= \frac{E_i \left(\frac{\tau}{\ln(2)} + k_{deg,i} \right)}{\sum_{j=1:335} E_i \left(\frac{\tau}{\ln(2)} + k_{deg,i} \right)} \quad (C.45)$$

Parameter Fitting

After and all of the raw data has been transformed according to the calculations described above, and fit to assure growth of a wildtype cell that will accommodate doubling in 9hours, we reach the parameters described in Table C.23.

Table C.23. Fixed parameters used in the Transcription process class.

Parameters	Value	Symbol	Source
RNA polymerase elongation rate	50 nt s ⁻¹	k_{el}	94, 421.
Transcription unit binding probabilities		P_{tu}	See 'Data and Calculations'
RNA polymerase state transition probabilities		P_{trans}	242.
Enzyme DNA footprints	See 'Enzymes/complexes'		264, 322.
Transcription unit sequences	See Table S3K, 304.		304.
Transcription unit directions	See Table S3K		304.
Transcription unit lengths	See Table S3K		304.
Transcription unit 5' coordinates	See Table S3K		304.
Cell Cycle Length	8.9 h	τ	Experimentally measured. See Experimental Methods.

Computational Representation

We use a Markov model to determine the state of each RNA polymerase. An RNA polymerase may exist in any of the four following states:

1. Free (not bound to a chromosome) (FS)
2. Non-specifically bound somewhere on a chromosome (NSB)
3. Specifically bound to the promoter of a transcription unit (SB)
4. Actively transcribing a transcription unit (AT)

Transition probabilities between these states are designed to maintain the occupancy of each state within a narrow window around their expected values²⁴². The transition probabilities (P_{trans}) are determined by four logistic control functions, tuned by the RNA polymerase state expectations.

Free State

All newly created or released RNA polymerases are in the free state. From the free state, a polymerase can transition to the non-specifically bound state, transition to the specifically bound state,

or remain in the free state.

Non-specifically bound state

From the non-specifically bound state, a polymerase can transition to the free state, transition to the specifically bound state, or remain in the non-specifically bound state. A random position on a chromosome for the non-specifically bound polymerase is selected from the polymerase accessible sites as determined by the Chromosome state.

Specifically bound state

From the specifically bound state, a polymerase can transition to the free state, non-specifically bound state, or actively transcribing state, or remain in the specifically bound state. A polymerase can only transition into the specifically bound state if a free sigma factor is available. Upon transition a polymerase into this state, we randomly pick a transcription unit to bind to. The probability of binding each transcription unit is based on experimentally measured gene expression and half life data, as detailed below.

Actively transcribing state

Once in the actively transcribing state, a polymerase will remain in the actively transcribing state until transcription is terminated, the RNA polymerase is displaced by another protein, or the RNA polymerase falls off due to a stall. In all these cases, the RNA polymerase falls off into the free state. For a polymerase in the actively transcribing state, we release any bound sigma factors and elongate the transcript according to nucleic acid limits (given the transcript sequence) and the elongation rate k_{el} . The transcript and polymerase are released if transcription is complete and the necessary termination factors are available.

Integration

Table C.24. State classes connected to the Transcription process class.

Connected States	Read from state	Written to state
Chromosome	<ul style="list-style-type: none"> Regions accessible for transcription machinery to bind Polymerized regions of DNA 	<ul style="list-style-type: none"> Positions and strands on the DNA where transcription machinery is bound
Rna	<ul style="list-style-type: none"> Counts of RNA species 	<ul style="list-style-type: none"> Updated counts of RNA species
RNA Polymerase	<ul style="list-style-type: none"> RNA polymerase states (FS, NSB, SB, AT) DNA positions and strands of DNA bound RNA polymerases Expectations of RNA polymerases in FS, NSB, SB, and AT states (used to derive transition Probabilities P_{trans}) 	<ul style="list-style-type: none"> Updated states of RNA polymerases Updated positions and strands of DNA bound RNA polymerases
Transcript	<ul style="list-style-type: none"> Transcription unit indexes of bound RNA polymerases RNA polymerase position within transcript Chromosome on which RNA polymerase resides (1 or 2) Transcription unit sequences, directions, lengths, and 5' start coordinates 	<ul style="list-style-type: none"> Updated transcription unit indexes of bound RNA polymerases Updated RNA polymerase position

Initial Conditions

All RNAs are initialized to the mature state. RNA polymerases are initialized as follows:

1. Each RNA polymerase is randomly assigned (with replacement) to one of the actively transcribing, specifically bound, non-specifically bound, or free states weighted by the expected occupancy of each state

2. Actively transcribing and specifically bound polymerases are randomly assigned to transcription units weighted by the transcription unit binding probabilities (P_{tu}).
3. Each transcription unit to which one or more actively transcribing polymerases have been assigned is divided into 1 segment for each polymerase
4. Actively transcribing polymerases are randomly assigned to positions within the assigned segment of their assigned transcription unit (positions near the segment border are not allowed to prevent polymerases from being too close to each other) with uniform probability.
5. Non-specifically bound polymerases are randomly assigned to an accessible region on the chromosome.

Dynamic Computation

At each timestep, we follow the following algorithm:

1. Assign all newly synthesized RNA polymerases to the free state
2. Use the Markov model to randomly transition RNA polymerases among the FS, NSB, SB, and AT states, weighted by the state transition probabilities P_{trans} .
3. Randomly assign RNA polymerases entering the NSB state to an accessible position on the Chromosome (performed by Chromosome state).
4. Randomly assign RNA polymerases entering the SB state to specific transcription units weighted by the product of the transcription unit binding probabilities (P_{tu}) and the binding probability fold changes. (Fold changes arise from other cellular processes and are described in the **Transcriptional Regulation** and **DNA Supercoiling** process classes.) We determine whether the selected promoter site is accessible and bind the polymerase to a chromosome using the **Chromosome** state class. RNA polymerase specific binding can only occur if there is an available sigma factor, and a sigma factor is accordingly decremented.
5. Simulate RNA polymerization by actively transcribing RNA polymerases with the aid of elongation factors.
 - (a) Release the sigma factor if this is the first second of elongation, and increment the free sigma factor count
 - (b) If all of the necessary elongation factors are available, elongate the transcript according to nucleic acid limits (given the transcript sequence) and the elongation rate k_{el} . We allocate available nucleic acids among the actively polymerizing RNA polymerases, by adding a base to each elongating transcript before moving to the next base of a given

transcript. Polymerization can also be limited if the new site at which the polymerase will land is not accessible (for example, if another RNA polymerase lies in its way)

6. If transcription is complete and the necessary termination factors are available,
 - (a) Release the completed transcript
 - (b) Transition the RNA polymerase to the free state
 - (c) Increment the RNA count

C.3.26 Transcriptional Regulation

Biology

Transcription factors are one of many mechanisms cells employ to respond to external signals and maintain homeostasis. Transcription factors regulate the synthesis rates of RNA by modulating the affinity of RNA polymerase for promoters. Transcriptional enhancers stabilize RNA polymerase-promoter complexes by contributing negative free energy to the complex, for example by providing additional surfaces for RNA polymerase binding. Transcriptional repressors destabilize RNA polymerase-promoter complexes, for example by sterically blocking promoters. This process models the binding of transcriptional regulators to promoters and the fold-change effect of transcriptional regulators on the affinity of RNA polymerase for individual promoters.

Reconstruction

The *M. genitalium* transcriptional regulatory network was reconstructed based on an extensive review of the primary literature^{9,14,61,64,114,140,266,273,275,279,293,297,312,430,449} and the proteomic database DBTBS³⁶². *M. genitalium* has few homologs to reported transcription factors. The reconstructed network contains five regulators which regulate 54 genes through 29 regulatory interactions, including one regulator (Spx, MG127) which interacts directly with RNA polymerase. Table S3P lists the five reconstructed transcriptional regulators and the binding site, motif, and affinity, and fold-change effect of each regulatory interaction.

Computational Representation

Mathematical Model

This process models the binding of transcriptional regulators to promoters and the fold-change effect of transcriptional regulators on RNA polymerase-promoter binding. Because *M. genitalium* transcriptional regulation is not well characterized, this process makes several simplifying assumptions. First, this process assumes that transcriptional regulator-DNA binding is kinetically fast and energetically favorable, and therefore proceeds to completion within the 1 s simulation time step. Second, because transcriptional regulator-promoter affinities have not been systematically characterized, this process assumes that transcriptional regulators bind promoters with affinity proportional to their fold change effect. Third, this process assumes that only one copy of each transcriptional regulator can bind at each promoter. Fourth, this process assumes that transcriptional regulators stably bind DNA. Consequently, transcriptional regulator dissociation is ignored except displacement by other DNA-binding proteins which is modeled by the **Transcription** and **Replication** processes. Finally, this process assumes that transcriptional regulators independently affect RNA polymerase, and thus their fold change effects add multiplicatively. Algorithm C.26 outlines the implementation of the transcriptional regulation model.

Integration

The **Protein Monomer** and **Protein Complex** states represent the free and DNA-bound copy numbers of each transcriptional regulator. The **Chromosome** state represents the exact chromosomal location of each DNA-bound transcriptional regulator. The **RNA Polymerase** state represents the fold change effect of transcriptional regulation on the affinity of RNA polymerase for each promoter.

Several processes including **Translation** model protein synthesis (see Section C.2.10). The **Transcription** process models (1) transcription initiation: RNA polymerase-promoter binding including the fold-change effects of transcriptional regulation, (2) transcript elongation, and 3) transcription termination.

Initial Conditions

Section C.1.4 outlines the cell state initialization algorithm. Briefly, after the **Protein Monomer** and **Protein Complex** states initialize the total copy number of each transcriptional regulator, the **Transcriptional Regulation** process initializes the status – free or DNA-bound – and chromosomal location of the transcriptional regulators to a steady-state of the transcriptional regulatory network by iteratively evaluating Algorithm C.26 until convergence. Because the transcriptional regulatory model assumes that transcriptional regulator-promoter binding proceeds to completion within the 1 s simulation time step and is stable, Algorithm C.26 converges in one iteration.

Dynamic Computation

Algorithm C.26 outlines the implementation of the **Transcriptional Regulation** model.

Algorithm C.26. Transcriptional regulation simulation.

Input: n_{ik} is true if promoter i is expressed in chromosome k
Input: $p_{c,i}$ free cytosolic copy number of transcriptional regulator i
Input: $p_{b,i}$ DNA-bound copy number of transcriptional regulator i
Input: x_{ij} Binding site of transcriptional regulator i at promoter j
Input: F_{ij} fold-change effect of transcriptional regulator i on promoter j
Input: b_{ijkl}^m, b_{ijkl}^c chromosomal protein occupancy as defined in Table C.2
Output: f_i fold-change effect of transcriptional regulation on RNA polymerase affinity for promoter i

Calculate the relative rate, r_{ijk} , transcriptional regulator i binds promoter j of chromosome k :

foreach DNA-binding transcriptional regulator i in promoter j of chromosome $k = \{1..2\}$ **do**

- $\quad r_{ijk} \leftarrow n_{jk} p_{c,i} F_{ij}$

Bind transcriptional regulators to the chromosome:

repeat

- \quad Select regulator i , promoter j , and chromosome $k \sim \text{multinomialRand}(1, r_{ijk} / \sum_{ijk} r_{ijk})$
- \quad **if** regulator i expressed ($p_{c,i} > 0$) and isRegionAccessible(promoter j of chromosome k to regulator i)
- \quad **then**

 - $\quad \quad$ Bind protein to chromosome: $b_{y \bullet ki}^z = 1 \forall y \in \{x_{ij}..x_{ij} + l_i - 1\}$, where $z = m$ for monomers and c for complexes
 - $\quad \quad$ Update free and bound copy numbers: $p_{c,i} \leftarrow p_{c,i} - 1$, $p_{b,i} \leftarrow p_{b,i} + 1$

- \quad Update binding rate: $r_{ijk} \leftarrow 0$

until no additional transcriptional regulator can bind DNA ($r_{ij} = 0 \forall i, j$)

Calculate the fold-change effect of transcriptional regulators on the affinity of RNA polymerase for each promoter

Initialize fold-change effects: $f_i \leftarrow 1 \forall i$

foreach promoter j of chromosome $k = \{1..2\}$ bound by DNA-binding transcriptional regulator i **do**

- \quad Add fold-change effects multiplicatively: $f_j \leftarrow f_j F_{ij}$

foreach promoter j regulated by an expressed non-DNA-binding transcriptional regulator i ($p_{c,i} > 0$) **do**

- \quad $f_j \leftarrow f_j F_{ij}$

C.3.27 Translation

Biology

Translation is the process whereby the ribosome, accessory enzymes, and tRNAs transcode mRNAs and produce amino acid polymers. Translation begins with the recruitment of the 30S and 50S ribosomal particles and initiation factor 3 (IF3) to an mRNA molecule. Next, the ribosomal particles assemble into a 70S ribosome on the mRNA molecule with the help of initiation factors. Third, the ribosome polymerizes amino acids presented by aminoacylated tRNAs with the help of elongation factors. Finally, a release factor recognizes the stop codon UAG or UAA, hydrolyzes the peptidyl tRNA bond, and dissociates. A ribosome recycling factor dissociates the E-site tRNA, an elongation factor G releases the release factor and 50S ribosome, and an initiation factor 3 dissociates the 30S ribosome, P-site tRNA, and mRNA²⁷⁸.

A ribosome may stall for various reasons including collisions with other proteins and limited resources required of translation. When a ribosome stalls, its last tRNA is expelled and replaced by the tRNA-like domain of a tmRNA molecule. Next, the mRNA-like domain of the tmRNA expels the bound mRNA. Third, the ribosome resumes polymerization, now using the tmRNA's mRNA-like domain as its template. This results in the production of an amino acid polymer containing a C-terminal proteolysis tag. Finally, the proteolysis tag will be recognized by the protein degradation machinery, and the amino acid polymer will be degraded into its individual component amino acids³⁰⁵.

Reconstruction

All of the enzymes required for translation are detailed in Table C.25, and all of the parameters are detailed in Table C.26.

Table C.25. Enzymes and complexes used in the Translation process class.

Enzymes/Complexes	Composition	Gene Name(s)
Initiation factor IF-1	(1) MG173	infA
Initiation factor IF-2	(1) MG142	infB
Initiation factor IF-3	(1) MG196	infC
Elongation factor G	(2) MG089	fusA
Elongation factor P	(1) MG026	efp
Elongation factor Tu	(2) MG451	tuf
Elongation factor Ts	(2) MG433	tsf
Peptide chain release factor 1	(1) MG258	prfA
Ribosome recycling factor	(1) MG435	frr
30S ribosomal subunit	(1 each) MGrrna16S, MG070, MG087, MG088, MG090, MG092, MG150, MG155, MG157, MG160, MG164, MG165, MG168, MG175, MG176, MG311, MG417, MG424, MG446, MG481, MG522 (1) Ribosome_30S, (1) MG196	16S rRNA, rpsB, rpsL, rpsG, -, rpsR, rpsJ, rpsS, rpsC, rpsQ, rpsN, rpsH, rpsE, rpsM, rpsK, rpsD,, rpsI, , rpsO, , rpsP, -, -
30S ribosomal subunit - initiation factor IF-3 complex		30S ribosomal subunit, infC
50S ribosomal subunit	(1 each) MGrrna23S, MGrrna5S, MG081, MG082, MG093, MG151, MG152, MG153, MG154, MG156, MG158, MG159, MG161, MG162, MG163, MG166, MG1676, MG169, MG174, MG178, MG197, MG198, MG257, MG325, MG361, MG362, MG363, MG418, MG426, MG444, MG466, MG473 (1) Ribosome_50S, (1) Ribosome_30S	23S rRNA, 5S rRNA, rplK, rplA, -, rplC, rplD, rplW, rplB, rplV, rplP, rplC, rplN, rplX, rplE, rplF, rplR, rplO, rpmJ, rplQ, rpmI, rplT, rpmE, rpmG, -, rplL, rpmF, rplM, rpmB, rplS, rpL34, rpmG-2
70S ribosome		50S ribosomal subunit, 30S ribosomal subunit
tmRNA, MCS6 (10sa RNA)	(1) MG0004	ssrA
SsrA binding protein	(1) MG059	smpB
peptidyl-tRNA hydrolase	(1) MG083	pth

Table C.26. Fixed parameters used in the Translation process class.

Parameter	Value	Symbol	Source
Ribosome elongation rate (amino acids/second)	16	k_{elong}	443.
Probability that stationary ribosome is moved to a stalled state (mRNA re- placed by tmRNA)	1×10^{-6}	P_{stalled}	See “Parameter Fitting”
tRNA sequences of the pro- tein monomers			Determined from genome sequence, tRNA code 363.
tRNA sequences of the pro- teolysis tags			Determined from genome sequence, tRNA code 363.

Parameter Assignment

Most of the parameters required for this process class were obtained from the literature or derived from the *M. genitalium* genome sequence. The probability that a ribosome stalled is an uncharacterized value, and therefore we set this probability to be very low, such that ribosome stalling is a rare event which does not typically limit the simulation.

Computational Representation

This process simulates protein translation by ribosomes and accessory initiation, elongation, and termination factors. Ribosomes transition from a free to active state if the necessary factors are present, and bind to an mRNA. The selection of a specific mRNA to bind to a ribosome is random and weighted by mRNA abundances. As the sequence of each gene is known, the codons presented by the mRNAs are translated using aminoacylated tRNAs and elongation factors at a rate of up to 16 amino acids per second (measured by radioactive labeling in *E. coli*)⁴⁴³. Once a stop codon has been reached, translation is terminated and the polypeptide will undergo further modifications by other processes in our simulation. The process class also simulates the identification of stalled ribosomes by the tmRNA, the replacement of the tRNA and mRNA with the tmRNA, and the synthesis of the proteolysis tag encoded by the tmRNA’s mRNA-like domain.

Integration

Table C.27. State classes connected to the Translation process class.

Connected State	Read from state	Written to state
Polypeptide	<ul style="list-style-type: none"> • List of ribosome-bound mRNAs • Lengths of nascent polypeptides (proteolysis tags) • tRNA sequences of protein monomers and proteolysis tags 	<ul style="list-style-type: none"> • Updated list of ribosome-bound mRNAs • Updated lengths of nascent polypeptides (proteolysis tags)
Ribosome	<ul style="list-style-type: none"> • List of ribosome-bound mRNAs • Lengths of nascent polypeptides (proteolysis tags) • State of each ribosome: free, actively translating, or stalled 	<ul style="list-style-type: none"> • Updated list of ribosome-bound mRNAs • Updated lengths of nascent polypeptides (proteolysis tags) • Updated state of each ribosome: free, actively translating, or stalled
Rna	<ul style="list-style-type: none"> • Counts of mRNA, aminoacylated tRNA, and aminoacylated tmRNA species 	<ul style="list-style-type: none"> • Updated counts of aminoacylated tRNA and aminoacylated tmRNA species

Initial Conditions

The simulation begins in a state in which ribosomes are already bound to mRNAs and in the process of elongating. Each ribosome is randomly assigned (without replacement) to an mRNA species, weighted by the current expression of the mRNAs. Then, each ribosome is assigned to positions within the assigned mRNA with uniform probability. No ribosomes are initialized to the stalled state, since the expected probability of stalling is negligible. No tmRNAs are initiated to the bound state.

Dynamic Computation

Initiation

Each ribosome exists in one of two states, free or actively transcribing. Ribosomes are created in the free state and may transition to the actively transcribing state as follows:

- If ribosome factor A is present, initiation factors (IF-1, IF-2, and IF-3) are present, and one unit of energy (GTP) is available
 - Randomly select free ribosomes to initiate up to the limits of ribosome factor A, initiation factors, and GTP
 - Randomly select mRNA species for each initiating ribosome to bind to, weighted by the counts of each mRNA species
 - Update the state of each ribosome and mRNA, and the amount of available substrates

Elongation

Next, we elongate polypeptides. We assume that one of each elongation factor (EF-tu, TS, and G) is sufficient for each ribosome, but require a separate set of factors for each ribosome. Elongation proceeds as follows:

- If elongation factors (EF-tu, TS, and G), aminoacylated tRNAs, and energy (GTP) are available,
 - Randomly select actively transcribing ribosomes to elongate up to the limits of elongation factors. Available amino acids and energy are allocated among actively translating ribosomes.
 - If the ribosome is newly initiated,
 - Release the initiation factors
 - Associate a tRNA for f-methionine to bind the first amino acid, and release a free tRNA
 - Else,
 - Derive the amino acid sequence of the translating gene by converting the genome sequence into a tRNA sequence using the amino acid code
 - Associate aminoacylated tRNAs to bind amino acids to the growing polypeptide up to the aminoacylated tRNA limit, energy limit, and elongation limit, k_{elong} . Release free tRNAs
 - Update the state of each ribosome, the progress of all actively translating ribosomes, and the amount of available substrates

Note, for cases in which translation of a peptide finishes partway through the time step, the elongation factors are released, but only available at the timestep. For simplicity, we do not explicitly model the transition between P and E sites.

Termination

Once all amino acids of a protein have been translated, one release factor, one recycling factor, one elongation factor G, and one GTP molecule are sufficient to terminate the polypeptide, as follows:

- If at release factors, recycling factors, elongation factors G, and energy (GTP) are available,
 - Randomly select completed polypeptide-mRNA-ribosome complexes to dissociate
 - Update the state of each ribosome and mRNA, monomer counts, and the amount of available substrates. The ribosome will be available to bind mRNA at the following iteration

Translation Stalling

The stalling of ribosomes is dealt with as follows:

- If in a timestep an elongating ribosome hasn't advanced,
 - Then with a small probability, $P_{stalled}$,
 - Transition the ribosome to the stalled state
 - Expel the mRNA and replace it with a tmRNA
 - Encode a proteolysis tag and mark the amino acid chain for degradation by the **Protein Decay** process class.

C.3.28 tRNA Aminoacylation

Biology

The **tRNA Aminoacylation** process class simulates the conjugation of amino acids to the tRNAs. tRNAs serve as mediators between the ribosome and the amino acids which form polypeptides. tRNAs are composed of short RNA sequences which recognize specific codons (triplets of bases) on mRNAs. Each tRNA binds to a specific amino acid, and then interacts with a ribosome to deliver this amino acid to an elongating polypeptide chain, according to the mRNA code.

tmRNAs are short RNA structures that add a proteolysis tag to the end of incomplete polypeptides upon ribosomal stalling, signaling the polypeptides for degradation. The **tRNA Aminoacylation** process class also simulates the aminoacylation of the tmRNA which delivers the amino acid alanine to stalled ribosomes.

Reconstruction

These aminoacylation reactions are both enzyme and energy dependent. *M.genitalium* is believed to have 36 tRNA aminoacylation reactions and 1 tmRNA aminoacylation reaction, and all of the reactions require 1 ATP^{364,436}. There are also two tRNA modification reactions that must occur. Since there is no glutaminyl-tRNA synthetase (to add a glutamine to a tRNA), glutamyl-tRNA synthetase first adds a glutamate to tRNA MG502, and then Glu-tRNA Gln amidotransferase converts the glutamate into a glutamine. Also, a methionylformyltransferase is used to add the formyl group onto the methionine on tRNA MG488 (formylmethionine is used as the start codon).

All of the enzymes required for tRNA aminoacylation are detailed in Table C.28, and all of the parameters are detailed in Table C.29.

Table C.28. Enzymes and complexes used in the tRNA Aminoacylation process class.

Enzymes/Complexes	Composition	Gene Name(s)
Alanyl-tRNA synthetase	(4) MG292	alaS
Arginyl-tRNA synthetase	(1) MG378	argS
Aspartyl-tRNA synthetase	(2) MG036	aspS
Asparaginyl-tRNA synthetase	(2) MG113	asnS
Cysteinyl-tRNA synthetase	(1) MG253	cysS
Glutamyl-tRNA synthetase	(1) MG462	gltX
Glycyl-tRNA synthetase	(2) MG251	glyS
Histidyl-tRNA synthetase	(2) MG035	hisS
Isoleucyl-tRNA synthetase	(1) MG345	ileS
Leucyl-tRNA synthetase	(1) MG266	leuS
Lysyl-tRNA synthetase	(2) MG136	lysS
Methionyl-tRNA synthetase	(2) MG021	metG
Phenylalanyl-tRNA synthetase	(2) MG194, (2) MG195	pheS, -
Prolyl-tRNA synthetase	(2) MG283	proS
Seryl-tRNA synthetase	(2) MG005	serS
Threonyl-tRNA synthetase	(2) MG375	thrS
Tryptophanyl-tRNA synthetase	(2) MG126	trpS
Tyrosyl-tRNA synthetase	(2) MG455	tyrS
Valyl-tRNA synthetase	(1) MG334	valS
Glutamyl-tRNA(Gln) amidotransferase	(1) MG098, (1) MG099, (1) MG100	-, -, gatB
Methionyl-tRNA formyltransferase	(1) MG365	

Table C.29. Fixed parameters used in the tRNA Aminoacylation process class.

Parameters	Source
Free metabolites required for each reaction	364, 436.
tRNA aminoacylated by each reaction	364, 436.
Enzymes required to aminoacylate each tRNA	436.
k_{cat} of the catalyzing enzyme of each aminoacylation reaction (s^{-1})	436.

Computational Representation

The **tRNA Aminoacylation** process maximizes the number of aminoacylation reactions (tRNA aminoacylations, tmRNA aminoacylations, and tRNA modifications) up to the limits of available RNAs, enzymes, and metabolites. The enzymatic bounds are calculated from the catalytic turnover constant (k_{cat}) of each enzyme. The required metabolites include, among others, amino acids and ATP. Since multiple reactions require the same metabolites and enzymes, reactions to occur within a given timestep are randomly selected using a probability distribution that is weighted by the abundances of the reaction requirements. That is, the limits of each reaction are calculated assuming that all of the available required resources would be allocated to the given reaction. The probability distribution for selecting given reactions is weighted by these calculated limits. Reactions are performed one by one until insufficient resources exist to perform any additional reactions. Intermediate steps in the aminoacylation of tRNAs are not represented.

Integration

The **tRNA Aminoacylation** process class reads from and writes to the **Rna** state class. It reads in the counts of free and aminoacylated tRNAs and tmRNA, and writes back the updated counts.

Initial Conditions

All of the tRNAs are initialized to an aminoacylated state.

Dynamic Computation

At each timestep,

- For each of the 39 possible reactions (36 tRNA aminoacylations, 1 tmRNA aminoacylation, 1 Glu-Gln amidotransfer, 1 methionylformyltransfer), i , calculate the maximum number of reactions that can occur:

$$\text{Reaction Limit } i = \min \begin{cases} \text{Number of available required metabolites (amino acid, ATP, ...)} \\ \text{Number of available required enzyme} \times k_{cat} \\ \text{Number of free tRNA or tmRNA} \end{cases}$$

- The available ATP, amino acids, and enzyme activities used in the limits calculation in Step

1, may be double counted as multiple reactions may require the same metabolites or enzymes. Therefore, the selection of which reactions occur, within the calculated limits, is randomly determined.

While resources are available:

- (a) Randomly select a reaction to perform, weighing the probability of selecting a given reaction by its calculated reaction limit
- (b) Decrement the used metabolites, enzyme activities, free tRNAs, free tmRNAs
- (c) Increment the produced metabolites, and modified RNA
- (d) Recalculate the reaction limits as in Step 1, based on the new availabilities

C.4 Experimental Procedures

C.4.1 Media Composition

Each liter of SP-4 is comprised of broth base 600 ml (Mycoplasma Broth Base 3.5 g (BD 211458), Bacto Tryptone 10 g (BD 211705), Bacto Peptone 5.3 g (BD 211677), distilled water 598 ml, pH 7.5 by KOH and autoclaved), 20% glucose 25 ml (CalBioChem 346351), CMRL 1066 10X 50 ml (ATCC20-2207), 7.5% sodium bicarbonate 5 ml (EMDSX-0320-1), 200 mM L-glutamine 5 ml, 2% yeast extract solution 35 ml (BD 210933), 2% autoclaved TC Yeastolate 100 ml (BD 255772), fetal bovine serum albumin (heat inactivated at 55°C for 2 hours) 170 ml (Gibco 26140-079), penicillin G (100,000 U ml⁻¹) 2.5 ml (Sigma P7794), and 0.5% phenol red 4 ml (Sigma P0290).

C.4.2 Frozen Stocks

Cells were harvested for storage as a frozen stock when the media in the 10 cm petri dish cultures was yellow (pH 6.3-6.7). The media from the 10 cm plate cultures was aspirated. Cells were collected by scraping the bottom of the plates, resuspended in 3 ml of FBS, and serial filtered through 1.2, 0.8, 0.45, and 0.2 µm polyethersulfone filters to sterilize and de-clump the cells. Stocks were stored at -80°C.

C.4.3 Colorimetric Growth Assay Serial Dilutions

Cells were harvested for serial dilutions when the media in the 10 cm petri dish culture was yellow (pH 6.3-6.7), and were harvested as described above for frozen stocks. The filtered suspension was

used to make a serial dilution plate. The initial solution in the serial dilution plate is comprised of 50 μ l of culture in 450 μ l of SP-4. (For strains that were especially difficult to culture, 100 μ l culture was added to 400 μ l SP-4.) From this concentration we made three 5-fold serial dilutions. Dilutions were performed in triplicate for each culture, and 200 μ l of each diluted sample was plated in a 96-well plate. The 96-well plates were stored at 37°C and 5% CO₂. Optical density readings were taken at 550 nm twice a day and used to make growth curves.

C.4.4 Colorimetric Growth Assay Calculations

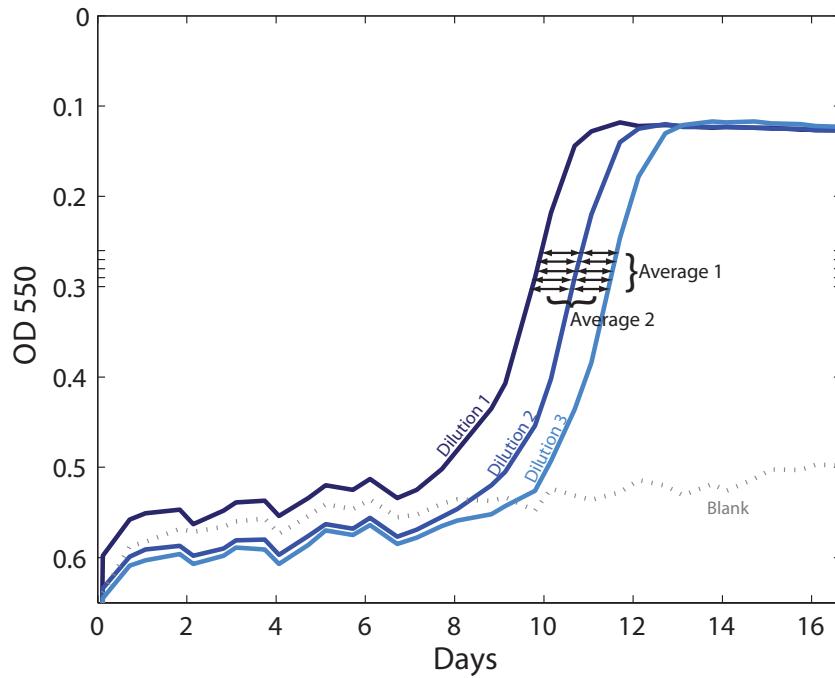


Figure C.16. Comparison of consecutive dilutions.

The growth rate constant and doubling time for each strain was calculated using the growth curves of serially diluted bacterial cultures measured by the colorimetric assay and the serial dilution factor.

Assuming exponential growth, the cell concentration (C) in dilution1 can be defined as:

$$C_{\text{dilution1}, \text{time}_x} = C_{\text{dilution1}, \text{time}_0} \exp(\text{growth rate} \times \text{time}_x) \quad (\text{C.46})$$

And the concentration of dilution2 as:

$$C_{\text{dilution2, time}_y} = C_{\text{dilution2, time}_0} \exp(\text{growth rate} \times \text{time}_y) \quad (\text{C.47})$$

Because the concentration of a dilution at the start of the experiment is five times less than that of the previous dilution:

$$C_{\text{dilution1, time}_0} = 5C_{\text{dilution2, time}_0} \quad (\text{C.48})$$

Furthermore, the two dilutions reach a given OD550 at different times x and y (see horizontal arrows in Figure C.16):

$$C_{\text{dilution1, time}_x} = C_{\text{dilution2, time}_y} \quad (\text{C.49})$$

(This calculation was done at 5 OD550 values: 0.26, 0.27, 0.28, 0.29, and 0.30)

By substitution, the relation for growth rate constant (for a specific pair of cultures and OD500) and is defined:

$$\text{growth rate constant} = \frac{\ln(5)}{\text{time}_y - \text{time}_x} \quad (\text{C.50})$$

The doubling time is then calculated as:

$$\text{doubling time} = \frac{\ln(2)}{\text{growth rate constant}} \quad (\text{C.51})$$

C.4.5 Quantitative PCR to Measure Cell Growth

Some strains grew so slowly that the colorimetric assay was inadequate to determine a growth rate. In these cases, a DNA quantification method was used to estimate the growth rate. This method calculates the number of chromosomes in a sample, and assumes that changes in chromosome count is correlated with changes in the *M. genitalium* cell count. The strains were cultured in 4 mL SP-4 and incubated at 37°C with 5% CO₂. Three replicate cultures were made for each of 14 days of

harvest, resulting in 42 cultures per strain. Each day, cells were harvested by scraping the bottom of the plate, spun down at 4575 xg for 15 minutes, and resuspended in 20–400 μ l of TE with 1% SDS. Samples were run on a 0.8% electrophoresis gel, and the DNA band was quantified using a Typhoon scanner and ImageQuant. The slope of the exponential region of the resulting growth curves was used to calculate a growth rate constant and doubling time (see Table S1). Comparing the relative growth of wild-type and the *tkt* gene deletion strain, we see that the results of the PCR method match those of the colorimetric assay.

C.5 Computational Implementation

C.5.1 Whole-Cell Model Architecture

This chapter outlines the computational implementation of the whole-cell model and briefly summarizes the most important whole-cell model classes and functions. All of the source code for the whole-cell model, as well as comprehensive documentation of each class and function is freely available at SimTK: <http://www.simtk.org/home/wholecell>.

As illustrated in Figure C.17, the `Simulation` class coordinates the entire whole-cell model and is the primary class users interact with to execute the whole-cell model.

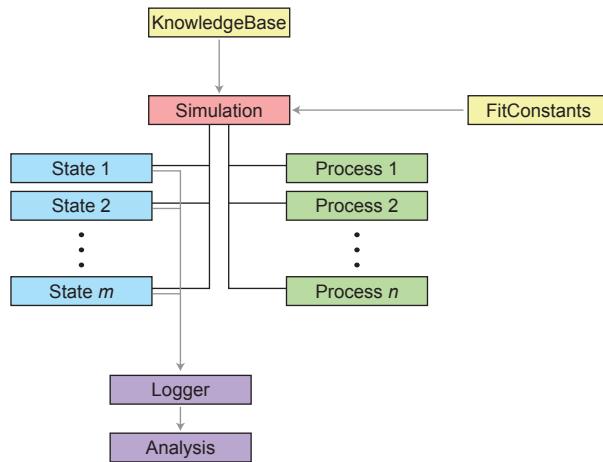


Figure C.17. Whole-cell model architecture.

The `Simulation` class performs several functions to coordinate the whole-cell model. First, the `Simulation` class instantiates all of the states and processes. Second, the `Simulation` class constructs an object graph of the states and processes. Specifically, the `Simulation` class triggers the states

and process to store references to the other states and processes with which they will interact during the simulation. Third, the **Simulation** class initializes the structural and quantitative parameters of each state and process. Specifically, the **Simulation** class passes the **KnowledgeBase** to the states and processes and triggers the states and process to retrieve data from the knowledge base. Fourth, the **Simulation** class manages the calculation of the initial cell state by triggering initialization methods of the states and processes. Fifth, the **Simulation** class oversees the simulation time line. Sixth, at each time step of the simulation the **Simulation** class orchestrates the calculation of the temporal evolution of the cell state by allocating shared resources among the processes and triggering the temporal evolution method of each of the processes. Finally, at each time step of the simulation the **Simulation** class triggers loggers to store the predicted values of the states.

The role of cell state allocation is most clearly illustrated by counterexample. If processes were executed serially and each process was passed the total count of each metabolite at the beginning of the time step, then there would be no constraint which prevents the processes from together using more than the total amount of a metabolite, resulting in a negative and unphysical metabolite count. If alternatively the processes were executed serially in the same order at each time step and each process updated the cell state following its execution, then there would be no constraint which prevents the earliest executed processes from using all of the copies of a metabolite needed by a later executed process. If processes were executed as in the previous example, except that processes were executed in a random order at each time step, then early processes could outcompete later processes as before, resulting in high frequency artifacts where processes randomly oscillate between metabolically on and off states on the same time scale as the simulation time step. In summary, input allocation has the benefit of reducing the process evaluation order dependence of the model without introducing high frequency artifacts.

The **KnowledgeBase** class represents the reconstructed *M. genitalium* knowledge base and provides this data to the states and processes. The **KnowledgeBase** is constructed outside the **Simulation** class, and is passed to the **Simulation** class to set the values of the parameters of the states and processes. The *M. genitalium* knowledge base was implemented in MySQL. The knowledge base web-interface was implemented in PHP.

Each cellular process sub-model is implemented as a subclass of **Process** and each cellular state is implemented as a subclass of **CellState**. This standardizes the implementation of the processes and states, and in particular, ensures that each process and state exposes a common interface.

Specifically, each state class (1) implements a common method to retrieve structural and quantitative data from the `KnowledgeBase`, (2) implements a common method to initialize its value at the beginning of each simulation, (3) implements a common method to calculate its contribution to the total cell mass, and (4) sets the value of a common property which tells the loggers which properties to store at each time step.

Each process class implements common methods that (1) retrieve structural and quantitative data from the `KnowledgeBase`, (2) model each process' contribution to the initialization and temporal evolution of the cell state, and (3) calculate the instantaneous and total life cycle metabolic and enzymatic requirements of each process. In addition to implementing the `Process` interface, each process satisfied a contract with the cell state classes to conserve mass and synchronize the redundantly represented parts of the cell state. For example, the process classes maintained synchrony between the DNA-bound copy number of each protein complex represented by the `Protein Complex` state and the location of each DNA-bound protein complex represented by the `Chromosome` state. Unit testing was used to verify that this contract between the states and processes was upheld.

The role of the `FitConstants` class is to fit the whole-cell model predictions to match the experimentally observed properties of *M. genitalium* including the mass doubling time, chemical composition, and RNA decay and expression. The `FitConstants` class implements an algorithm which computationally refines the values of the parameters of the states and processes. See Section C.1.3 for further discussion of the simulation fitting algorithm.

The role of the `Logger` classes is to store the simulated dynamics represented by the cell states, and to later retrieve this information for computational analysis. Logged dynamics are plotted and analyzed by the several classes in the `analysis` package, including the `PaperFigures` class which produces Figure 2-6 of the accompanying manuscript.

C.5.2 Test-Driven Development

The whole-cell model was extensively tested throughout the development process to ensure correctness. Using the MATLAB xUnit framework¹¹⁰, we developed 1,058 tests of the whole-cell model MATLAB code, including 100% of all state and process class methods and 92% of all state and process class lines. First, we developed 791 tests of each individual state and process class. For example, we developed several tests of the `Metabolism` process which check that the predicted and observed growth

rates are approximately equal, that the predicted growth rate responds correctly to the external environment and enzyme copy numbers, and that the `Metabolism` process conserves mass. Second, we developed 41 tests of collections of related processes and states. For example, we developed several tests of the process classes responsible for modeling translation and protein maturation. These tests check that these processes receive adequate amino acids from the `Metabolism` process, translate mRNA, and produce mature proteins. Third, we developed five tests of the entire simulation which check that the model predicts exponential growth with growth rates approximately equal to the experimentally observed rate, that all of the mass fractions – DNA, RNA, protein, metabolite, etc. – of the cell are approximately equal to their experimentally observed sizes, that the simulated and observed gene expression are approximately equal, and that the simulation obeys mass conservation. Stochastic functions were tested by checking the mean of their outputs. Additionally, we developed tests of the `KnowledgeBase`, analysis, and utility classes.

The whole-cell model code was maintained using Subversion⁷⁰. All tests were run at each code revision using the freely available Hudson continuous integration server²⁶², providing rapid feedback throughout the development process.

Separately, the model was experimentally validated as discussed in the accompanying manuscript and Section C.1.6.

C.5.3 Distributed Simulation

The whole-cell model was implemented in object-oriented MATLAB (R2010b) and simulated on a 128 core Rocks (v5.4)³³² Linux cluster. Simulation code was compiled using the MATLAB compiler²³⁷ and executed on the cluster nodes using the freely distributable MATLAB component runtime²³⁷. The open-source resource manager Torque⁴¹² and cluster scheduler Maui²³⁹ were used to distribute simulation jobs among the cluster nodes.

C.5.4 Computational Analysis

The whole-cell model simulations were analyzed using the MATLAB code in the `edu.stanford.covert.cell.sim.analysis` package.

C.5.5 Knowledge Base

The *M. genitalium* knowledge base was stored using a modified version of the BioWarehouse schema²¹³ in a MySQL relational database. Several tables and columns were added to the BioWarehouse schema primarily to represent additional functional genomic data. The knowledge base was viewed and edited using a web-interface implemented in PHP. The `KnowledgeBase` classes represented the knowledge base MySQL relational database in MATLAB.

C.5.6 Source Code Organization

The whole-cell model source code is available at SimTK, <http://www.simtk.org/home/wholecell>. The model source code is contained in the `simulation/src` directory and organized into several packages using the Java package naming convention:

- `edu.stanford.covert.cell.sim`: Simulation class and cellular process and state super-classes
- `edu.stanford.covert.cell.sim.analysis`: simulation analysis code
- `edu.stanford.covert.cell.sim.constant`: classes which represent constants such gene names
- `edu.stanford.covert.cell.sim.process`: implementations of all cellular processes
- `edu.stanford.covert.cell.sim.state`: implementations of all cell states
- `edu.stanford.covert.cell.sim.util`: utility classes for fitting, logging, plotting, printing, and more
- `edu.stanford.covert.cell.kb`: classes which represent the *M. genitalium* knowledge base
- `edu.stanford.covert.db`: classes for interacting with MySQL relational databases
- `edu.stanford.covert.io`: classes for reading and writing MAT- and JSON-formatted data
- `edu.stanford.covert.test`: classes for unit test logging, assessing code coverage, and mocking
- `edu.stanford.covert.util`: various utility functions

The `simulation` directory contains several unpackaged MATLAB, Perl, PHP, and Shell scripts for running simulations and unit tests on individual machines and compute clusters. The whole-cell model test code is similarly organized to the source code in the `simulation/src_test` directory. The knowledge base source code is located in the `knowledgebase` directory.

C.5.7 Key Classes & Functions

`CellState`

`edu.stanford.covert.cell.sim.CellState` is the base class from which all cell state classes are derived. `CellState` defines the interface each state exposes to `Simulation`. This interface enables `Simulation` to construct each state using data from the knowledge base, initialize the value of each state, account for the contribution of each state to the total cell mass, and log the dynamics of each state to disk. `MoleculeCountState` extends `CellState` by defining properties which represent the identity and copy numbers of molecules, and by providing a function which calculates the state's contribution to the total cell mass.

`Chromosome`

`edu.stanford.covert.cell.sim.state.Chromosome` implements the `Chromosome` state and provides an API which cellular process classes use to modify the chromosome object. To efficiently represent chromosomes, each property of the `Chromosome` class, except catenation, is implemented as an instance of the `CircularSparseMat` class of size $L \times 4$ where the first two columns represent the base-paired strands of the first chromosome and the second two columns represent the second chromosome. Table C.31 summarizes the properties of the `Chromosome` class. `polymerizedRegions(i, j)` sparsely represents the length in nt of a contiguous DNA strand starting at nucleotide i of strand/chromosome j . `linkingNumbers(i, j)` sparsely represents the linking number of a contiguous double-stranded DNA region starting at nucleotide i of strand/chromosome j . `gapSites(i, j)` is true if the sugar-phosphate of nucleotide i of strand/chromosome j is absent, and false otherwise. `abasicSites(i, j)` is true if the base of nucleotide i of strand/chromosome j is absent, and false otherwise. `damagedBases(i, j)` and `damagedSugarPhosphates(i, j)` are enumerations indicating the `Metabolite` identity of the base and sugar-phosphate of nucleotide i of strand/chromosome j . `intrastrandCrossLinks(i, j)` is true if nucleotide i of strand/chromosome j is cross linked to nucleotide $i + 1$ of the same strand and chromosome, and false otherwise. `strandBreaks(i, j)` is true if the phosphodiester bond between nucleotides i and $i + 1$ of strand/chromosome j is broken, and false otherwise. `hollidayJunctions(i, j)` is true if nucleotide i of strand/chromosome j forms a Holliday junction with the opposite strand, and false otherwise. `monomerBoundSites(i, j)` and `complexBoundSites(i, j)` are enumerations indicating the `Protein Monomer` or `Protein Complex` identity of the protein bound starting at nucleotide i of strand/chromosome j .

Due its complexity, the `Chromosome` class provides an API which process classes use to access and modify its properties. First, the API allows the `Chromosome` class to prevent processes from specifying invalid configurations of the chromosome state such as single-stranded binding proteins binding to double-stranded DNA or multiple proteins occupying the same nucleotide. Second, the API ensures that the `Chromosome` properties remains synchronized with the rest of the cell state. In particular, the API ensures that the `monomerBoundSites` and `complexBoundSites` properties remain synchronized with the copy numbers of DNA-bound proteins stored in the `Protein Monomer` and `Protein Complex` classes. Third, the API provides an abstraction layer over the chromosome representation, allowing process classes to focus on modeling biology. In particular, the API provides process classes methods for calculating the chromosomal regions accessible to each protein species and binding and releasing proteins to and from the chromosome. Furthermore, the API centralizes the modeled interactions among DNA-binding proteins, and specifically the reconstructed rules which describe the bound proteins each protein species is able to displace from the chromosome (see Table S3O). Fourth, the API eliminates the need to redundantly implement similar codes in each of the chromosome-interacting process classes. Finally, the API allows the `Chromosome` class to implement caching and other strategies to optimize simulation run-time performance. Table C.30 summarizes the public methods of the `Chromosome` class.

Each chromosome-interacting process class inherits from `ChromosomeProcessAspect`. This class provides several additional methods to help process classes interact with the `Chromosome` class. See Section C.5.7 for further discussion of the implementation of `ChromosomeProcessAspect`.

`ChromosomeProcessAspect`

`edu.stanford.covert.cell.sim.ChromosomeProcessAspect` provides several convenience methods which cellular process classes use to access and modify the properties of the `Chromosome` class. C.32 summarizes the public methods of the `ChromosomeProcessAspect` class.

`CircularSparseMat`

`edu.stanford.covert.util.CircularSparseMat` efficiently represents sparse, circular, multidimensional matrices. `CircularSparseMat` inherits from `SparseMat`, adding support for circular matrix reference and assignment. Several properties of the `Chromosome` class are implemented as instances of `CircularSparseMat`.

Table C.30. Chromosome class public methods.

Function	Description
Accessors	
isRegionSingleStranded	Checks if region is single-stranded.
isRegionDoubleStranded	Checks if region is double-stranded.
isRegionPolymerized	Checks if region is polymerized.
isRegionNotPolymerized	Checks if region is not polymerized.
isRegionProteinFree	Checks if region is not occupied by proteins.
isRegionUndamaged	Checks if region is unmodified (except methylated MunI R/M sites).
isRegionAccessible	Checks if a protein species can bind a region. Checks if region is polymerized, protein free or the query protein is capable of displacing all bound proteins, and undamaged.
getAccessibleRegions	Returns all regions accessible (polymerized, protein free or the query protein is capable of displacing all bound proteins, and undamaged) to a protein species.
sampleAccessibleSites	Efficiently returns a short list of regions accessible (polymerized, protein free or the query protein is capable of displacing all bound proteins, and undamaged) to a protein species.
sampleAccessibleRegions	Efficiently returns a short list of sites vulnerable to a specific type of DNA damage.
Modifiers	
setRegionPolymerized	Marks a region polymerized.
setRegionUnwound	Moves a region of strand 2 of chromosome 1 to strand 2 of chromosome 2. Conserves the linking number of chromosome 1 by increasing the superhelical density of the downstream double-stranded region of chromosome 1.
setSiteProteinBound	If the region is accessible to the query protein, binds query protein to region. Displaces any previously bound proteins. Updates Protein Monomer and Protein Complex states.
setRegionProteinUnbound	Releases all proteins bound to a region. Updates Protein Monomer and Protein Complex states.
setSiteDamaged	Introduces a gap site, abasic site, modified base or sugar phosphate, cross link, strand break, or Holliday junction at the query nucleotide.
stochasticallySetProteinUnbound	Stochastically releases all copies of a protein species at a specified probability. Updates Protein Monomer and Protein Complex states.
modifyBoundProtein	If the query protein and previous bound protein have the same footprint, changes the identity of the protein bound at a site. Updates Protein Monomer and Protein Complex states.

Table C.31. Computational representation of the Chromosome class.

Physical Property	Name	Size	Type
Polymerization	<code>polymerizedRegions</code>	$L \times 4$	<code>CircularSparseMat<int></code>
Winding	<code>linkingNumbers</code>	$L \times 4$	<code>CircularSparseMat<double></code>
Modification			
Gap site	<code>gapSites</code>	$L \times 4$	<code>CircularSparseMat<logical></code>
Abasic site	<code>abasicSites</code>	$L \times 4$	<code>CircularSparseMat<logical></code>
Sugar-phosphate	<code>damagedSugarPhosphates</code>	$L \times 4$	<code>CircularSparseMat<int></code>
Base	<code>damagedBases</code>	$L \times 4$	<code>CircularSparseMat<int></code>
Intrastrand cross link	<code>intrastrandCrossLinks</code>	$L \times 4$	<code>CircularSparseMat<logical></code>
Strand break	<code>strandBreaks</code>	$L \times 4$	<code>CircularSparseMat<logical></code>
Holliday junction	<code>hollidayJunctions</code>	$L \times 4$	<code>CircularSparseMat<logical></code>
Protein occupancy			
Monomer	<code>monomerBoundSites</code>	$L \times 4$	<code>CircularSparseMat<int></code>
Complex	<code>complexBoundSites</code>	$L \times 4$	<code>CircularSparseMat<int></code>
Catenation	<code>segregated</code>	1×1	<code>logical</code>

Table C.32. ChromosomeProcessAspect class public methods.

Function	Description
Accessors	
<code>isDnaBound</code>	Checks if a site is bound by a specified protein species.
<code>findProteinInRegion</code>	Returns positions of a protein species within a specified region.
Modifiers	
<code>bindProteinToChromosome</code>	Convenience function for <code>setSiteProteinBound</code> . Updates local enzyme state and global <code>Protein Monomer</code> and <code>Protein Complex</code> states.
<code>bindProteinToChromosomeStochastically</code>	Binds multiple copies of a protein species to specified positions each with a specified probability. Updates local enzyme state and global <code>Protein Monomer</code> and <code>Protein Complex</code> states.
<code>modifyProteinOnChromosome</code>	Convenience function for <code>modifyBoundProtein</code> . Updates local enzyme state and global <code>Protein Monomer</code> and <code>Protein Complex</code> states.
<code>releaseProteinFromChromosome</code>	Convenience function for <code>stochasticallySetProteinUnbound</code> . Updates local enzyme state and global <code>Protein Monomer</code> and <code>Protein Complex</code> states.
<code>releaseProteinFromSites</code>	Convenience function for <code>setRegionProteinUnbound</code> . Updates local enzyme state and global <code>Protein Monomer</code> and <code>Protein Complex</code> states.

Compartment

`edu.stanford.covert.cell.sim.constant.Compartment` defines the identifier and name of each of the six modeled compartments: cytosol, extracellular space, membrane, nucleoid, terminal organelle cytosol, and terminal organelle membrane. State classes which derive from `MoleculeCountState` represent the copy number of each molecule in each of these six compartments.

DatabaseLogger

`edu.stanford.covert.cell.sim.util.DatabaseLogger` provides methods to store and retrieve simulated single cell dynamics and simulation meta data to/from a relational database. `DatabaseLogger` implements the logger interface defined by `Logger`.

DiskLogger

`edu.stanford.covert.cell.sim.util.DiskLogger` was the primary logger used to store the simulated single cell dynamics. `DiskLogger` provides methods to store and retrieve the simulated dynamics of individual cells and simulation meta data to/from disk. `DiskLogger` implements the logger interface defined by `Logger`. `SimulationEnsemble` provides methods to retrieve simulated data stored by `DiskLogger` for multiple cells. `SimulationDiskUtil` provides methods to retrieve and search simulation meta data.

FitConstants

`edu.stanford.covert.cell.sim.util.FitConstants` provides several methods for fitting the simulated dynamics of cellular populations to experimental observations as well as computationally aligning the quantitative parameters of the cellular process sub-models such that the sub-models are mutually consistent. See Section C.1.3 for further discussion of simulation fitting.

Gene

`edu.stanford.covert.cell.sim.constant.Gene` defines the identifier, name, type (m-, r-, s-, or t-RNA), location, length, and directionality of each gene.

Logger

`DatabaseLogger`, `DiskLogger`, and `SummaryLogger` define three ways of storing simulated single cell dynamics. `DatabaseLogger` provides methods to store and retrieve simulated data to/from a relational database. `DiskLogger` provides methods to store and retrieve simulated data directly to/from disk. `SummaryLogger` provides methods to store a small amount of key simulated data to/from disk. `DatabaseLogger`, `DiskLogger`, and `SummaryLogger` derive from `edu.stanford.covert.cell.sim.util.Logger`. `Logger` defines the common interface which `Simulation` uses to initialize, append, and finalize each simulation log.

KnowledgeBase

The primary function of `edu.stanford.covert.cell.kb.KnowledgeBase` is to represent the curated knowledge base of *M. genitalium* physiology, and to provide this information to the states and processes. In this way, `KnowledgeBase` serves as an abstraction layer between the knowledge base relational database and the whole-cell MATLAB model. Specifically, the knowledge base represents several reconstructed properties of *M. genitalium* including the genomic sequence; the location, length, and direction and observed expression of half-life of each gene; the transcription unit organization of the chromosome; the subunit composition of each macromolecular complex; the chemical composition of *M. genitalium* and its typical external environment; and the stoichiometry, kinetics, energetics, and catalysis of each chemical reaction.

KnowledgeBaseObject

`edu.stanford.covert.cell.kb.KnowledgeBaseObject` is the base class from which all MATLAB classes which represent objects contained the *M. genitalium* knowledge base are derived, including `KnowledgeBase`. See Section C.5.7 for further discussion of the role of the *M. genitalium* knowledge base.

MoleculeCountState

`edu.stanford.covert.cell.sim.MoleculeCountState` is the superclass for all cell state classes which represent the copy numbers of individual molecules including the `Metabolite`, `Rna`, `Protein Monomer`, and `Protein Complex` classes. `MoleculeCountState` extends `CellState` by defining properties which represent the identifier, name, and molecular weight of each molecular species and

the copy number of each species in each of the six compartments defined by `Compartment`.

`PaperFigures`

The `run` method of `edu.stanford.covert.cell.sim.analysis.PaperFigures` produces Figure 2-6 presented in the accompanying manuscript.

`polymerize`

`edu.stanford.covert.cell.sim.util.polymerize` implements a model of nutrient and energy allocation among nascent polymers. The `Replication`, `Transcription`, and `Translation` processes use `Polymerize` to model the allocation of dNTPs, NTPs, and amino acids among active DNA polymerases, RNA polymerases, and ribosomes. See Section C.3.18, C.3.25, or C.3.27 for further discussion of the mathematical model implemented by the `Polymerize` function.

`Process`

`edu.stanford.covert.cell.sim.Process` is the base class from which all process sub-model classes are derived. `Process` defines the public interface that each process exposes to `Simulation`. This interface enables `Simulation` to construct each sub-model using data from the knowledge base, evaluate each process' contributions to the initialization and temporal evolution of the cell's state, determine the parts of the cell state accessed and modified by each sub-model, and to calculate the life cycle and instantaneous metabolic demands of each process. `ReactionProcess` extends `Process` by adding several properties which represent the structure of the modeled biological network as well as one method which initializes the values of those properties using the knowledge base.

`RandStream`

The MATLAB built in `RandStream` class provides four low-level methods – `rand`, `randi`, `randn`, and `randperm` – for generating random numbers from a specific random stream. `edu.stanford.covert.util.RandStream` extends the functionality of the built in `RandStream` class by (1) providing methods to execute four statistics toolbox functions – multinomially-distributed random number generation (`mnrnd`), Poisson-distributed random number generation (`poissrnd`), a wrapper over several random number generation methods (`random`), and multinomially-distributed random number generation with and without replacement (`randsample`) – on a specific random stream, and (2) adding

four methods `randCounts`, `stochasticRound`, `randomlySelectRows`, and `randomlySelectNRows`. `randCounts` returns the number of times each of N objects with counts m_i for $i = 1..N$ is selected without replacement. `stochasticRound` stochastically rounds elements of a matrix M . With probability $M_{ij} \bmod 1$ `stochasticRound` replaces M_{ij} with its ceiling, and otherwise replaces M_{ij} with its floor. `randomlySelectRows` randomly returns each row of a matrix with a specified probability. `randomlySelectNRows` randomly returns N rows of a matrix, each with equal probability.

`ReactionProcess`

`edu.stanford.covert.cell.sim.ReactionProcess` is the base class from which several process classes representing large network models including `Metabolism` are derived. `ReactionProcess` extends `Process` by defining several properties which represent the structure of the modeled biological network and by providing one method which initializes the values of those properties using the knowledge base.

`SparseMat`

`edu.stanford.covert.util.SparseMat` efficiently represents sparse multidimensional matrices. Externally, `SparseMat` behaves similarly to matrices created using the built in MATLAB `sparse` function with two exceptions: (1) `SparseMat` supports multidimensional matrices, whereas the built in `sparse` function supports only one- and two-dimensional matrices, and (2) `SparseMat` redefines linear indexing in reference and assignment operations as syntactic sugar for the composition of linear indexing with `sub2ind`. `SparseMat` supports many common matrix operations including addition, subtraction, multiplication, and division. Internally, `SparseMat` represents an N -dimensional matrix containing n non-zero elements as an $n \times (N + 1)$ two-dimensional matrix containing the indices and values of each non-zero element. `CircularSparseMat` extends `SparseMat` by enabling circular matrix reference and assignment. Several properties of the `Chromosome` class are implemented as instances of `CircularSparseMat`.

`Simulation`

`edu.stanford.covert.cell.sim.Simulation` is the primary class users interact with to run and analyze whole-cell model simulations. The primary functions of `Simulation` are several-fold: (1) `initializeConstants` constructs the states and processes using data contained in the knowledge base, (2) `initializeState` randomly initializes the cell state, (3) `evolveState` sequentially executes the

process sub-models in a random order, (4) `applyOptions`, `applyParameters`, `applyPerturbations` override the default values of each option and parameter, for example to simulate gene disruption, (5) `run` executes a complete simulation by initializing the cell state, iteratively executing `evolveState`, and optionally triggering loggers to store the predicted single cell dynamics.

`SimulationEnsemble`

`edu.stanford.covert.cell.sim.util.SimulationEnsemble` provides methods for retrieving simulated single cell dynamics stored by `DiskLogger` and `SummaryLogger` for multiple cells. `DiskLogger` and `SummaryLogger` provide methods for retrieving the simulated dynamics of individual cells.

`SimulationDiskUtil`

`edu.stanford.covert.cell.sim.util.SimulationDiskUtil` provides methods to retrieve and search simulation meta data generated by `DiskLogger`.

`SimulationStateSideEffect`

`edu.stanford.covert.cell.sim.SimulationStateSideEffect` and `edu.stanford.covert.cell.sim.SimulationStateSideEffectItem` represent the side effects of one process on parts of the cell state outside the direct focus of that process. For example, when a sub-model binds one protein to the chromosome, displacing another protein from the chromosome modeled by a different cellular process, these classes represent the side effect of displacing that second protein on the DNA-bound copy number of that second protein redundantly represented by the `Protein Monomer` state. During the execution of each cellular process sub-model, cellular states use `SimulationStateSideEffect` to keep track of their side effects on other parts of the cell state primarily modeled by other sub-models. After the completion of the execution of each sub-model, the `Simulation` class processes the accumulated side effects and updates the affected states. The principal advantages of this approach are two-fold. First, this approach minimizes direct communication among the cellular states which reduces complexity and run time. Second, this approach outlines the side effects of processes on parts of the cell state outside their direct purview.

Each atomic, mass-balanced modification of a distant part of the cell's state is represented by an instance of the `SimulationStateSideEffect` class of size 1×1 whose children of type `SimulationStateSideEffectItem` represent the specific effect of the modification on the other parts of the cell

state. For example, each event where a process displaces a foreign protein from the chromosome is represented by an instance of the `SimulationStateSideEffect` class which has two children each of type `SimulationStateSideEffectItem`, one of which represents the decrement of the DNA-bound copy number of the displaced protein and a second which represents the increment of the free copy number of the displaced protein.

`SimulationStateSideEffectItem`

See Section C.5.7 for discussion of the `SimulationStateSideEffectItem` class.

`SummaryLogger`

`edu.stanford.covert.cell.sim.util.SummaryLogger` provides methods to store a key subset of simulated single cell dynamics to disk. `SummaryLogger` implements the common logger interface defined by `Logger`. `SimulationEnsemble` provides methods to retrieve results logged by `SummaryLogger`.

C.5.8 Third-Party Software

The *M. genitalium* whole-cell model and knowledge base were developed using the software libraries and applications listed below.

Knowledge Base *pear.php.net*

- BioWarehouse
biowarehouse.ai.sri.com
 - Excel
office.microsoft.com/en-us/excel
 - GeSHi
qbnz.com/highlighter
 - Marvin
chemaxon.com/products/marvin
 - MySQL
mysql.com
 - MySQL connector J
mysql.com/products/connector
 - Pear
- PHP
php.net
 - PHPExcel
phpexcel.codeplex.com

Modeling

- GLPKmex
glpkmex.sourceforge.net
- MATLAB
mathworks.com/products/matlab
- MATLAB Utilities
home.online.no/ pjacklam/matlab/software/util
- multiprod
mathworks.com/matlabcentral/fileexchange/8773
- JSON Marshaller
code.google.com/p/jsonmarshaller
- MATLAB Compiler
mathworks.com/products/compiler
- Maui
clusterresources.com/products/maui
- Torque
adaptivecomputing.com/products/torque.php
- Perl
perl.org
- Rocks
www.rocksclusters.org

Simulation

- CentOS
centos.org

Analysis & Visualization

- Apache
apache.org
- boundedline
mathworks.com/matlabcentral/fileexchange/27485
- Bullseye
mathworks.com/matlabcentral/fileexchange/16458
- DHTML Window
dynamicdrive.com/dynamicindex8
- ffmpeg
ffmpeg.org
- flowplayer
flowplayer.org
- Funet_Bezier
mathworks.com/matlabcentral/fileexchange/6661
- Illustrator
www.adobe.com/products/illustrator.html
- Inkscape
inkscape.org
- L^AT_EX
www.latex-project.org
- jqGrid
trirand.com/blog
- pdftk
pdflabs.com/tools/pdftk-the-pdf-toolkit
- PHP
php.net
- propertygrid
mathworks.com/matlabcentral/fileexchange/28732
- rude
mathworks.com/matlabcentral/fileexchange/6436
- Silk
famfamfam.com/lab/icons/silk
- spanFigure
mathworks.com/matlabcentral/fileexchange/31604
- swtest
mathworks.com/matlabcentral/fileexchange/13964
- tick2text
mathworks.com/matlabcentral/fileexchange/16003
- Uniform
uniformjs.com
- uitabpanel
mathworks.com/matlabcentral/fileexchange/11546

Interactive Visualization

- alivepdf
alivepdf bytearray.org
- amfphp
silexlabs.org/amfphp
- as3gif
code.google.com/p/as3gif
- AS3FlexDB
code.google.com/p/as3flexdb
- degrafa
degrafa.org
- Flash
adobe.com/products/flashplayer.html
- Flex
adobe.com/products/flex.html
- flexlib
code.google.com/p/flexlib
- Flex Menu Accelerators
rphelan.com/2008/03/17/flex-menu-accelerators
- Flex Multiline Button
forestandthetrees.com/flex-multiline-button
- MS Visual C#
microsoft.com/visualstudio
- Premier

adobe.com/products/premiere.html

- print-as3
code.google.com/p/printf-as3
- Screen2Video
viscomsoft.com/products/screen2video
- Tooltip Speech Bubble
blog.flexmp.com/2008/09/10
- tweener
code.google.com/p/tweener

Testing & Development

- absolutepath
mathworks.com/matlabcentral/fileexchange/3857
- Doxygen
www.stack.nl/~dimitri/doxygen
- Hudson
hudson-ci.org
- MATLAB xUnit framework
mathworks.com/matlabcentral/fileexchange/22846
- m2cpp
mathworks.com/matlabcentral/fileexchange/25925
- m2html
artefact.tk/software/matlab/m2html
- Subversion
subversion.apache.org

Appendix D

List of supplementary materials

This thesis contains three supplementary tables and one supplementary movie.

D.1 Supplemental tables

This thesis contains three supplementary Excel workbooks which contain three supplementary tables (Table 1-3) to the *M. genitalium* whole-cell model described in Chapter 4. These supplementary tables are reproduced with permission from Karr JR, Sanghvi JC, Macklin DN, Gutschow MV, Jacobs JM, Bolival B, Assad-Garcia N, Glass JI & Covert MW. A whole-cell computational model predicts phenotype from genotype. *Cell* 150, 389–401 (2012); copyright 2012 Elsevier Inc.

Table S3A-S3R define the reconstructed organism. Table S3S provides a complete list of all the sources of the reconstructed *M. genitalium* organism. Table S3T-S3BK describe the reconstruction process. Table S3BL-S3BO describe the parameters, metabolites, and enzymes of each cell state variable and process sub-model. Table S2B-S2C list the computationally refined values of the reconstructed cellular composition and gene expression. Table S2A-S2D list the computationally predicted growth rate, metabolite concentrations, and reaction fluxes. Table S1 lists the experimentally measured growth rate of 12 single-gene disruption strains.

Experimental results

- S1. *Experimentally measured growth rates of wild type *M. genitalium* and 12 single-gene disruption strains.*

Growth rates of wild type *M. genitalium* and of 12 single-gene disruption strains were determined using a colorimetric growth assay and by qPCR. See Section C.4 for further discussion of both methods.

Computational results*S2A. Wild type in silico population.*

Life cycle lengths, cell cycle phase lengths, growth rates, masses, and cellular compositions of 192 in silico wild type cells.

S2B. Average in silico dry mass composition.

Average dry cell mass composition and metabolic production and process byproduction of a population of in silico wild type cells.

S2C. Comparison of average in silico and experimentally observed gene expression.

Comparison of predicted gene expression with that observed by Weiner et al., 2003. Values listed are the average copy number of each RNA and protein gene product at the beginning of the life cycle.

S2D. Average in silico metabolic reaction fluxes.

Average predicted metabolic reaction fluxes of a population of in silico wild type cells.

S2E. Comparison of average in silico and experimentally observed metabolite concentrations.

Comparison of predicted metabolite concentrations with that observed by Bennett et al.²⁴ and curated by the CyberCell database³⁸⁷.

S2F. Average in silico DNA-bound protein collisions.

Average predicted frequency of DNA-bound protein collisions between pairs of 13 DNA-binding proteins.

S2G. Single-gene disruption strain phenotypes.

Predicted phenotype of each single-gene disruption strain and comparison of the predicted and observed¹⁶⁹ essentiality of each gene.

S2H. Single-gene disruption strain phenotypic classes.

Summary of each single-gene disruption phenotypic class, including the number of strains belonging to each class and the identities of the representative in silico cells plotted in Figure 6B.

M. genitalium* reconstructionS3A. Taxonomy.*

M. genitalium taxonomy and cross references.

S3B. States.

16 MATLAB classes represent the instantaneous state of *M. genitalium*.

S3C. Processes.

The *M. genitalium* whole-cell model is hybrid computational model composed of 28 sub-models of distinct cellular processes and biochemical reactions. Each sub-model is implemented as a MATLAB class. The sub-models are initialized in the order shown here.

S3D. Parameters.

The majority of the experimental data used to train the model is organized into structured tables. All experimental data not organized into structured tables is represented as a “parameter” and listed in this table. This table indicates the value, dimension, and primary source of each parameter as well as the state, process, reaction, and/or molecule which the parameter describes.

S3E. Compartments.

The *M. genitalium* whole-cell model is implemented as a compartment model. Each compartment is assumed to be well-stirred on the 1 s simulation time scale.

S3F. External stimuli.

The *M. genitalium* whole-cell model represents 10 external stimuli including temperature, several types of radiation, and three Boolean-valued stress conditions. This table indicates the simulated values of the stimuli.

S3G. Metabolites.

The *M. genitalium* whole-cell model represents 722 small molecules. This table describes the physical properties of each metabolite: empirical formula, SMILES, charge, hydrophobicity, molecular weight, volume, pK_as, pI, logP, logD, and maximal uptake and efflux rates. The molecular weight, volume, pK_as, pI, logP, and logD of each metabolite were calculated using ChemAxon Marvin. The table also lists a two-level classification over metabolites, notes cross references for each metabolite into external databases, and indicates the reactions in which each metabolite participates.

S3H. Media composition.

In silico media composition. The media composition is based on the experimentally characterized composition of the SP-4 media on which *M. genitalium* is generally cultured, with additional essential molecules added to support *in silico* cellular growth. Non-gaseous molecules are added to the extracellular environment at the beginning of the simulation and either drain or accumulate over the length of the simulation. Gaseous molecules are continuously perfused into the extracellular environment and are maintained at a constant concentration throughout the simulation.

S3I. Dry biomass composition.

The small molecule composition of the dry mass of *M. genitalium* was compiled from several experimental studies. See Section C.2.5 for further discussion.

S3J. Genes.

M. genitalium contains 525 genes. This table lists the genomic location of each gene: start coordinate, length, direction; the type of each gene and codon and amino acid for tRNA genes; the experimentally characterized essentiality of each gene; the observed expression of each gene under physiologic, cold and heat shock conditions; and the observed half-life of each RNA transcript. The table also lists several calculated properties of each gene: the sequence and G/C content of each gene, the molecular weight and pI of each transcript, and the half-life, decay rate, synthesis rate, and RNA polymerase binding probability of each RNA. Additionally, the table lists the homologs of each gene in several related organisms, cross references for each gene in external databases, the transcription unit into which each gene is organized, and the reactions in which each gene product participates.

S3K. Transcription units.

The *M. genitalium* genomic model is organized into 335 transcription units. The transcription unit structure reconstruction was based primarily on the transcription unit structure of *M. pneumoniae* experimentally defined by Güell and colleagues¹⁴⁰ and the promoter sites predicted by Weiner and colleagues⁴³⁰. See Section C.2.12 for further discussion. The genomic location of each transcription unit was calculated from that of its constituent genes.

S3L. Genome features.

In addition to genes, transcription units, and promoters, the *M. genitalium* genomic model contains three types of genomic features: DnaA boxes, restriction/modification (R/M) sites, and short tandem repeats. The locations of DnaA boxes and R/M sites were predicted based on consensus sequences. Short tandem repeats were reconstructed from predictions by Ma et al.²³⁰ and Washio et al.⁴²⁸. This table also encodes the genomic location of the proteolysis tag of the tmRNA gene. Additionally, this table indicates the genes and transcription units which overlap each genomic feature.

S3M. Protein monomers.

M. genitalium protein monomer physical properties – localization, signal sequence, N-terminal methionine cleavage, prosthetic groups, chaperone requirements, topology, reaction catalysis, and DNA footprint – were reconstructed from several sources. The sequence, molecular formula, weight, pI, instability, stability, aliphatic index, GRAVY, extinction coefficient, and absorption were calculated for each protein monomer from the DNA sequence of the corresponding gene.

S3N. Macromolecular complexes.

M. genitalium macromolecular complex physical properties – subunit and small molecule composition, localization, prosthetic groups, disulfide bonds, chaperone requirements, DNA footprint, and reaction catalysis – were reconstructed from several sources. The molecular weight of each complex was calculated based on its subunit composition.

S3O. Reactions.

M. genitalium chemical reaction physical properties – small molecule stoichiometry, thermodynamics, kinetics, cofactors, and enzyme catalysis – were reconstructed from several sources.

S3P. Transcriptional regulation.

The *M. genitalium* transcriptional regulatory network contains five transcriptional regulators and regulates 27 transcription units containing 54 genes. This table lists the physical properties of each transcriptional regulatory interaction: transcription factor binding site, affinity, and expression fold change effect. The transcriptional regulatory network was reconstructed from several sources.

S3Q. Post-translational regulation.

The *M. genitalium* whole-cell model includes post-translational Boolean regulatory rules governing the activity of 10 proteins. These regulatory rules are functions of external stimuli and the concentrations in mM of antibiotics and other intracellular metabolites. The rules were reconstructed from several sources including DrugBank¹⁹².

S3R. Notes.

Collection of notes describing the *M. genitalium* reconstruction and model design.

S3S. References.

The *M. genitalium* whole-cell model was trained using data curated from over 900 primary sources, databases, reviews, and books.

S3T. Genome functional annotation.

M. genitalium genes were functionally annotated through extensive curation of the primary literature, genome databases, functional genomics predictors, and homology searches.

S3U. Transcription unit structure.

The *M. genitalium* transcription unit structure was reconstructed primarily based on the *M. pneumoniae* transcription unit structure experimentally defined by Güell and colleagues¹⁴⁰, and the promoter structure predicted by Weiner and colleagues⁴³⁰. See Section C.2.12 for further discussion.

S3V. Short tandem repeats.

M. genitalium contains 19 short tandem repeats. The reconstructed short tandem repeats are a consensus of those predicted by Ma et al.²³⁰ and Washio et al.⁴²⁸.

S3W. M. genitalium mRNA expression.

M. genitalium relative mRNA expression was reconstructed from the *M. pneumoniae* mRNA expression reported by Weiner and colleagues⁴³¹ by (1) mapping the expression of *M. pneumoniae* mRNA onto their *M. genitalium* homologs, (2) converting the discretely reported mRNA expression from Figure 4 of Weiner et al. to numerical values using the provided color scale, and (3) transforming the reported log-scale data to a linear scale⁴³¹. The expression of unobserved and non-homologous RNA were imputed as the average expression of other members of the same operon, or for mRNA not organized into operons as the average expression of all mRNA. Absolute mRNA expression was reconstructed by additionally considering the total *M. genitalium* RNA mass and the relative contributions of m, r, s, and tRNA to the total RNA mass. See Table S3AU for additional information.

S3X. tRNA expression.

Relative tRNA expression was reconstructed by equating the relative expression of each tRNA species to the relative frequency its coded base amino acid divided by the number of tRNA which code for the amino acid. Absolute tRNA expression was reconstructed by additionally considering the total *M. genitalium* RNA mass and the relative contributions of m, r, s, and tRNA to the total RNA mass. See Table S3AU for additional information.

S3Y. M. genitalium mRNA half-lives.

M. genitalium mRNA half-lives were reconstructed by mapping the experimentally measured half-lives of *E. coli* mRNA onto their *M. genitalium* homologs (column F). Half-lives were reconstructed on the basis of the half-lives measured for growth on M9 media because M9 media mostly similarly corresponds to the SP-4 media of *M. genitalium* and because a larger number of mRNA half-lives were reported for M9 than for LB media. The half-lives of non-homologous mRNA were imputed as the average half-life of all homologous mRNA. Finally, we adjusted the half-lives of several mRNA to satisfy the constraint that cotranscriptionally expressed mRNA have the same synthesis rate, approximately equal to the product of their expressions and half-lives.

S3Z. rRNA modifications and enzymes.

M. genitalium rRNA modifications were reconstructed by (1) mapping the *E. coli* rRNA modifications curated by Peil onto their *M. genitalium* homologs (columns B-L)²⁹⁸, and (2) only retaining modifications for which an rRNA modification enzyme is annotated in the *M. genitalium* genome (column A). The catalysis and kinetics of rRNA modification reactions were reconstructed from several experimental studies^{19,147,204,403}.

S3AA. tRNA modifications.

M. genitalium tRNA modifications were reconstructed by (1) identifying the complement of tRNA modification enzymes in *M. genitalium*, (2) identifying the modifications catalyzed by each enzyme in *E. coli*, and (3) mapping the observed *E. coli* tRNA modifications onto their *M. genitalium* homologs considering sequence alignment. See Section C.3.22 for further discussion.

S3AB. tRNA modification enzymes.

The kinetics and catalysis of tRNA modifications were reconstructed from several experimental studies^{250,272,307,345,346,349}.

S3AC. Protein monomer localization.

The localization of each *M. genitalium* protein monomer was reconstructed as a consensus of several sources of experimental observations and computational predictions of protein localization and signal peptides: ASAP¹³³, BRENDA⁵⁹, DBSubLoc¹⁴³, EchoBASE²⁵⁸, GenoBASE³⁴⁷, Proteome Analyst³⁹⁰, PSORTdb³³¹, and UniProt⁷¹. Protein localizations experimentally observed in other organisms were mapped to their *M. genitalium* homologs. Table S3E defines the subcellular localizations (column D). Protein monomers were considered membrane-associated if either (1) they are localized to the membrane or (2) they are subunits of macromolecular complexes which are localized to the membrane.

S3AD. Protein complex localization.

The subcellular localization of each *M. genitalium* protein complexes was calculated by (1) assigning all complexes whose subunits are all cytoplasmically localized to the cytoplasm, and (2) assigning all complexes containing at least one membrane-localized subunit to the membrane. Table S3AC lists the reconstructed subcellular localization of each protein monomer.

S3AE. Protein localization signal sequences.

The subcellular localization type II signal sequence of each protein monomer was reconstructed as the consensus of several experimental observations and statistical predictions of signal sequences and protein localization: Phobius¹⁸², PrediSi²⁸², SignalP²³, SOSUI^{135,154}, and SPdb⁶⁵.

S3AF. Disulfide bonds.

M. genitalium disulfide bonds were reconstruction as the consensus of experimental observations of disulfide bonds in homologs^{63,71} and computationally predicted disulfide bonds³⁰². We rejected most computationally predicted disulfide bonds.

S3AG. Protein folding.

M. genitalium chaperone-substrate interactions were reconstructed from three proteome-scale studies of chaperone-substrate interactions: *E. coli* DnaK⁹⁵, *E. coli* GroEL¹⁸⁷, and *B. subtilis* GroEL¹¹³. *M. genitalium* chaperone-substrate interactions were inferred from these studies by mapping chaperone-substrate interactions onto their *M. genitalium* homologs. For completeness, we also reproduce the SecB-substrate interactions observed by Baars and colleagues in *E. coli*¹³.

S3AH. Protein modifications.

M. genitalium protein modifications were reconstructed based on several experimental studies of protein modification including a proteome-wide study of phosphorylations in *M. genitalium*³⁸⁶ and a proteome-wide study of covalent modifications in *S. oneidensis* MR-1¹⁴⁵. The *M. genitalium* protein modification network was reconstructed by mapping observed modifications onto the homologs of their protein substrates for which (1) a specific modification locus was identified, and (2) *M. genitalium* contains a protein modification enzyme. Most of the kinetics of the *M. genitalium* protein modification reactions were reconstructed by mapping experimentally observed protein modification reaction kinetics in *E. coli*, *S. enterica*, and *S. mutans* to their *M. genitalium* homologs^{117,137,241}.

S3AI. Macromolecular complexation.

The *M. genitalium* macromolecule complex network was reconstructed based on a consensus of several curated databases of experimental observations of macromolecular complexes – BioCyc¹⁸⁸, BRENDA⁵⁹, GenoBASE³⁴⁷, and UniProt⁷¹ – as well as several primary literature sources. Experimental observations made in other organisms were mapped onto their *M. genitalium* homologs.

S3AJ. Protein structures.

The structure and topology of each *M. genitalium* protein monomer was reconstructed as the consensus of computational predictions¹⁸² and experimental observations in other organisms extrapolated to *M. genitalium* on the basis of sequence homology⁷¹. PDB entries for homologs of each *M. genitalium* protein were reconstructed from three databases^{71,133,347}.

S3AK. DNA footprints.

The DNA footprint of each *M. genitalium* protein monomer (except DisA, MG105) was reconstructed as the consensus of experimental observations of the DNA footprint size of *M. genitalium* homologs reported in three databases – 3D-Footprint⁷², NDB²⁸, and ProNIT²⁰⁵ – and several primary sources as well as the lengths of the binding motifs of each DNA-binding protein. The DNA footprint of DisA has not been experimentally characterized and was predicted by regressing DNA footprint size on protein mass.

S3AL. Reaction kinetics.

M. genitalium reaction kinetics were reconstructed based on the reported kinetics of reactions catalyzed by *M. genitalium* homologs curated in three databases: SABIO-RK⁴³⁶, BRENDA⁵⁹, and BioCyc¹⁸⁸. Among reported kinetic rates we chose the highest rate reported for the nearest *M. genitalium* homolog.

S3AM. Cofactors.

The prosthetic groups and cofactors required for each chemical reaction were reconstructed from the primary literature and reports organized into seven databases: SABIO-RK⁴³⁶, BRENDA⁵⁹, BioCyc¹⁸⁸, GenoBASE³⁴⁷, UniProt⁷¹, Kinetikon¹⁹⁰, and Metal-MaCiE⁸. Prosthetic groups reported as lists of multiple equivalent metal ions were simplified to a single chemical species to minimize the number of unique modeled protein structures by choosing the most common chemical species according to cell composition (Na^+ , K^+ > Mg^{2+} , Cl^- > Fe^{2+} , Fe^{3+} > Ca^{2+} > Mn^{2+} > Cu^{2+} > Mo^{6+} , Zn^{2+} , Co^{2+} , Ni^{2+} ^{278,312,387}), or the species with the highest protein affinity according to the Irving-Williams Series (Mn^{2+} < Fe^{2+} , Fe^{3+} < Co^{2+} < Ni^{2+} < Cu^{2+} > Zn^{2+} ⁴¹³).

S3AN. Post-translational regulation.

M. genitalium post-translational regulation was reconstructed based on reports organized in four databases: ASAP¹³³, BRENDA⁵⁹, DrugBank¹⁹², and UniProt⁷¹.

S3AO. Enzyme pH dependence.

The pH dependence of the catalysis of each chemical reaction was reconstructed from reports organized in three databases: BRENDA⁵⁹, BioCyc¹⁸⁸, and UniProt⁷¹.

S3AP. Enzyme temperature dependence.

The temperature dependence of the catalysis of each chemical reaction was reconstructed from reports organized in three databases: BRENDA⁵⁹, BioCyc¹⁸⁸, and UniProt⁷¹.

S3AQ. Orphan reactions.

Orphan chemical reactions were added to the *M. genitalium* model to provide pathways for cellular growth.

S3AR. Cell mass.

The average *M. genitalium* cell mass was reconstructed from several sources. Zhao et al. reported that the average diameter of *M. genitalium* is 200 nm⁴⁴⁷. Assuming a density of 1.1 g ml⁻¹¹¹⁶, a fractional water content of 70%³⁹, and spherical geometry, we calculated that the average *M. genitalium* cell mass at the beginning of the cell cycle is 13.10 fg and the average dry weight is 3.93 fg. Table S3AS-S3AT describe the average composition of the *M. genitalium* dry mass.

S3AS. Dry biomass composition, by fraction.

The composition of the *M. genitalium* dry mass was reconstructed using a combination of several experimental observations. First, the *M. genitalium* DNA composition was reconstructed by calculating the sum of the masses of a single chromosome and free dNTPs (column H). This corresponds to a dry mass fraction of 16.88%, significantly higher than the 4% reported by Morowitz et al.²⁶¹. Second, the *M. genitalium* polyamine, vitamin, cofactor, and ion weight fractions were reconstructed by equating them to their observed *E. coli* weight fractions (column I)¹²⁰. Third, the *M. genitalium* free nucleotide composition was reconstructed based on the observed nucleotide concentrations (column I)^{5,25}. Lastly, the remainder of the *M. genitalium* dry mass (column K) was set equal to the relative compositions of RNA, protein, lipid, and carbohydrate observed by Morowitz et al. in *M. gallisepticum* (column C) weighted by column G²⁶¹. Table S3AT lists the molecular composition of each dry weight fraction. Table S3AV-S3BE describe the reconstruction of the molecular composition of each dry weight fraction.

S3AT. Dry biomass composition, by metabolite.

The *M. genitalium* dry mass composition was reconstructed from several sources. Table S3AS describes the composition of *M. genitalium* dry mass by weight fraction and its reconstruction. This table describes the molecular composition of the *M. genitalium* dry mass. Table S3AV-S3BE describe the reconstruction of the DNA/dNXP, RNA/NXP, protein/amino acid, lipid, polyamine, vitamin, cofactors, ion, and carbohydrate weight fractions.

S3AU. RNA composition.

The composition of the *M. genitalium* RNA pool (composition of m-, r-, s-, and tRNA) was reconstructed based on the reported *E. coli* RNA pool composition (column D,²⁷⁸) and adjusted to support *in silico* *M. genitalium* growth (column E): (1) The size of the *E. coli* sRNA pool was not reported. The *M. genitalium* sRNA pool was set to 2.4% to provide sufficient ribonuclease P and tmRNA to mature tRNA and abort stalled ribosomes. (2) The size of the tRNA pool was doubled to ensure that at least one copy of each tRNA species would be expressed at all times throughout the cell cycle in healthy wild type *M. genitalium* cells to support translation. (3) The size of the mRNA pool was doubled to reduce the variance in protein expression and the rate of *M. genitalium* growth. (4) The size of the rRNA pools were proportionally reduced to compensate for the increased sizes of the m-, s-, and tRNA pools.

S3AV. dNMP composition.

The composition of the *M. genitalium* dNMP pool was reconstructed based on the G/C content of the *M. genitalium* genome (see Table S3AW) and the molecular weight of DNA-incorporated “dNMPs” (column C).

S3AW. G/C content.

The *M. genitalium* genome is G/C poor. The G/C content was used to reconstruct the composition of *M. genitalium* dNMP pool (see Table S3AV).

S3AX. NMP composition.

The composition of the *M. genitalium* nucleotide pool was reconstructed based on the reported NMP composition of *M. gallisepticum* (columns D-F)²⁶¹.

S3AY. Amino acid composition.

The composition of the *M. genitalium* amino acid pool was reconstructed based on the observed amino acid composition of *M. gallisepticum* (columns D-F)²⁶¹ with missing data imputed (columns G-I) by (1) assigning trace amounts of unmeasured amino acids equally to cysteine and tryptophan, (2) assigning the reported aspartate composition equally to aspartate and asparagine, and (3) assigning the reported glutamic acid composition equally to glutamic acid and glutamine.

S3AZ. Lipid composition, by metabolite.

The *M. genitalium* lipid pool composition was reconstructed based on the experimentally observed *E. coli* lipid composition (column D): 15% cardiolipin, 5% diacylglycerol, 0% phosphoglycerol, 60% phosphoglycerol phosphate, 20% sterol; 43% 160, 33% 161, 24% 181²⁷⁷.

S3BA. Lipid composition, by class.

Experimentally observed lipid pool composition of *M. gallisepticum*²⁶¹.

S3BB. Polyamine composition.

The *M. genitalium* polyamine pool was reconstructed based on the experimentally observed composition of the *E. coli* polyamine pool (column C)²⁷⁸.

S3BC. Vitamin and cofactor composition.

The composition of the *M. genitalium* vitamin and cofactor pool was reconstructed based on the curated composition of the *E. coli* pool (column C)¹²⁰.

S3BD. Ion composition.

The composition of the *M. genitalium* ion pool was reconstructed based on the experimentally observed composition of the *E. coli* ion pool (column C)^{201,278}.

S3BE. Carbohydrate composition.

The composition of the *M. genitalium* carbohydrate pool was reconstructed by (1) setting the concentration of (p)ppGpp to its experimentally observed value¹³⁰, and (2) assigning the remainder of the carbohydrate pool to UDP-galactofuranose.

S3BF. Growth-associated maintenance energy, by process.

Calculated energy requirements of *E. coli* DNA, RNA, and protein synthesis²⁷⁸.

S3BG. Growth-associated maintenance energy and byproducts, by precursor.

The energetic substrates and byproducts required for *M. genitalium* growth were reconstructed based on the reported metabolic cost of *E. coli* growth, the reported costs of *E. coli* macromolecule synthesis (Table S3BF)²⁷⁸ and the known byproducts of replication, transcription, translation, and ATP hydrolysis.

S3BH. Growth-associated maintenance energy and byproducts.

The net metabolic effect of *M. genitalium* energy usage. See Table S3BG for additional information.

S3BI. Media composition.

The whole-cell model media composition (column AC) was reconstructed primarily based on the experimentally characterized composition of the components of SP-4 media (Columns I-AB, see also Table S3BJ). The modeled media includes additional metabolites present in the *M. pneumoniae* minimal media reported by Yus et al.⁴⁴⁵ as well as any additional metabolite required for *in silico* *M. genitalium* growth (AEPP, CO, DDCA, HDCA, HDCEA, LIPOATE, MN, MODB, NI, OCDCA, OCDCEA, THF, TTDCA, and TTDCEA). Supplemental metabolites were added either according to amounts reported by Yus et al. (nucleobases uracil, cytidine, guanine), amounts equal to similar components (ions Mn²⁺, Co²⁺, MoO₄²⁻; polyamines spermidine and putrescine; vitamin THF; and dipeptides), or amounts in equilibrium with partial gas pressures (dissolved oxygen and carbon dioxide).

S3BJ. SP-4 media composition.

The composition of each component of SP-4 media was reconstructed from product data sheets published by BD¹, Sigma Aldrich⁶, Solabia^{371,372,374-377}, and Thermo Scientific³⁴⁸. Amino acid and protein content was assigned to dipeptides.

S3BK. Antibiotics.

Antibiotics and putative multidrug transporter substrates included in the whole-cell model. Putative transporter substrates were identified by mapping experimentally observed transporter substrates onto homologous *M. genitalium* transporters. The susceptibility of *M. genitalium* to each antibiotic was reconstructed from Taylor-Robinson and Bébér³⁹⁸. The mechanism of action of each antibiotic was reconstructed from DrugBank¹⁹².

S3BL. Model parameters.

This table lists the name, state/process location, and size of each model parameter.

S3BM. Process - metabolites.

This table lists the metabolites which participate in each modeled cellular process.

S3BN. Process - gene products.

This table lists the gene products which participate in each modeled cellular process.

S3BO. Functionally unmodeled genes.

This table lists the genes whose RNA and/or protein products were not functionally modeled.

Fitted Parameters***S3BP. Computationally fit cell mass composition.***

This table lists this the computationally fit *M. genitalium* dry cell mass composition as well as the metabolic production and process byproduction of each metabolite.

S3BQ. Computationally fit gene expression.

This table lists the experimentally reconstructed and computationally fit expression of each RNA and protein gene product. Values listed are the average copy number per cell at the beginning of the life cycle.

D.2 Supplementary movies

This thesis contains one supplementary movie (Movie S1) to the *M. genitalium* whole-cell model described in Chapter 4. The movie illustrates the predicted life cycle of one in silico *M. genitalium* cell. The movie is reproduced with permission from Karr JR, Sanghvi JC, Macklin DN, Gutschow MV, Jacobs JM, Bolival B, Assad-Garcia N, Glass JI & Covert MW. A whole-cell computational model predicts phenotype from genotype. *Cell* 150, 389–401 (2012); copyright 2012 Elsevier Inc.

Bibliography

1. B. D. *BD Bionutrients Technical Manual*. 2011.
2. Kremling A, Bettenbrock K, and Gilles ED. “Analysis of global control of *Escherichia coli* carbohydrate uptake”. In: *BMC Sys Biol* 1 2007. P. 42.
3. Varma A and Palsson B/O. “Stoichiometric flux balance models quantitatively predict growth and metabolic by-product secretion in wild-type *Escherichia coli* W3110”. In: *Appl Environ Microbiol* 60 1994. Pp. 3724–3731.
4. P. F. Agris, F. A. Vendeix, and W. D. Graham. “tRNA’s wobble decoding of the genome: 40 years of modification”. In: *J Mol Biol* 366.1 2007. Pp. 1–13.
5. K. R. Albe, M. H. Butler, and B. E. Wright. “Cellular concentrations of enzymes and their substrates”. In: *J Theor Biol* 22143.2 1990. Pp. 163–95.
6. Sigma Aldrich. *CMRL-1066*. 2011.
7. D. E. Anderson, F. J. Gueiros-Filho, and H. P. Erickson. “Assembly dynamics of FtsZ rings in *Bacillus subtilis* and *Escherichia coli* and effects of FtsZ-regulating proteins”. In: *J Bacteriol* 186.17 2004. Pp. 5775–81.
8. C. Andreini, I. Bertini, G. Cavallaro, G. L. Holliday, and J. M. Thornton. “Metal-MACiE: a database of metals involved in biological catalysis”. In: *Bioinformatics* 25.16 2009. Pp. 2088–9.
9. L. C. Antunes, R. B. Ferreira, C. P. Lostroh, and E. P. Greenberg. “A mutational analysis defines *Vibrio fischeri* LuxR binding sites”. In: *J Bacteriol* 190.13 2008. Pp. 4392–7.
10. Burgard AP and Maranas CD. “Optimization-based framework for inferring and testing hypothesized metabolic objective functions”. In: *Biotechnol bioeng* 82 2003. Pp. 670–677.
11. J. C. Atlas, E. V. Nikolaev, S. T. Browning, and M. L. Shuler. “Incorporating genome-wide DNA sequence information into a dynamic whole-cell model of *Escherichia coli*: application to DNA replication”. In: *IET Syst Biol* 2.5 2008. Pp. 369–82.
12. Bodenmiller B et al. “Multiplexed mass cytometry profiling of cellular states perturbed by small-molecule regulators”. In: *Nat Biotechnol* 30 2012. Pp. 858–867.

13. L. Baars et al. "Defining the role of the *Escherichia coli* chaperone SecB using comparative proteomics". In: *J Biol Chem* 281.15 2006. Pp. 10024–34.
14. N. Baichoo, T. Wang, R. Ye, and J. D. Helmann. "Global analysis of the *Bacillus subtilis* Fur regulon and the iron starvation stimulon". In: *Mol Microbiol* 45.6 2002. Pp. 1613–29.
15. M. Bailly et al. "A single tRNA base pair mediates bacterial tRNA-dependent biosynthesis of asparagine". In: *Nucleic Acids Res* 34.21 2006. Pp. 6083–94.
16. W. W. Baldwin, R. Myer, N. Powell, E. Anderson, and A. L. Koch. "Buoyant density of *Escherichia coli* is determined solely by the osmolarity of the culture medium". In: *Arch Microbiol* 164.2 1995. Pp. 155–7.
17. M. F. Balish. "Subcellular structures of mycoplasmas". In: *Front Biosci* 11 2006. Pp. 2017–27.
18. M. F. Balish and D. C. Krause. "Mycoplasmas: a distinct cytoskeleton for wall-less bacteria". In: *J Mol Microbiol Biotechnol* 11.3-5 2006. Pp. 244–55.
19. G. N. Basturea and M. P. Deutscher. "Substrate specificity and properties of the *Escherichia coli* 16S rRNA methyltransferase, RsmE". In: *RNA* 13.11 2007. Pp. 1969–76.
20. Bennett BD et al. "Absolute metabolite concentrations and implied enzyme active site occupancy in *Escherichia coli*". In: *Nat Chem Biol* 5 2009. Pp. 593–599.
21. M. Bejerano-Sagie et al. "A checkpoint protein that scans the chromosome for damage at the start of sporulation in *Bacillus subtilis*". In: *Cell* 125.4 2006. Pp. 679–90.
22. J. D. Bendtsen, L. J. Jensen, N. Blom, G. Von Heijne, and S. Brunak. "Feature-based prediction of non-classical and leaderless protein secretion". In: *Protein Eng Des Sel* 17.4 2004. Pp. 349–56.
23. J. D. Bendtsen, H. Nielsen, G. von Heijne, and S. Brunak. "Improved prediction of signal peptides: SignalP 3.0". In: *J Mol Biol* 340.4 2004. Pp. 783–95.
24. B. D. Bennett et al. "Absolute metabolite concentrations and implied enzyme active site occupancy in *Escherichia coli*". In: *Nat Chem Biol* 5.8 2009. Pp. 593–9.
25. B. D. Bennett et al. "Absolute metabolite concentrations and implied enzyme active site occupancy in *Escherichia coli*". In: *Nat Chem Biol* 5.8 2009. Pp. 593–9.
26. D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers. "GenBank". In: *Nucleic Acids Res* 38.Database issue 2010. Pp. D46–51.

27. D. T. Beranek. "Distribution of methyl and ethyl adducts following alkylation with mono-functional alkylating agents". In: *Mutat Res* 231.1 1990. Pp. 11–30.
28. H. M. Berman et al. "The nucleic acid database. A comprehensive relational database of three-dimensional structures of nucleic acids". In: *Biophys J* 63.3 1992. Pp. 751–9.
29. J. A. Bernstein, A. B. Khodursky, P. H. Lin, S. Lin-Chao, and S. N. Cohen. "Global analysis of mRNA decay and abundance in *Escherichia coli* at single-gene resolution using two-color fluorescent DNA microarrays". In: *Proc Natl Acad Sci U S A* 99.15 2002. Pp. 9697–702.
30. S. Bjelland et al. "Oxidation of thymine to 5-formyluracil in DNA: mechanisms of formation, structural implications, and base excision by human cell free extracts". In: *Biochemistry* 34.45 1995. Pp. 14758–64.
31. G. R. Björk, B. Huang, O. P. Persson, and A. S. Byström. "A conserved modified wobble nucleoside (mcm5s2U) in lysyl-tRNA is required for viability in yeast". In: *RNA* 13.8 2007. Pp. 1245–55.
32. G. R. Björk et al. "Transfer RNA modification". In: *Annu Rev Biochem* 56 1987. Pp. 263–87.
33. E. P. Black et al. "Factors influencing germination of *Bacillus subtilis* spores via activation of nutrient receptors by high pressure". In: *Appl Environ Microbiol* 71.10 2005. Pp. 5879–87.
34. A. Blanchard and G. Browning. *Mycoplasmas: molecular biology, pathogenicity and strategies for control*. CRC Press, 2005.
35. K. Bloom and A. Joglekar. "Towards building a chromosome segregation machine". In: *Nature* 463.7280 2010. Pp. 446–56.
36. C. L. Borges, J. A. Parente, M. Pereira, and C. M. de Almeida Soares. "Identification of the GTPase superfamily in *Mycoplasma synoviae* and *Mycoplasma hyopneumoniae*". In: *Genet Mol Biol* 30.1 2007. Pp. 212–218.
37. E. Boye. "DisA, a busy bee that monitors chromosome integrity". In: *Cell* 125.4 2006. Pp. 641–3.
38. Bratton BP, Mooney RA, and Weisshaar JC. "Spatial distribution and diffusive motion of RNA polymerase in live *Escherichia coli*". In: *J Bacteriol* 193 2011. Pp. 5138–5146.
39. D. Bray. *Cell movements: from molecules to motility*. Garland Science, 2001.
40. S. Brenner. "Sequences and consequences". In: *Philos Trans R Soc London* 365.1537 2010. Pp. 207–12.

41. R. A. Britton. "Role of GTPases in bacterial ribosome assembly". In: *Annu Rev Microbiol* 63 2009. Pp. 155–76.
42. M. Brocchi, A. T. R. de Vasconcelos, and A. Zaha. "Restriction-modification systems in Mycoplasma spp". In: *Genet. Mol. Biol* 30.1 2007.
43. S. T. Browning, M. Castellanos, and M. L. Shuler. "Robust control of initiation of prokaryotic chromosome replication: essential considerations for a minimal cell". In: *Biotechnol Bioeng* 88.5 2004. Pp. 575–84.
44. R. C. Bruckner, P. L. Gunyuzlu, and R. L. Stein. "Coupled kinetics of ATP and peptide hydrolysis by Escherichia coli FtsH protease". In: *Biochemistry* 42.36 2003. Pp. 10843–52.
45. S. D. Bruner, D. P. Norman, and G. L. Verdine. "Structural basis for recognition and repair of the endogenous mutagen 8-oxoguanine in DNA". In: *Nature* 403.6772 2000. Pp. 859–66.
46. Z. Bryant. *Conversations with Zev Bryant*. 2010.
47. W. Bujalowski and M. J. Jezewska. "Interactions of Escherichia coli primary replicative helicase DnaB protein with single-stranded DNA. The nucleic acid does not wrap around the protein hexamer". In: *Biochemistry* 34.27 1995. Pp. 8513–9.
48. R. H. Burdon. *Genes and the environment*. CRC Press, 1999.
49. H. Cabedo et al. "The Escherichia coli trmE (mnme) gene, involved in tRNA modification, codes for an evolutionarily conserved GTPase with unusual biochemical properties". In: *EMBO J* 18.24 1999. Pp. 7063–76.
50. S. V. Cannon, A. Cummings, and G. W. Teebor. "5-Hydroxymethylcytosine DNA glycosylase activity in mammalian tissue". In: *Biochem Biophys Res Commun* 151.3 1988. Pp. 1173–9.
51. F. M. Carvalho et al. "DNA repair in reduced genome: the Mycoplasma model". In: *Gene* 360.2 2005. Pp. 111–9.
52. M. Castellanos, K. Kushiro, S. K. Lai, and M. L. Shuler. "A genomically/chemically complete module for synthesis of lipid membrane in a minimal cell". In: *Biotechnol Bioeng* 97.2 2007. Pp. 397–409.
53. M. Castellanos, D. B. Wilson, and M. L. Shuler. "A modular minimal cell model: purine and pyrimidine transport and metabolism". In: *Proc Natl Acad Sci U S A* 101.17 2004. Pp. 6681–6.

54. I. Catrein, R. Herrmann, A. Bosserhoff, and T. Ruppert. “Experimental proof for a signal peptidase I like activity in *Mycoplasma pneumoniae*, but absence of a gene encoding a conserved bacterial type I SPase”. In: *FEBS J* 272.11 2005. Pp. 2892–900.
55. Fowlkes CC et al. “A Conserved Developmental Patterning Network Produces Quantitatively Different Output in Multiple Species of *Drosophila*”. In: *PLoS Genet* 7 2011. e1002346.
56. I. Chambaud, H. Wróblewski, and A. Blanchard. “Interactions between mycoplasma lipoproteins and the host immune system”. In: *Trends Microbiol* 7.12 1999. Pp. 493–9.
57. I. Chambaud et al. “The complete genome sequence of the murine respiratory pathogen *Mycoplasma pulmonis*”. In: *Nucleic Acids Res* 29.10 2001. Pp. 2145–53.
58. S. Chandrasekaran and N. D. Price. “Probabilistic integrative modeling of genome-scale metabolic and regulatory networks in *Escherichia coli* and *Mycobacterium tuberculosis*”. In: *Proc Natl Acad Sci U S A* 107.41 2010. Pp. 17845–50.
59. A. Chang, M. Scheer, A. Grote, I. Schomburg, and D. Schomburg. “BRENDA, AMENDA and FRENDNA the enzyme information system: new content and tools in 2009”. In: *Nucleic Acids Res* 37. Database issue 2009. Pp. D588–92.
60. R. Chaudhry, A. K. Varshney, and P. Malhotra. “Adhesion proteins of *Mycoplasma pneumoniae*”. In: *Front Biosci* 12 2007. Pp. 690–9.
61. J. Chen et al. “MMDB: Entrez’s 3D-structure database”. In: *Nucleic Acids Res* 31.1 2003. Pp. 474–7.
62. Y. Chen, K. Bjornson, S. D. Redick, and H. P. Erickson. “A rapid fluorescence assay for FtsZ assembly indicates cooperative assembly with a dimer nucleus”. In: *Biophys J* 88.1 2005. Pp. 505–14.
63. I. G. Choi et al. “Crystal structure of a stress inducible protein from *Mycoplasma pneumoniae* at 2.85 Å resolution”. In: *J Struct Funct Genomics* 4.1 2003. Pp. 31–4.
64. S. Y. Choi, D. Reyes, M. Leelakriangsak, and P. Zuber. “The global regulator Spx functions in the control of organosulfur metabolism in *Bacillus subtilis*”. In: *J Bacteriol* 188.16 2006. Pp. 5741–51.
65. K. H. Choo, T. W. Tan, and S. Ranganathan. “SPdb—a signal peptide database”. In: *BMC Bioinformatics* 6 2005. P. 249.

66. S. Chrysogelos and J. Griffith. "Escherichia coli single-strand binding protein organizes single-stranded DNA in nucleosome-like units". In: *Proc Natl Acad Sci U S A* 79.19 1982. Pp. 5803–7.
67. M. S. Ciampi. "Rho-dependent terminators and transcription termination". In: *Microbiology* 152.Pt 9 2006. Pp. 2515–28.
68. S. D. Cline and P. C. Hanawalt. "Who's on first in the cellular response to DNA damage?" In: *Nat Rev Mol Cell Biol* 4.5 2003. Pp. 361–72.
69. P. H. Clingen et al. "Induction of cyclobutane pyrimidine dimers, pyrimidine(6-4)pyrimidone photoproducts, and Dewar valence isomers by natural sunlight in normal human mononuclear cells". In: *Cancer Res* 55.11 1995. Pp. 2245–8.
70. B. Collins-Sussman, B. W. Fitzpatrick, and C. M. Pilato. *Version Control with Subversion*. O'Reilly Media, 2011.
71. UniProt Consortium. "The Universal Protein Resource (UniProt) 2009". In: *Nucleic Acids Res* 37.Database issue 2009. Pp. D169–74.
72. B. Contreras-Moreira. "3D-footprint: a database for the structural analysis of protein-DNA complexes". In: *Nucleic Acids Res* 38.Database issue 2010. Pp. D91–7.
73. C. M. Cordova et al. "Identification of the origin of replication of the Mycoplasma pulmonis chromosome and its use in oriC replicative plasmids". In: *J Bacteriol* 184.19 2002. Pp. 5426–35.
74. S. J. Cordwell, D. J. Basseal, J. D. Pollack, and I. Humphrey-Smith. "Malate/lactate dehydrogenase in mollicutes: evidence for a multienzyme protein". In: *Gene* 195.2 1997. Pp. 113–20.
75. M. W. Covert, E. M. Knight, J. L. Reed, M. J. Herrgard, and B. Ø. Palsson. "Integrating high-throughput and computational data elucidates bacterial networks". In: *Nature* 429.6987 2004. Pp. 92–6.
76. M. W. Covert, C. H. Schilling, and B. O. Palsson. "Regulation of gene expression in flux balance models of metabolism". In: *J Theor Biol* 213.1 2001. Pp. 73–88.
77. M. W. Covert, N. Xiao, T. J. Chen, and J. R. Karr. "Integrating metabolic, transcriptional regulatory and signal transduction models in Escherichia coli". In: *Bioinformatics* 24.18 2008. Pp. 2044–50.

78. R. Cowan and S. N. Chiu. "A Stochastic Model of Fragment Formation When DNA Replicates". In: *Journal of Applied Probability* 31 1994. Pp. 301–308.
79. M. M. Cox. "Recombinational DNA repair in bacteria and the RecA protein". In: *Prog Nucleic Acid Res Mol Biol* 63 1999. Pp. 311–66.
80. D. M. Creasy and J. S. Cottrell. "Unimod: Protein modifications for mass spectrometry". In: *Proteomics* 4.6 2004. Pp. 1534–6.
81. G. M. Culver. "Assembly of the 30S ribosomal subunit". In: *Biopolymers* 68.2 2003. Pp. 234–49.
82. Di Ventura D, Lemerle C, Michalodimitrakis K, and Serrano L. "From in vivo to in silico biology and back". In: *Nature* 443 2006. Pp. 527–533.
83. Sarkar D, Le Meur N, and Gentleman R. "Using flowViz to visualize flow cytometry data". In: *Bioinformatics* 24 2008. Pp. 878–879.
84. R. E. Dalbey and G. von Heijne, eds. *Protein Targeting, Transport, and Translocation*. Academic Press, San Diego, 2002.
85. J. M. Daley and T. E. Wilson. "Rejoining of DNA double-strand breaks as a function of overhang length". In: *Mol Cell Biol* 25.3 2005. Pp. 896–906.
86. M. J. Davey and B. E. Funnell. "The P1 plasmid partition protein ParA. A role for ATP in site-specific DNA binding". In: *J Biol Chem* 269.47 1994. Pp. 29908–13.
87. T. Davidsen et al. "The comprehensive microbial resource". In: *Nucleic Acids Res* 38.Database issue 2010. Pp. D340–5.
88. E. H. Davidson et al. "A genomic regulatory network for development". In: *Science* 295.5560 2002. Pp. 1669–78.
89. A. Dawid, V. Croquette, M. Grigoriev, and F. Heslot. "Single-molecule study of RuvAB-mediated Holliday-junction migration". In: *Proc Natl Acad Sci U S A* 101.32 2004. Pp. 11611–6.
90. Chang DE et al. "Carbon nutrition of *Escherichia coli* in the mouse intestine". In: *Proc Natl Acad Sci USA* 101 2004. Pp. 7427–7432.
91. N. H. Dekker et al. "Thermophilic topoisomerase I on a single DNA molecule". In: *J Mol Biol* 329.2 2003. Pp. 271–82.
92. *Delta Mass: A Database of Protein Post Translational Modifications*.

93. I. A. Demina et al. "Proteome of the bacterium *Mycoplasma gallisepticum*". In: *Biochemistry (Mosc)* 74.2 2009. Pp. 165–74.
94. P. P. Dennis, M. Ehrenberg, D. Fange, and H. Bremer. "Varying rate of RNA chain elongation during rrn transcription in *Escherichia coli*". In: *J Bacteriol* 191.11 2009. Pp. 3740–6.
95. E. Deuerling et al. "Trigger Factor and DnaK possess overlapping substrate pools and binding specificities". In: *Mol Microbiol* 47.5 2003. Pp. 1317–28.
96. F. Diella, C. M. Gould, C. Chica, A. Via, and T. J. Gibson. "Phospho.ELM: a database of phosphorylation sites—update 2008". In: *Nucleic Acids Res* 36.Database issue 2008. Pp. D240–4.
97. M. Dizdaroglu. "Measurement of radiation-induced damage to DNA at the molecular level". In: *Int J Radiat Biol* 61.2 1992. Pp. 175–83.
98. M. M. Domach, S. K. Leung, R. E. Cahn, G. G. Cocks, and M. L. Shuler. "Computer model for glucose-limited growth of a single cell of *Escherichia coli* B/r-A". In: *Biotechnol Bioeng* 26.9 1984. P. 1140.
99. S. M. Doyle, O. Bilsel, and C. M. Teschke. "SecA folding kinetics: a large dimeric protein rapidly forms multiple native states". In: *J Mol Biol* 341.1 2004. Pp. 199–214.
100. Bandura DR et al. "Mass cytometry: technique for real time single cell multitarget immunoassay based on inductively coupled plasma time-of-flight mass spectrometry". In: *Anal Chem* 81 2009. Pp. 6813–6822.
101. M. F. Duffy, I. D. Walker, and G. F. Browning. "The immunoreactive 116 kDa surface protein of *Mycoplasma pneumoniae* is encoded in an operon". In: *Microbiology* 143 (Pt 10) 1997. Pp. 3391–402.
102. S. Dunin-Horkawicz et al. "MODOMICS: a database of RNA modification pathways". In: *Nucleic Acids Res* 34.Database issue 2006. Pp. D145–9.
103. F. Duong and W. Wickner. "Distinct catalytic roles of the SecYE, SecG and SecDFyajC subunits of preprotein translocase holoenzyme". In: *EMBO J* 16.10 1997. Pp. 2756–68.
104. K. Dybvig and L. L. Voelker. "Molecular biology of mycoplasmas". In: *Annu Rev Microbiol* 50 1996. Pp. 25–57.
105. M. R. Dyson, N. Mandal, and U. L. RajBhandary. "Relationship between the structure and function of *Escherichia coli* initiator tRNA". In: *Biochimie* 75.12 1993. Pp. 1051–60.

106. Bolton E, Wang Y, Thiessen PA, and Bryant SH. "Annual Reports in Computational Chemistry". In: ed. by Wheeler RA and Spellmeyer DC. Vol. 4. Washington DC: American Chemical Society, 2008. Chap. PubChem: Integrated Platform of Small Molecules and Biological Activities, pp. 217–241.
107. Gasteiger E et al. "The Proteomics Protocols Handbook". In: ed. by Walker JM. Totowa NJ: Humana Press, 2005. Chap. Protein Identification and Analysis Tools on the ExPASy Server, pp. 571–607.
108. G. Eberle et al. "1,N6-etheno-2'-deoxyadenosine and 3,N4-etheno-2'-deoxycytidine detected by monoclonal antibodies in lung and liver DNA of rats exposed to vinyl chloride". In: *Carcinogenesis* 10.1 1989. Pp. 209–12.
109. *EcoliHub*. 2010.
110. S. L. Eddins. "Automated Software Testing for MATLAB". In: *Computing in Science & Engineering* PP.99 2009. Pp. 48–54.
111. B. A. Edgar and K. J. Kim. "Cell biology. Sizing up the cell". In: *Science* 325.5937 2009. Pp. 158–9.
112. J. A. Eisen and P. C. Hanawalt. "A phylogenomic study of DNA repair genes, proteins, and processes". In: *Mutat Res* 435.3 1999. Pp. 171–213.
113. A. Endo and Y. Kurusu. "Identification of in vivo substrates of the chaperonin GroEL from *Bacillus subtilis*". In: *Biosci Biotechnol Biochem* 71.4 2007. Pp. 1073–7.
114. K. N. Erwin, S. Nakano, and P. Zuber. "Sulfate-dependent repression of genes that function in organosulfur metabolism in *Bacillus subtilis* requires Spx". In: *J Bacteriol* 187.12 2005. Pp. 4042–9.
115. Sayers EW et al. "Database resources of the National Center for Biotechnology Information". In: *Nucleic Acids Res* 38 2010. Pp. D5–16.
116. E. Fahy, M. Sud, D. Cotter, and S. Subramaniam. "LIPID MAPS online tools for lipid research". In: *Nucleic Acids Res* 35.Web Server issue 2007. W606–12.
117. C. Fan, H. J. Fromm, and T. A. Bobik. "Kinetic and functional analysis of L-threonine kinase, the PduX enzyme of *Salmonella enterica*". In: *J Biol Chem* 284.30 2009. Pp. 20240–8.
118. P. Fariselli, G. Finocchiaro, and R. Casadio. *SPEP: a Signal Peptide Predictor Based on Neural Network Systems*. 2003.

119. A. M. Feist, M. J. Herrgård, I. Thiele, J. L. Reed, and B. O. Palsson. “Reconstruction of biochemical networks in microorganisms”. In: *Nat Rev Microbio* 7.2 2009. Pp. 129–43.
120. A. M. Feist et al. “A genome-scale metabolic reconstruction for Escherichia coli K-12 MG1655 that accounts for 1260 ORFs and thermodynamic information”. In: *Mol Syst Biol* 3 2007. P. 121.
121. *FindMod*.
122. C. M. Fraser et al. “The minimal gene complement of *Mycoplasma genitalium*”. In: *Science* 270.5235 1995. Pp. 397–403.
123. E. C. Friedberg et al. *DNA repair and mutagenesis*. ASM Press, 2006.
124. J. C. Fromme, S. D. Bruner, W. Yang, M. Karplus, and G. L. Verdine. “Product-assisted catalysis in base-excision DNA repair”. In: *Nat Struct Biol* 10.3 2003. Pp. 204–11.
125. L. Furchtgott, N. S. Wingreen, and K. C. Huang. “Mechanisms for maintaining cell shape in rod-shaped Gram-negative bacteria”. In: *Mol Microbiol* 81.2 2011. Pp. 340–53.
126. J. Gallagher, N. N. Kaderbhai, and M. A. Kaderbhai. “Kinetic constants of signal peptidase I using cytochrome b5 as a precursor substrate”. In: *Biochim Biophys Acta* 1550.1 2001. Pp. 1–5.
127. J. S. Garavelli. “The RESID Database of Protein Modifications as a resource and annotation tool”. In: *Proteomics* 4.6 2004. Pp. 1527–33.
128. J. L. Gardy et al. “PSORTb v.2.0: expanded prediction of bacterial protein subcellular localization and insights gained from comparative proteome analysis”. In: *Bioinformatics* 21.5 2005. Pp. 617–23.
129. D. Gasparutto, E. Muller, S. Boiteux, and J. Cadet. “Excision of the oxidatively formed 5-hydroxyhydantoin and 5-hydroxy-5-methylhydantoin pyrimidine lesions by Escherichia coli and Saccharomyces cerevisiae DNA N-glycosylases”. In: *Biochim Biophys Acta* 1790.1 2009. Pp. 16–24.
130. M. L. Gatewood and G. H. Jones. “(p)ppGpp inhibits polynucleotide phosphorylase from streptomycetes but not from Escherichia coli and increases the stability of bulk mRNA in Streptomyces coelicolor”. In: *J Bacteriol* 192.17 2010. Pp. 4275–80.
131. D. G. Gibson et al. “Complete chemical synthesis, assembly, and cloning of a *Mycoplasma genitalium* genome”. In: *Science* 319.5867 2008. Pp. 1215–20.

132. D. G. Gibson et al. "Creation of a bacterial cell controlled by a chemically synthesized genome". In: *Science* 329.5987 2010. Pp. 52–6.
133. J. D. Glasner et al. "ASAP, a systematic annotation package for community analysis of genomes". In: *Nucleic Acids Res* 31.1 2003. Pp. 147–51.
134. J. I. Glass et al. "Essential genes of a minimal bacterium". In: *Proc Natl Acad Sci U S A* 103.2 2006. Pp. 425–30.
135. M. Gomi, M. Sonoyama, and S. Mitaku. "High performance system for signal peptide prediction: SOSUIsignal". In: *Chem-Bio Info J* 4 2004. Pp. 142–7.
136. J. Gore et al. "Mechanochemical analysis of DNA gyrase using rotor bead tracking". In: *Nature* 439.7072 2006. Pp. 100–4.
137. D. E. Green, T. W. Morris, J. Green, J. E. Jr Cronan, and J. R. Guest. "Purification and properties of the lipoate protein ligase of Escherichia coli". In: *Biochem J* 309 (Pt 3) 1995. Pp. 853–62.
138. J. E. Grimwade, V. T. Ryan, and A. C. Leonard. "IHF redistributes bound initiator protein, DnaA, on supercoiled oriC of Escherichia coli". In: *Mol Microbiol* 35.4 2000. Pp. 835–44.
139. C. O. Gualerzi and C. L. Pon. "Initiation of mRNA translation in prokaryotes". In: *Biochemistry* 26.25 1990. Pp. 5881–9.
140. M. Güell et al. "Transcriptome complexity in a genome-reduced bacterium". In: *Science* 326.5957 2009. Pp. 1268–71.
141. G. Guetens, G. De Boeck, M. Highley, A. T. van Oosterom, and E. A. de Bruijn. "Oxidative DNA damage: biological significance and methods of analysis". In: *Crit Rev Clin Lab Sci* 39.4-5 2002. Pp. 331–457.
142. M. Gulston, J. Fulford, T. Jenner, C. de Lara, and P. O'Neill. "Clustered DNA damage induced by gamma radiation in human fibroblasts (HF19), hamster (V79-4) cells and plasmid DNA is revealed as Fpg and Nth sensitive sites". In: *Nucleic Acids Res* 30.15 2002. Pp. 3464–72.
143. T. Guo, S. Hua, X. Ji, and Z. Sun. "DBSubLoc: database of protein subcellular localization". In: *Nucleic Acids Res* 32.Database issue 2004. Pp. D122–4.
144. X. C. Guo, P. T. Ravi Rajagopalan, and D. Pei. "A direct spectrophotometric assay for peptide deformylase". In: *Anal Biochem* 273.2 1999. Pp. 298–304.

145. N. Gupta et al. "Whole proteome analysis of post-translational modifications: applications of mass-spectrometry for proteogenomic annotation". In: *Genome Res* 17.9 2007. Pp. 1362–77.
146. Kitano H. "Systems biology: a brief overview". In: *Science* 295 2002. Pp. 1662–1664.
147. C. S. Hamilton et al. "Mechanistic investigations of the pseudouridine synthase RluA using RNA containing 5-fluorouridine". In: *Biochemistry* 45.39 2006. Pp. 12029–38.
148. F. G. Hansen, B. B. Christensen, and T. Atlung. "The initiator titration model: computer simulation of chromosome and minichromosome control". In: *Res Microbiol* 142.2-3 1991. Pp. 161–7.
149. K. Haraguchi et al. "Synthesis and characterization of oligonucleotides containing formamidopyrimidine lesions (Fapy.dA, Fapy.dG) at defined sites". In: *Nucleic Acids Res Suppl.1* 2001. Pp. 129–30.
150. Z. Hatahet, Y. W. Kow, A. A. Purmal, R. P. Cunningham, and S. S. Wallace. "New substrates for old enzymes. 5-Hydroxy-2'-deoxycytidine and 5-hydroxy-2'-deoxyuridine are substrates for Escherichia coli endonuclease III and formamidopyrimidine DNA N-glycosylase, while 5-hydroxy-2'-deoxyuridine is a substrate for uracil DNA N-glycosylase". In: *J Biol Chem* 269.29 1994. Pp. 18814–20.
151. G. von Heijne. "A new method for predicting signal sequence cleavage sites". In: *Nucleic Acids Res* 14.11 1986. Pp. 4683–90.
152. A. R. Hesketh et al. "Primary and secondary metabolism, and post-translational protein modifications, as portrayed by proteomic analysis of Streptomyces coelicolor". In: *Mol Microbiol* 46.4 2002. Pp. 917–32.
153. C. M. Hester and J. Lutkenhaus. "Soj (ParA) DNA binding is mediated by conserved arginines and is essential for plasmid segregation". In: *Proc Natl Acad Sci U S A* 104.51 2007. Pp. 20326–31.
154. T. Hirokawa, S. Boon-Chieng, and S. Mitaku. "SOSUI: classification and secondary structure prediction system for membrane proteins". In: *Bioinformatics* 14.4 1998. Pp. 378–9.
155. Morowitz HJ, Tourtellotte ME, Guild WR, Castro E, and Woese C. "The chemical composition and submicroscopic morphology of *Mycoplasma gallisepticum*, Avian PPLO 5969". In: *J Mol Biol* 4 1962. Pp. 93–103.
156. S. Huecas et al. "The interactions of cell division protein FtsZ with guanine nucleotides". In: *J Biol Chem* 282.52 2007. Pp. 37515–28.

157. M. I. Hutchings, T. Palmer, D. J. Harrington, and I. C. Sutcliffe. “Lipoprotein biogenesis in Gram-positive bacteria: knowing when to hold ‘em, knowing when to fold ‘em”. In: *Trends Microbiol* 17.1 2009. Pp. 13–21.
158. M. Huynen, B. Snel, W. 3rd Lathe, and P. Bork. “Predicting protein function by genomic context: quantitative evaluation and qualitative inferences”. In: *Genome Res* 10.8 2000. Pp. 1204–10.
159. T Ideker, T Galitski, and L Hood. “A new approach to decoding life: systems biology”. In: *Annu Rev Genomics Hum Genet* 2 2001. Pp. 343–372.
160. S. Imam, Z. Chen, S. Roos, and M. Pohlschröder. *Identification of diverse Gram-positive type IV pili and development of PilFind software for type IV pilin prediction*. 2009.
161. K. Ito and Y. Akiyama. “Cellular functions, mechanism of action, and regulation of FtsH protease”. In: *Annu Rev Microbiol* 59 2005. Pp. 211–31.
162. Schellenberger J, Park JO, Conrad TM, and Palsson B/O. “BiGG: a Biochemical Genetic and Genomic knowledgebase of large scale metabolic reconstructions”. In: *BMC Bioinformatics* 11 2010. P. 213.
163. J. D. Jaffe, H. C. Berg, and G. M. Church. “Proteogenomic mapping as a complementary method to perform genome annotation”. In: *Proteomics* 4.1 2004. Pp. 59–77.
164. J. D. Jaffe et al. “The complete genome and proteome of Mycoplasma mobile”. In: *Genome Res* 14.8 2004. Pp. 1447–61.
165. Courtright JB and Henning U. “Malate dehydrogenase mutants in *Escherichia coli* K-12”. In: *J Bacteriol* 102 1970. Pp. 722–728.
166. Russell JB and Cook GM. “Energetics of bacterial growth: balance of anabolic and catabolic reactions”. In: *Microbiol Rev* 59 1995. Pp. 48–62.
167. Orth JD et al. “A comprehensive genome-scale reconstruction of *Escherichia coli* metabolism—2011”. In: *Mol Syst Biol* 7 2011. P. 535.
168. R. B. Jensen and L. Shapiro. “Cell-cycle-regulated expression and subcellular localization of the Caulobacter crescentus SMC chromosome structural protein”. In: *J Bacteriol* 185.10 2003. Pp. 3068–75.
169. Glass JI et al. “Essential genes of a minimal bacterium”. In: *Proc Natl Acad Sci USA* 77 2006. Pp. 1175–81.

170. Ingraham JL, Maal/oe O, and Neidhardt FC. *Growth of the Bacterial Cell*. Sunderland, MA: Sinauer Associates, Inc, 1983.
171. Reed JL, Vo TD, Schilling CH, and Palsson B/O. “An expanded genome-scale model of *Escherichia coli* K-12 (iJR904 GSM/GPR)”. In: *Genome Biol* 4 2003. R54.
172. Reed JL et al. “Systems approach to refining genome annotation”. In: *Proc Natl Acad Sci USA* 103 2006. Pp. 17480–17484.
173. Lee JM, Gianchandani EP, Eddy JA, and Papin JA. “Dynamic analysis of integrated signaling, metabolic, and regulatory networks”. In: *PLoS Comput Biol* 4 2008. e1000086.
174. Karr JR, Sanghvi JC, Macklin DN, Arora A, and Covert MW. “WholeCellKB: model organism databases for comprehensive whole-cell models”. In: *Nucleic Acids Res* 41 2013. Pp. D787–92.
175. Karr JR et al. “A Whole-Cell Computational Model Predicts Phenotype from Genotype”. In: *Cell* 150 2012. Pp. 389–401.
176. S. Jun and B. Mulder. “Entropy-driven spatial organization of highly confined polymers: lessons for the bacterial chromosome”. In: *Proc Natl Acad Sci U S A* 103.33 2006. Pp. 12388–93.
177. Bettenbrock K et al. “A quantitative approach to catabolite repression in *Escherichia coli*”. In: *J Biol Chem* 281 2006. Pp. 2578–2584.
178. Bettenbrock K et al. “Correlation between growth rates, EIIA^{Crr} phosphorylation, and intracellular cyclic AMP levels in *Escherichia coli* K-12”. In: *J Bacteriol* 189 2007. Pp. 6891–6900.
179. Smallbone K, Simeonidis E, Broomhead DS, and Kell DB. “Something from nothing - bridging the gap between constraint-based and kinetic modeling”. In: *FEBS J* 274 2007. Pp. 5576–5585.
180. Yugi K, Nakayama Y, Kinoshita A, and Tomita M. “Hybrid dynamic/static method for large-scale simulation of metabolism”. In: *Theor Biol Med Model* 2 2005. P. 42.
181. C. M. Kaiser et al. “Real-time observation of trigger factor function on translating ribosomes”. In: *Nature* 444.7118 2006. Pp. 455–60.
182. L. Käll, A. Krogh, and E. L. Sonnhammer. “A combined transmembrane topology and signal peptide prediction method”. In: *J Mol Biol* 338.5 2004. Pp. 1027–36.

183. M. Kanehisa et al. "KEGG for linking genomes to life and the environment". In: *Nucleic Acids Res* 36.Database issue 2008. Pp. D480–4.
184. R. Kaplan and D. Apirion. "Decay of ribosomal ribonucleic acid in Escherichia coli cells starved for various nutrients". In: *J Biol Chem* 250.8 1975. Pp. 3174–8.
185. M. A. Karymov et al. "Structure, dynamics, and branch migration of a DNA Holliday junction: a single-molecule fluorescence and modeling study". In: *Biophys J* 95.9 2008. Pp. 4372–83.
186. G. W. C. Kaye and T. H. Laby. *Tables of physical and chemical constants*. Longman Sc & Tech, 1995.
187. M. J. Kerner et al. "Proteome-wide analysis of chaperonin-dependent protein folding in Escherichia coli". In: *Cell* 122.2 2005. Pp. 209–20.
188. I. M. Keseler et al. "EcoCyc: a comprehensive view of Escherichia coli biology". In: *Nucleic Acids Res* 37.Database issue 2009. Pp. D464–70.
189. T. Kimura et al. "Ribosome-small-subunit-dependent GTPase interacts with tRNA-binding sites on the ribosome". In: *J Mol Biol* 381.2 2008. Pp. 467–77.
190. *Kinetikon*.
191. L. Kittler, A. Bell, B. C. Baguley, and G. Löber. "Inhibition of restriction endonucleases by DNA sequence-reading ligands". In: *Biochem Mol Biol Int* 40.2 1996. Pp. 263–72.
192. C. Knox et al. "DrugBank 3.0: a comprehensive resource for 'omics' research on drugs". In: *Nucleic Acids Res* 39.Database issue 2011. Pp. D1035–41.
193. A. de Kok, A. F. Hengeveld, A. Martin, and A. H. Westphal. "The pyruvate dehydrogenase multi-enzyme complex from Gram-negative bacteria". In: *Biochim Biophys Acta* 1385.2 1998. Pp. 353–66.
194. S. Kol, N. Nouwen, and A. J. Driessens. "Mechanisms of YidC-mediated insertion and assembly of multimeric membrane protein complexes". In: *J Biol Chem* 283.46 2008. Pp. 31269–73.
195. A. Kornberg and T. Baker. *DNA Replication*. University Science Books, 2005.
196. D. C. Krause. "Mycoplasma pneumoniae cytadherence: organization and assembly of the attachment organelle". In: *Trends Microbiol* 6.1 1998. Pp. 15–8.
197. D. C. Krause and M. F. Balish. "Cellular engineering in a minimal microbe: structure and assembly of the terminal organelle of Mycoplasma pneumoniae". In: *Mol Microbiol* 51.4 2004. Pp. 917–24.

198. D. C. Krause and M. F. Balish. "Structure, function, and assembly of the terminal organelle of *Mycoplasma pneumoniae*". In: *FEMS Microbiol Lett* 198.1 2001. Pp. 1–7.
199. D. C. Krause et al. "Transposon mutagenesis reinforces the correlation between *Mycoplasma pneumoniae* cytoskeletal protein HMW2 and cytadherence". In: *J Bacteriol* 179.8 1997. Pp. 2668–77.
200. K. A. Krebes, L. B. Dirksen, and D. C. Krause. "Phosphorylation of *Mycoplasma pneumoniae* cytadherence-accessory proteins in cell extracts". In: *J Bacteriol* 177.15 1995. Pp. 4571–4.
201. R. Krishnaswamy and D. B. Wilson. "Construction and characterization of an *Escherichia coli* strain genetically engineered for Ni(II) bioaccumulation". In: *Appl Environ Microbiol* 66.12 2000. Pp. 5383–6.
202. D. W. Kufe, E. Frei, and J. F. Holland. *Cancer medicine-6 review*. BC Decker Inc, 2003.
203. S. Kühner et al. "Proteome organization in a genome-reduced bacterium". In: *Science* 326.5957 2009. Pp. 1235–40.
204. I. Kumagai, K. Watanabe, and T. Oshima. "A thermostable tRNA (guanosine-2')-methyltransferase from *Thermus thermophilus* HB27 and the effect of ribose methylation on the conformational stability of tRNA". In: *J Biol Chem* 257.13 1982. Pp. 7388–95.
205. M. D. Kumar et al. "ProTherm and ProNIT: thermodynamic databases for proteins and protein-nucleic acid interactions". In: *Nucleic Acids Res* 34.Database issue 2006. Pp. D204–6.
206. S. Kunzelmann, C. Morris, A. P. Chavda, J. F. Eccleston, and M. R. Webb. "Mechanism of Interaction between Single-Stranded DNA Binding Protein and DNA". In: *Biochemistry* 49.5 2010. Pp. 843–52.
207. I. Kurth and M. O'Donnell. *Replisome Dynamics during Chromosome Duplication*. ASM Press, 2009.
208. C. Lartigue et al. "Creating bacterial strains from genomes that have been cloned and engineered in yeast". In: *Science* 325.5948 2009. Pp. 1693–6.
209. C. Lartigue et al. "Genome transplantation in bacteria: changing one species to another". In: *Science* 317.5838 2007. Pp. 632–8.
210. C. T. Lauhon, W. M. Erwin, and G. N. Ton. "Substrate specificity for 4-thiouridine modification in *Escherichia coli*". In: *J Biol Chem* 279.22 2004. Pp. 23022–9.

211. J. E. LeClerc, A. Borden, and C. W. Lawrence. “The thymine-thymine pyrimidine-pyrimidone (6-4) ultraviolet light photoproduct is highly mutagenic and specifically induces 3' thymine-to-cytosine transitions in *Escherichia coli*”. In: *Proc Natl Acad Sci U S A* 88.21 1991. Pp. 9685–9.
212. I. Lee and C. K. Suzuki. “Functional mechanics of the ATP-dependent Lon protease- lessons from endogenous protein and synthetic peptide substrates”. In: *Biochim Biophys Acta* 1784.5 2008. Pp. 727–35.
213. T. J. Lee et al. “BioWarehouse: a bioinformatics database warehouse toolkit”. In: *BMC Bioinformatics* 7 2006. P. 170.
214. T. Y. Lee et al. “dbPTM: an information repository of protein post-translational modification”. In: *Nucleic Acids Res* 34.Database issue 2006. Pp. D622–7.
215. So LH et al. “General properties of transcriptional time series in *Escherichia coli*”. In: *Nat Genet* 43 2011. Pp. 554–560.
216. Z. Li, M. J. Trimble, Y. V. Brun, and G. J. Jensen. “The structure of FtsZ filaments in vivo suggests a force-generating role in cell division”. In: *EMBO J* 26.22 2007. Pp. 4694–708.
217. K. Lind, B. O. Lindhardt, H. J. Schütten, J. Blom, and C. Christiansen. “Serological cross-reactions between *Mycoplasma genitalium* and *Mycoplasma pneumoniae*”. In: *J Clin Microbiol* 20.6 1984. Pp. 1036–43.
218. T. Lindahl and D. E. Barnes. “Repair of endogenous DNA damage”. In: *Cold Spring Harb Symp Quant Biol* 65 2000. Pp. 127–33.
219. J. E. Lindsley. *DNA Topology: Supercoiling and Linking*. John Wiley & Sons, Ltd, 2005.
220. B. Liu and B. M. Alberts. “Head-on collision between a DNA replication apparatus and RNA polymerase transcription complex”. In: *Science* 267.5201 1995. Pp. 1131–7.
221. M. Lluch-Senar, E. Querol, and J. Piñol. “Cell division in a minimal bacterium in the absence of ftsZ”. In: *Mol Microbiol* 78.2 2010. Pp. 278–89.
222. J. R. Lobry. “Origin of Replication of *Mycoplasma genitalium*”. In: *Science* 272.5262 1996. Pp. 745–6.
223. P. Lund, ed. *Molecular Chaperones in the Cell*. Oxford University Press, New York, 2001.

224. F. Lustig et al. “The nucleotide in position 32 of the tRNA anticodon loop determines ability of anticodon UCC to discriminate among glycine codons”. In: *Proc Natl Acad Sci U S A* 90.8 1993. Pp. 3343–7.
225. Güell M et al. “Transcriptome complexity in a genome-reduced bacterium”. In: *Science* 326 2009. Pp. 1268–1271.
226. Kanehisa M and Goto S. “KEGG: kyoto encyclopedia of genes and genomes”. In: *Nucleic Acids Res* 28 2000. Pp. 27–30.
227. Kanehisa M, Goto S, Sato Y, Furumichi M, and Tanabe M. “KEGG for integration and interpretation of large-scale molecular datasets”. In: *Nucleic Acids Res* 40 2012. Pp. D109–114.
228. Meyer M, Munzner T, DePace A, and Pfister H. “MulteeSum: A Tool for Comparative Spatial and Temporal Gene Expression Data”. In: *IEEE Trans Vis Comput Graph* 16 2010. Pp. 908–917.
229. Scheer M et al. “BRENDA, the enzyme information system in 2011”. In: *Nucleic Acids Res* 39 2011. Pp. D670–676.
230. L. Ma et al. “Short tandem repeat sequences in the *Mycoplasma genitalium* genome and their use in a multilocus genotyping system”. In: *BMC Microbiol* 8 2008. P. 130.
231. B. Macek et al. “The serine/threonine/tyrosine phosphoproteome of the model bacterium *Bacillus subtilis*”. In: *Mol Cell Proteomics* 6.4 2007. Pp. 697–707.
232. D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova. “Entrez Gene: gene-centered information at NCBI”. In: *Nucleic Acids Res* 39.Database issue 2011. Pp. D52–7.
233. J. Maniloff, R. McElheney, L. Finch, and J. Baseman, eds. *Phylogeny of Mycoplasmas*. American Society of Microbiology Press: Washington, DC, 1992.
234. C. Margulies and J. M. Kaguni. “Ordered and sequential binding of DnaA protein to oriC, the chromosomal origin of *Escherichia coli*”. In: *J Biol Chem* 271.29 1996. Pp. 17035–40.
235. E. Martínez-Salas, J. A. Martín, and M. Vicente. “Relationship of *Escherichia coli* density to growth rate and cell age”. In: *J Bacteriol* 147.1 1981. Pp. 97–100.
236. Marvin.
237. *MATLAB Compiler*. 2012.

238. P. de Matos et al. "Chemical Entities of Biological Interest: an update". In: *Nucleic Acids Res* 38. Database issue 2010. Pp. D249–54.
239. *Maui Cluster Scheduler*. 2012.
240. D. Mazel, S. Pochet, and P. Marlière. "Genetic characterization of polypeptide deformylase, a distinctive enzyme of eubacterial translation". In: *EMBO J* 13.4 1994. Pp. 914–23.
241. R. M. McCarron and Y. F. Chang. "Aspartokinase of Streptococcus mutans: purification, properties, and regulation". In: *J Bacteriol* 134.2 1978. Pp. 483–91.
242. W. R. McClure. "Mechanism and control of transcription initiation in prokaryotes". In: *Annu Rev Biochem* 54 1985. Pp. 171–204.
243. P. McGlynn and C. P. Guy. "Replication forks blocked by protein-DNA complexes have limited stability in vitro". In: *J Mol Biol* 381.2 2008. Pp. 249–55.
244. C. L. McGowin, L. Ma, D. H. Martin, and R. B. Pyles. "Mycoplasma genitalium-encoded MG309 activates NF-kappaB via Toll-like receptors 2 and 6 to elicit proinflammatory cytokine secretion from human genital epithelial cells". In: *Infect Immun* 77.3 2009. Pp. 1175–81.
245. B. McLeod. "In vitro characterization of the ParA family protein Soj from bacillus subtilis". PhD thesis. The University of British Columbia, 2008.
246. Smoot ME, Ono K, Ruscheinski J, Wang PL, and Ideker T. "Cytoscape 2.8: new features for data integration and network visualization". In: *Bioinformatics* 27 2011. Pp. 431–432.
247. T. Meinnel, Y. Mechulam, and S. Blanquet. "Methionine as translation start signal: a review of the enzymes of the pathway in Escherichia coli". In: *Biochimie* 75.12 1993. Pp. 1061–75.
248. T. Meinnel, Y. Mechulam, and S. Blanquet. "Methionine as translation start signal: a review of the enzymes of the pathway in Escherichia coli". In: *Biochimie* 75.12 1993. Pp. 1061–75.
249. W. Messer. "The bacterial replication initiator DnaA. DnaA and oriC, the bacterial mode to initiate DNA replication". In: *FEMS Microbiol Rev* 26.4 2002. Pp. 355–74.
250. S. Meyer, A. Wittinghofer, and W. Versées. "G-domain dimerization orchestrates the tRNA wobble modification reaction in the MnmE/GidA complex". In: *J Mol Biol* 392.4 2009. Pp. 910–22.
251. F. Miao, M. Bouziane, and T. R. O'Connor. "Interaction of the recombinant human methylpurine-DNA glycosylase (MPG protein) with oligodeoxyribonucleotides containing either hypoxanthine or abasic sites". In: *Nucleic Acids Res* 26.17 1998. Pp. 4034–41.

252. M. L. Miller et al. “NetPhosBac - a predictor for Ser/Thr phosphorylation sites in bacterial proteins”. In: *Proteomics* 9.1 2009. Pp. 116–25.
253. P. B. Miller, D. G. Scraba, M. Leyritz-Wills, K. L. Maltman, and R. A. Warren. “Formation and possible functions of alpha-putrescylthymine in bacteriophage phi W-14 DNA: analysis of bacteriophage mutants with decreased levels of alpha-putrescylthymine in their DNAs”. In: *J Virol* 47.3 1983. Pp. 399–405.
254. M. Mir et al. “Optical measurement of cycle-dependent cell growth”. In: *Proc Natl Acad Sci U S A* 108.32 2011. Pp. 13124–9.
255. E. V. Mirkin, D. Castro Roa, E. Nudler, and S. M. Mirkin. “Transcription regulatory elements are punctuation marks for DNA replication”. In: *Proc Natl Acad Sci U S A* 103.19 2006. Pp. 7276–81.
256. E. V. Mirkin and S. M. Mirkin. “Mechanisms of transcription-replication collisions in bacteria”. In: *Mol Cell Biol* 25.3 2005. Pp. 888–95.
257. E. V. Mirkin and S. M. Mirkin. “Replication fork stalling at natural impediments”. In: *Microbiol Mol Biol Rev* 71.1 2007. Pp. 13–35.
258. R. V. Misra, R. S. Horler, W. Reindl, I. I. Goryanin, and G. H. Thomas. “EchoBASE: an integrated post-genomic database for Escherichia coli”. In: *Nucleic Acids Res* 33.Database issue 2005. Pp. D329–33.
259. D. L. Mitchell, J. Jen, and J. E. Cleaver. “Sequence specificity of cyclobutane pyrimidine dimers in DNA treated with solar (ultraviolet B) radiation”. In: *Nucleic Acids Res* 20.2 1992. Pp. 225–9.
260. H. J. Morowitz. *Beginnings of Cellular Life*. Yale University Press, New Haven CT, 1992.
261. H. J. Morowitz, M. E. Tourtellotte, W. R. Guild, E. Castro, and C. Woese. “The chemical composition and submicroscopic morphology of Mycoplasma gallisepticum, Avian PPLO 5969”. In: *J Mol Biol* 4.2 1962. Pp. 93–103.
262. M. Moser and T. O’Brien. *Hudson Extensible Continuous Integration Server*. Sonatype, Inc., 2011.
263. P. F. Mühlradt, M. Kiess, H. Meyer, R. Süßmuth, and G. Jung. “Isolation, structure elucidation, and synthesis of a macrophage stimulatory lipopeptide from Mycoplasma fermentans acting at picomolar concentration”. In: *J Exp Med* 185.11 1997. Pp. 1951–8.

264. K. S. Murakami, S. Masuda, and S. A. Darst. "Structural basis of transcription initiation: RNA polymerase holoenzyme at 4 Å resolution". In: *Science* 296.5571 2002. Pp. 1280–4.
265. N. E. Murray. "Type I restriction systems: sophisticated molecular machines (a legacy of Bertani and Weigle)". In: *Microbiol Mol Biol Rev* 64.2 2000. Pp. 412–34.
266. O. Musatovova, S. Dhandayuthapani, and J. B. Baseman. "Transcriptional heat shock response in the smallest known self-replicating cell, *Mycoplasma genitalium*". In: *J Bacteriol* 188.8 2006. Pp. 2845–55.
267. A. R. Mushegian and E. V. Koonin. "A minimal gene set for cellular life derived by comparison of complete bacterial genomes". In: *Proc Natl Acad Sci U S A* 93.19 1996. Pp. 10268–73.
268. A. Muto, Y. Andachi, H. Yuzawa, F. Yamao, and S. Osawa. "The organization and evolution of transfer RNA genes in *Mycoplasma capricolum*". In: *Nucleic Acids Res* 18.17 1990. Pp. 5037–43.
269. Covert MW and Palsson B/O. "Transcriptional regulation in constraints-based metabolic models of *Escherichia coli*". In: *J Biol Chem* 277 2002. Pp. 28058–28064.
270. Kotecha N, Krutzik PO, and Irish JM. "Web-based analysis and publication of flow cytometry experiments". In: *Curr Protoc Cytom* Chapter 10 2010. Unit 10.17.
271. K. Nakanishi and O. Nureki. "Recent progress of structural biology of tRNA processing and modification". In: *Mol Cells* 19.2 2005. Pp. 157–66.
272. K. Nakanishi et al. "Structural basis for lysidine formation by ATP pyrophosphatase accompanied by a lysine-specific loop and a tRNA-recognition domain". In: *Proc Natl Acad Sci U S A* 102.21 2005. Pp. 7487–92.
273. M. M. Nakano et al. "Promoter recognition by a complex of Spx and the C-terminal domain of the RNA polymerase alpha subunit". In: *PLoS One* 5.1 2010. e8664.
274. S. Nakano, K. N. Erwin, M. Ralle, and P. Zuber. "Redox-sensitive transcriptional control by a thiol/disulphide switch in the global regulator, Spx". In: *Mol Microbiol* 55.2 2005. Pp. 498–510.
275. S. Nakano, E. Küster-Schöck, A. D. Grossman, and P. Zuber. "Spx-dependent global transcriptional control is induced by thiol-specific oxidative stress in *Bacillus subtilis*". In: *Proc Natl Acad Sci U S A* 100.23 2003. Pp. 13603–8.

276. Price ND, Reed JL, and Palsson B/O. “Genome-scale models of microbial cells: evaluating the consequences of constraints”. In: *Nat Rev Microbiol* 2 2004. Pp. 886–897.
277. F. C. Neidhardt and R. Curtiss. *Escherichia coli and Salmonella : cellular and molecular biology*. ASM Press, Washington DC, 1996.
278. F. C. Neidhardt, J. L. Ingraham, and M. Schaechter. *Physiology of the Bacterial Cell: A Molecular Approach*. Sinauer Associates Inc, 1990.
279. K. J. Newberry, S. Nakano, P. Zuber, and R. G. Brennan. “Crystal structure of the *Bacillus subtilis* anti-alpha, global transcriptional regulator, Spx, in complex with the alpha C-terminal domain of RNA polymerase”. In: *Proc Natl Acad Sci U S A* 102.44 2005. Pp. 15839–44.
280. L. H. Nguyen, D. Barsky, J. P. Erzberger, and D. M. 3rd Wilson. “Mapping the protein-DNA interface and the metal-binding site of the major human apurinic/apyrimidinic endonuclease”. In: *J Mol Biol* 298.3 2000. Pp. 447–59.
281. A. W. Nicholson. “Function, mechanism and regulation of bacterial ribonucleases”. In: *FEMS Microbiol Rev* 23.3 1999. Pp. 371–90.
282. H. Nielsen, J. Engelbrecht, S. Brunak, and G. von Heijne. “Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites”. In: *Protein Eng* 10.1 1997. Pp. 1–6.
283. K. H. Nierhaus. “The assembly of prokaryotic ribosomes”. In: *Biochimie* 73.6 1991. Pp. 739–55.
284. M. Nöllmann et al. “Multiple modes of *Escherichia coli* DNA gyrase activity revealed by force and torque”. In: *Nat Struct Mol Biol* 14.4 2007. Pp. 264–71.
285. D. P. Norman, S. D. Bruner, and G. L. Verdine. “Coupling of substrate recognition and catalysis by a human base-excision DNA repair protein”. In: *J Am Chem Soc* 123.2 2001. Pp. 359–60.
286. Baliga NS et al. “Coordinate regulation of energy transduction modules in *Halobacterium* sp. analyzed by a global systems approach”. In: *Proc Natl Acad Sci USA* 99 2002. Pp. 14913–14918.
287. J. D. Orth, I. Thiele, and B. O. Palsson. “What is flux balance analysis?” In: *Nat Biotechnol* 28.3 2010. Pp. 245–8.

288. M. Osawa, D. E. Anderson, and H. P. Erickson. “Reconstitution of contractile FtsZ rings in liposomes”. In: *Science* 320.5877 2008. Pp. 792–4.
289. Qiu P et al. “Extracting a cellular hierarchy from high-dimensional cytometry data with SPADE”. In: *Nat Biotechnol* 29 2011. Pp. 886–91.
290. Shannon P et al. “Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks”. In: *Genome Res* 13 2003. Pp. 2498–2504.
291. R. Pain, ed. *Mechanisms of protein folding*. Vol. 32. Oxford University Press: USA, 2000.
292. M. J. Pallen, A. C. Lam, M. Antonio, and K. Dunbar. “An embarrassment of sortases - a richness of substrates?” In: *Trends Microbiol* 9.3 2001. Pp. 97–102.
293. S. J. Pamp, D. Frees, S. Engelmann, M. Hecker, and H. Ingmer. “Spx is a global effector impacting stress tolerance and biofilm formation in *Staphylococcus aureus*”. In: *J Bacteriol* 188.13 2006. Pp. 4861–70.
294. J. Park et al. “PcrA helicase dismantles RecA filaments by reeling in DNA in uniform steps”. In: *Cell* 142.4 2010. Pp. 544–55.
295. Karp PD, Paley S, and Romero P. “The Pathway Tools Software”. In: *Bioinformatics* 18 2002. S225–S232.
296. Karp PD et al. “Pathway Tools version 13.0: Integrated Software for Pathway/Genome Informatics and Systems Biology”. In: *Briefings in Bioinformatics* 11 2010. Pp. 40–79.
297. N. Peekhaus and T. Conway. “Positive and negative transcriptional regulation of the *Escherichia coli* gluconate regulon gene gntT by GntR and the cyclic AMP (cAMP)-cAMP receptor protein complex”. In: *J Bacteriol* 180.7 1998. Pp. 1777–85.
298. L. Peil. “Ribosome assembly factors in *Escherichia coli*”. PhD thesis. Tartu University, 2009.
299. P. Peluso, S. O. Shan, S. Nock, D. Herschlag, and P. Walter. “Role of SRP RNA in the GTPase cycles of Ffh and FtsY”. In: *Biochemistry* 40.50 2001. Pp. 15224–33.
300. H. Peng and K. J. Marians. “The interaction of *Escherichia coli* topoisomerase IV with DNA”. In: *J Biol Chem* 270.42 1995. Pp. 25286–90.
301. M. Pertea, K. Ayanbule, M. Smedinghoff, and S. L. Salzberg. “OperonDB: a comprehensive database of predicted operons in microbial genomes”. In: *Nucleic Acids Res* 37.Database issue 2009. Pp. D479–82.

302. F. Peske, M. V. Rodnina, and W. Wintermeyer. “Sequence of steps in ribosome recycling as defined by kinetic analysis”. In: *Mol Cell* 18.4 2005. Pp. 403–12.
303. B. J. Peter et al. “Genomic transcriptional response to loss of chromosomal supercoiling in *Escherichia coli*”. In: *Genome Biol* 5.11 2004. R87.
304. J. D. Peterson, L. A. Umayam, T. Dickinson, E. K. Hickey, and O. White. “The Comprehensive Microbial Resource”. In: *Nucleic Acids Res* 29.1 2001. Pp. 123–5.
305. S. Petry, A. Weixlbaumer, and V. Ramakrishnan. “The termination of translation”. In: *Curr Opin Struct Biol* 18.1 2008. Pp. 70–7.
306. P. Pham et al. “Two distinct modes of RecA action are required for DNA polymerase V-catalyzed translesion synthesis”. In: *Proc Natl Acad Sci U S A* 99.17 2002. Pp. 11061–6.
307. K. Phannachet, Y. Elias, and R. H. Huang. “Dissecting the roles of a strictly conserved tyrosine in substrate recognition and catalysis by pseudouridine 55 synthase”. In: *Biochemistry* 44.47 2005. Pp. 15488–94.
308. O. Q. Pich, R. Burgos, M. Ferrer-Navarro, E. Querol, and J. Piñol. “Mycoplasma genitalium mg200 and mg386 genes are involved in gliding motility but not in cytadherence”. In: *Mol Microbiol* 60.6 2006. Pp. 1509–19.
309. O. Q. Pich, R. Burgos, M. Ferrer-Navarro, E. Querol, and J. Piñol. “Role of Mycoplasma genitalium MG218 and MG317 cytoskeletal proteins in terminal organelle organization, gliding motility and cytadherence”. In: *Microbiology* 154.Pt 10 2008. Pp. 3188–98.
310. O. Q. Pich, R. Burgos, E. Querol, and J. Piñol. “P110 and P140 cytadherence-related proteins are negative effectors of terminal organelle duplication in Mycoplasma genitalium”. In: *PLoS One* 4.10 2009. e7452.
311. G. Piec, J. Mirkovitch, S. Palacio, P. F. Mühlradt, and R. Felix. “Effect of MALP-2, a lipopeptide from *Mycoplasma fermentans*, on bone resorption in vitro”. In: *Infect Immun* 67.12 1999. Pp. 6281–5.
312. S. J. Pirt. 1975.
313. A. Politi et al. “Mathematical modeling of nucleotide excision repair reveals efficiency of sequential assembly strategies”. In: *Mol Cell* 19.5 2005. Pp. 679–90.

314. J. D. Pollack, M. A. Myers, T. Dandekar, and R. Herrmann. "Suspected utility of enzymes with multiple activities in the small genome Mycoplasma species: the replacement of the missing "household" nucleoside diphosphate kinase gene and activity by glycolytic kinases". In: *OMICS* 6.3 2002. Pp. 247–58.
315. I. M. Porter, G. A. Khoudoli, and J. R. Swedlow. "Chromosome condensation: DNA compaction in real time". In: *Curr Biol* 14.14 2004. R554–6.
316. W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Integration of Ordinary Differential Equations*. Cambridge University Press, 2007.
317. T. Proft, H. Hilbert, H. Plagens, and R. Herrmann. "The P200 protein of Mycoplasma pneumoniae shows common features with the cytadherence-associated proteins HMW1 and HMW3". In: *Gene* 171.1 1996. Pp. 79–82.
318. Lum PY, Paquette J, Singh G, and Carlsson G. *Patient Stratification using Topological Data Analysis and Iris*. http://www.ayasdi.com/_downloads/Patient_Stratification_using_Topological_Data_Analysis.pdf. 2013.
319. Mahadevan R, Edwards JS, and Doyle FJ 3rd. "Dynamic flux balance analysis of diauxic growth in *Escherichia coli*". In: *Biophys J* 83 2002. Pp. 1331–1340.
320. Schuetz R, Kuepfer L, and Sauer U. "Systematic evaluation of objective functions for predicting intracellular fluxes in *Escherichia coli*". In: *Mol Sys Biol* 3 2007. P. 119.
321. A. Raine, N. Ivanova, J. E. Wikberg, and M. Ehrenberg. "Simultaneous binding of trigger factor and signal recognition particle to the *E. coli* ribosome". In: *Biochimie* 86.7 2004. Pp. 495–500.
322. U. Ramesh and C. F. Meares. "Footprint of the sigma protein". In: *Biochem Biophys Res Commun* 160.1 1989. Pp. 121–5.
323. S. Razin and R. Herrmann, eds. *Cell Division*. Springer, 2002.
324. S. Razin and E. Jacobs. "Mycoplasma adhesion". In: *J Gen Microbiol* 138.3 1992. Pp. 407–22.
325. S. Razin, D. Yogev, and Y. Naot. "Molecular biology and pathogenicity of mycoplasmas". In: *Microbiol Mol Biol Rev* 62.4 1998. Pp. 1094–156.
326. J. T. Reardon and A. Sancar. "Nucleotide excision repair". In: *Prog Nucleic Acid Res Mol Biol* 79 2005. Pp. 183–235.

327. J. A. Reems and C. S. McHenry. "Escherichia coli DNA polymerase III holoenzyme footprints three helical turns of its primer". In: *J Biol Chem* 269.52 1994. Pp. 33091–6.
328. P. Régnier and C. M. Arraiano. "Degradation of mRNA in bacteria: emergence of ubiquitous features". In: *Bioessays* 22.3 2000. Pp. 235–44.
329. J. T. Regula et al. "Defining the mycoplasma 'cytoskeleton': the protein composition of the Triton X-100 insoluble fraction of the bacterium *Mycoplasma pneumoniae* determined by 2-D gel electrophoresis and mass spectrometry". In: *Microbiology* 147.Pt 4 2001. Pp. 1045–57.
330. R. Remak. *Untersuchungen über die Entwicklung der Wirbelthiere*. G. Reimer, Berlin, 1855.
331. S. Rey et al. "PSORTdb: a protein subcellular localization database for bacteria". In: *Nucleic Acids Res* 33.Database issue 2005. Pp. D164–8.
332. *Rocks Open-Source Toolkit for Real and Virtual Clusters*. 2012.
333. U. Römling. "Great times for small molecules: c-di-AMP, a second messenger candidate in Bacteria and Archaea". In: *Sci Signal* 1.33 2008. pe39.
334. R. W. Rose, T. Brüser, J. C. Kissinger, and M. Pohlschröder. "Adaptation of protein secretion to extremely high-salt conditions by extensive use of the twin-arginine translocation pathway". In: *Mol Microbiol* 45.4 2002. Pp. 943–50.
335. J. Rouvière-Yaniv, M. Yaniv, and J. E. Germond. "E. coli DNA binding protein HU forms nucleosomelike structure with circular double-stranded DNA". In: *Cell* 17.2 1979. Pp. 265–74.
336. M. C. le Roux and A. A. Hoosen. "Mycoplasma genitalium: a brief review". In: *South Afr J Epidemiol Infect* 25.4 2010. Pp. 7–10.
337. Pomerantz RT and O'Donnell M. "Direct restart of a replication fork stalled by a head-on RNA polymerase". In: *Science* 327 2010. Pp. 590–592.
338. C. J. Rudolph, P. Dhillon, T. Moore, and R. G. Lloyd. "Avoiding and resolving conflicts between DNA replication and transcription". In: *DNA Repair (Amst)* 6.7 2007. Pp. 981–93.
339. Luo RY et al. "Dynamic analysis of optimality in myocardial energy metabolism under normal and ischemic conditions". In: *Mol Sys Biol* 2 2006. P. 2006.0031.
340. Brenner S. "Sequences and consequences". In: *Philos Trans R Soc Lond B Biol Sci* 365 2010. Pp. 207–212.

341. T. Samuelsson, Y. S. Guindy, F. Lustig, T. Borén, and U. Lagerkvist. “Apparent lack of discrimination in the reading of certain codons in *Mycoplasma mycoides*”. In: *Proc Natl Acad Sci U S A* 84.10 1987. Pp. 3166–70.
342. K. Sankaran and H. C. Wu. “Lipid modification of bacterial prolipoprotein. Transfer of diacylglycerol moiety from phosphatidylglycerol”. In: *J Biol Chem* 269.31 1994. Pp. 19701–6.
343. L. Schaefer et al. “Multiple GTPases participate in the assembly of the large ribosomal subunit in *Bacillus subtilis*”. In: *J Bacteriol* 188.23 2006. Pp. 8252–8.
344. H. L. Schmidt, W. Stöcklein, J. Danzer, P. Kirch, and B. Limbach. “Isolation and properties of an H₂O-forming NADH oxidase from *Streptococcus faecalis*”. In: *Eur J Biochem* 156.1 1986. Pp. 149–55.
345. D. Schomburg, I. Schomburg, and A. Chang, eds. *Class 2 Transferases I: EC 2.1.1*. Vol. 28. Springer Berlin, Heidelberg, 2006.
346. D. Schomburg, I. Schomburg, and A. Chang, eds. *tRNA Sulfurtransferase*. Vol. 39. Springer Berlin, Heidelberg, 2008.
347. Nara Institute of Science and Technology. *GenoBase*.
348. Thermo Scientific. *HyClone Fetal Bovine Serum, U.S. Origin, Certificate of Analysis*. 2011.
349. A. Scrima, I. R. Vetter, M. E. Armengod, and A. Wittinghofer. “The structure of the TrmE GTP-binding protein and its implications for tRNA modification”. In: *EMBO J* 24.1 2005. Pp. 23–33.
350. B. Sedgwick. “Repairing DNA-methylation damage”. In: *Nat Rev Mol Cell Biol* 5.2 2004. Pp. 148–57.
351. E. Seeberg and R. P. Fuchs. “Acetylaminofluorene bound to different guanines of the sequence -GGCGCC- is excised with different efficiencies by the UvrABC excision nuclease in a pattern not correlated to the potency of mutation induction”. In: *Proc Natl Acad Sci U S A* 87.1 1990. Pp. 191–4.
352. D. Segrè, D. Vitkup, and G. M. Church. “Analysis of optimality in natural and perturbed metabolic networks”. In: *Proc Natl Acad Sci U S A* 99.23 2002. Pp. 15112–7.

353. D. W. Selinger, R. M. Saxena, K. J. Cheung, G. M. Church, and C. Rosenow. "Global RNA half-life analysis in *Escherichia coli* reveals positional patterns of transcript degradation". In: *Genome Res* 13.2 2003. Pp. 216–23.
354. A. T. Selvan and K. Sankaran. "Localization and characterization of prolipoprotein diacylglyceryl transferase (Lgt) critical in bacterial lipoprotein biosynthesis". In: *Biochimie* 90.11-12 2008. Pp. 1647–55.
355. J. Seo and K. J. Lee. "Post-translational modifications and their biological functions: proteomic analysis and systematic approaches". In: *J Biochem Mol Biol* 37.1 2004. Pp. 35–44.
356. Altschul SF, Gish W, Miller W, Myers EW, and Lipman DJ. "Basic local alignment search tool". In: *J Mol Biol* 215 1990. Pp. 403–410.
357. T. Shimizu, Y. Kida, and K. Kuwano. "A triacylated lipoprotein from *Mycoplasma genitalium* activates NF-kappaB through Toll-like receptor 1 (TLR1) and TLR2". In: *Infect Immun* 76.8 2008. Pp. 3672–8.
358. T. Shimizu, Y. Kida, and K. Kuwano. "Triacylated lipoproteins derived from *Mycoplasma pneumoniae* activate nuclear factor-kappaB through toll-like receptors 1 and 2". In: *Immunology* 121.4 2007. Pp. 473–83.
359. S. Shuman. "DNA ligases: progress and prospects". In: *J Biol Chem* 284.26 2009. Pp. 17365–9.
360. O'Donoghue SI et al. "Visualizing biological data-now and in the future". In: *Nat Meth* 7 2010. S2–4.
361. V. S. Sidorenko and D. O. Zharkov. "Correlated cleavage of damaged DNA by bacterial and human 8-oxoguanine-DNA glycosylases". In: *Biochemistry* 47.34 2008. Pp. 8970–6.
362. N. Sierro, Y. Makita, M. de Hoon, and K. Nakai. "DBTBS: a database of transcriptional regulation in *Bacillus subtilis* containing upstream intergenic conservation information". In: *Nucleic Acids Res* 36.Database issue 2008. Pp. D93–6.
363. P. Simoneau, R. Wenzel, R. Herrmann, and P. C. Hu. "Nucleotide sequence of a tRNA cluster from *Mycoplasma pneumoniae*". In: *Nucleic Acids Res* 18.9 1990. P. 2814.
364. P. Simoneau et al. "Codon reading scheme in *Mycoplasma pneumoniae* revealed by the analysis of the complete set of tRNA genes". In: *Nucleic Acids Res* 21.21 1993. Pp. 4967–74.

365. K. H. Sippel et al. "Insights into Mycoplasma genitalium metabolism revealed by the structure of MG289, an extracytoplasmic thiamine binding lipoprotein". In: *Proteins* 79.2 2011. Pp. 528–36.
366. M. Sluijter et al. "The Mycoplasma genitalium MG352-encoded protein is a Holliday junction resolvase that has a non-functional orthologue in Mycoplasma pneumoniae". In: *Mol Microbiol* 77.5 2010. Pp. 1261–77.
367. Paley SM and Karp PD. "The Pathway Tools cellular overview diagram and Omics Viewer". In: *Nucleic Acids Res* 34 2006. Pp. 771–3778.
368. P. F. Smith. "Lipoglycans from mycoplasmas". In: *Crit Rev Microbiol* 11.2 1984. Pp. 157–86.
369. B. A. Sokhansanj, G. R. Rodrigue, J. P. Fitch, and D. M. 3rd Wilson. "A quantitative model of human DNA base excision repair. I. Mechanistic insights". In: *Nucleic Acids Res* 30.8 2002. Pp. 1817–25.
370. B. A. Sokhansanj and D. M. 3rd Wilson. "Oxidative DNA damage background estimated by a system model of base excision repair". In: *Free Radic Biol Med* 37.3 2004. Pp. 422–7.
371. Solabia. *Bacteriological Meat Extract - A1720*. 2011.
372. Solabia. *Beef Heart Infusion - A1502*. 2011.
373. Solabia. *Biotechnology Products*. Retrieved from <http://www.solabia.com/>. 2011.
374. Solabia. *Pancreatic Digest of Casein - A1403 / A1433*. 2011.
375. Solabia. *Pork Meat Peptone - A1728*. 2011.
376. Solabia. *Tryptone V - A1443*. 2011.
377. Solabia. *Yeast Extract - A1202*. 2011.
378. Perfetto SP, Chattopadhyay PK, and Roederer M. "Seventeen-colour flow cytometry: unravelling the immune system". In: *Nat Rev Immunol* 4 2004. Pp. 648–655.
379. C. Speck and W. Messer. "Mechanism of origin unwinding: sequential binding of DnaA to double- and single-stranded DNA". In: *EMBO J* 20.6 2001. Pp. 1469–76.
380. C. C. Staats, J. Boldo, L. Broetto, M. Vainstein, and A. Schrank. "Comparative genome analysis of proteases, oligopeptide uptake and secretion systems in Mycoplasma spp". In: *Genetics and Molecular Biology* 30.1 2007. Pp. 225–229.

381. P. E. Stephens, M. G. Darlison, H. M. Lewis, and J. R. Guest. "The pyruvate dehydrogenase complex of Escherichia coli K12. Nucleotide sequence encoding the dihydrolipoamide acetyltransferase component". In: *Eur J Biochem* 133.3 1983. Pp. 481–9.
382. A. M. Stevens, K. M. Dolan, and E. P. Greenberg. "Synergistic binding of the Vibrio fischeri LuxR transcriptional activator domain and RNA polymerase to the lux promoter region". In: *Proc Natl Acad Sci U S A* 91.26 1994. Pp. 12619–23.
383. A. M. Stevens, N. Fujita, A. Ishihama, and E. P. Greenberg. "Involvement of the RNA polymerase alpha-subunit C-terminal domain in LuxR-dependent activation of the Vibrio fischeri luminescence genes". In: *J Bacteriol* 181.15 1999. Pp. 4704–7.
384. M. D. Stone et al. "Chirality sensing by Escherichia coli topoisomerase IV and the mechanism of type II topoisomerases". In: *Proc Natl Acad Sci U S A* 100.15 2003. Pp. 8654–9.
385. T. R. Strick, T. Kawaguchi, and T. Hirano. "Real-time detection of single-molecule DNA compaction by condensin I". In: *Curr Biol* 14.10 2004. Pp. 874–80.
386. H. C. Su, C. A. 3rd Hutchison, and M. C. Giddings. "Mapping phosphoproteins in Mycoplasma genitalium and Mycoplasma pneumoniae". In: *BMC Microbiol* 7 2007. P. 63.
387. S. Sundararaj et al. "The CyberCell Database (CCDB): a comprehensive, self-updating, relational database to coordinate and facilitate in silico modeling of Escherichia coli". In: *Nucleic Acids Res* 32.Database issue 2004. Pp. D293–5.
388. I. V. Surovtsev, J. J. Morgan, and P. A. Lindahl. "Kinetic modeling of the assembly, dynamic steady state, and contraction of the FtsZ ring in prokaryotic cytokinesis". In: *PLoS Comput Biol* 4.7 2008. e1000102.
389. P. F. Suthers et al. "A genome-scale metabolic reconstruction of Mycoplasma genitalium, iPS189". In: *PLoS Comput Biol* 5.2 2009. e1000285.
390. D. Szafron et al. "Proteome Analyst: custom predictions with explanations in a web-based tool for high-throughput proteome annotations". In: *Nucleic Acids Res* 32.Web Server issue 2004. W365–71.
391. Shlomi T, Berkman O, and Ruppin E. "Regulatory on/off minimization of metabolic flux changes after genetic perturbations". In: *Proc Natl Acad Sci USA* 102 2005. Pp. 7695–7700.
392. Shlomi T, Eisenberg Y, Sharan R, and Ruppin E. "A genome-scale computational study of the interplay between transcriptional regulation and metabolism". In: *Mol Sys Biol* 3 2007. P. 101.

393. Vora T, Hottes AK, and Tavazoie S. "Protein occupancy landscape of a bacterial genome". In: *Mol Cell* 35 2009. Pp. 247–253.
394. R. Tanaka, Y. Andachi, and A. Muto. "Evolution of tRNAs and tRNA genes in *Acholeplasma laidlawii*". In: *Nucleic Acids Res* 19.24 1991. Pp. 6787–92.
395. Y. Taniguchi et al. "Quantifying *E. coli* proteome and transcriptome with single-molecule sensitivity in single cells". In: *Science* 329.5991 2010. Pp. 533–8.
396. T. A. Tatusova, I. Karsch-Mizrachi, and J. A. Ostell. "Complete genomes in WWW Entrez: data representation and analysis". In: *Bioinformatics* 15.7-8 1999. Pp. 536–43.
397. P. D. Taylor, C. P. Toseland, T. K. Attwood, and D. R. Flower. "LIPPRED: A web server for accurate prediction of lipoprotein signal sequences and cleavage sites". In: *Bioinformation* 1.5 2006. Pp. 176–9.
398. D. Taylor-Robinson and C. Bébéar. "Antibiotic susceptibilities of mycoplasmas and treatment of mycoplasmal infections". In: *J Antimicrob Chemother* 40.5 1997. Pp. 622–30.
399. G. W. Teebor, R. J. Boorstein, and J. Cadet. "The repairability of oxidative free radical mediated damage to DNA: a review". In: *Int J Radiat Biol* 54.2 1988. Pp. 131–50.
400. P. Theiss, A. Karpas, and K. S. Wise. "Antigenic topology of the P29 surface lipoprotein of *Mycoplasma fermentans*: differential display of epitopes results in high-frequency phase variation". In: *Infect Immun* 64.5 1996. Pp. 1800–9.
401. I. Thiele, N. Jamshidi, R. M. Fleming, and B. O. Palsson. "Genome-scale reconstruction of *Escherichia coli*'s transcriptional and translational machinery: acknowledge base, its mathematical formulation, and its functional characterization". In: *PLoS Comput Biol* 5.3 2009. e1000312.
402. I. Thiele and B. O. Palsson. "A protocol for generating a high-quality genome-scale metabolic reconstruction". In: *Nat Protoc* 5.1 2010. Pp. 93–121.
403. J. Thompson and E. Cundliffe. "Novel patterns of ultraviolet mutagenesis and Weigle reactivation in *Staphylococcus aureus* and phage phi 11". In: *J Gen Microbiol* 124 1981. Pp. 291–7.
404. J. W. Tobias, T. E. Shrader, G. Rocap, and A. Varshavsky. "The N-end rule in bacteria". In: *Science* 254.5036 1991. Pp. 1374–7.

405. K. Tomii and M. Kanehisa. “A comparative analysis of ABC transporters in complete microbial genomes”. In: *Genome Res* 8.10 1998. Pp. 1048–59.
406. M. Tomita et al. “E-CELL: software environment for whole-cell simulation”. In: *Bioinformatics* 15.1 1999. Pp. 72–84.
407. D. Tomkiewicz, N. Nouwen, R. van Leeuwen, S. Tans, and A. J. Driessens. “SecA supports a constant rate of preprotein translocation”. In: *J Biol Chem* 281.23 2006. Pp. 15709–13.
408. E. J. Tomko, C. J. Fischer, A. Niedziela-Majka, and T. M. Lohman. “A nonuniform stepping mechanism for *E. coli* UvrD monomer translocation along single-stranded DNA”. In: *Mol Cell* 26.3 2007. Pp. 335–47.
409. S. Tornaletti. “DNA repair in mammalian cells: Transcription-coupled DNA repair: directing your effort where it’s most needed”. In: *Cell Mol Life Sci* 66.6 2009. Pp. 1010–20.
410. S. Tornaletti. “Transcription arrest at DNA damage sites”. In: *Mutat Res* 577.1-2 2005. Pp. 131–45.
411. S. Tornaletti and P. C. Hanawalt. “Effect of DNA lesions on transcription elongation”. In: *Biochimie* 81.1-2 1999. Pp. 139–46.
412. *Torque Resource Manager*. 2012.
413. S. Tottey et al. “Protein-folding location can regulate manganese-binding versus copper- or zinc-binding”. In: *Nature* 455.7216 2008. Pp. 1138–42.
414. J. G. Tully, D. Taylor-Robinson, D. L. Rose, R. M. Cole, and J. M. Bove. “Mycoplasma genitalium, a New Species from the Human Urogenital Tract”. In: *Int J Syst Bacteriol* 33 1983. Pp. 387–396.
415. J. G. Tully, D. Taylor-Robinson, D. L. Rose, R. M. Cole, and J. M. Bove. “Mycoplasma genitalium, a New Species from the Human Urogenital Tract”. In: *Int J Syst Bacteriol* 33 1983. Pp. 387–396.
416. W. C. Uicker, L. Schaefer, and R. A. Britton. “The essential GTPase RbgA (YlqF) is required for 50S ribosome assembly in *Bacillus subtilis*”. In: *Mol Microbiol* 59.2 2006. Pp. 528–40.
417. C. Ullspurger and N. R. Cozzarelli. “Contrasting enzymatic activities of topoisomerase IV and DNA gyrase from *Escherichia coli*”. In: *J Biol Chem* 271.49 1996. Pp. 31549–55.
418. M. L. Urbanowski, C. P. Lostroh, and E. P. Greenberg. “Reversible acyl-homoserine lactone binding to purified *Vibrio fischeri* LuxR protein”. In: *J Bacteriol* 186.3 2004. Pp. 631–7.

419. B. Van Houten, H. Gamper, A. Sancar, and J. E. Hearst. “DNase I footprint of ABC excinuclease”. In: *J Biol Chem* 262.27 1987. Pp. 13180–7.
420. R. Visse, M. de Ruijter, G. F. Moolenaar, and P. van de Putte. “Analysis of UvrABC endonuclease reaction intermediates on cisplatin-damaged DNA using mobility shift gel electrophoresis”. In: *J Biol Chem* 267.10 1992. Pp. 6736–42.
421. U. Vogel and K. F. Jensen. “Effects of the antiterminator BoxA on transcription elongation kinetics and ppGpp inhibition of transcription elongation in Escherichia coli”. In: *J Biol Chem* 270.31 1995. Pp. 18335–40.
422. R. de Vries. “DNA condensation in bacteria: Interplay between macromolecular crowding and nucleoid proteins”. In: *Biochimie* 92.12 2010. Pp. 1715–21.
423. C. T. Walsh, S. Garneau-Tsodikova, and G. J. Jr Gatto. “Protein posttranslational modifications: the chemistry of proteome diversifications”. In: *Angew Chem Int Ed Engl* 44.45 2005. Pp. 7342–72.
424. J. C. Wang. “DNA topoisomerases”. In: *Annu Rev Biochem* 65 1996. Pp. 635–92.
425. J. C. Wang. “Helical repeat of DNA in solution”. In: *Proc Natl Acad Sci U S A* 76.1 1979. Pp. 200–3.
426. Y. Wang et al. “PubChem: a public information system for analyzing bioactivities of small molecules”. In: *Nucleic Acids Res* 37.Web Server issue 2009. W623–33.
427. T. Washio, J. Sasayama, and M. Tomita. “Analysis of complete genomes suggests that many prokaryotes do not rely on hairpin formation in transcription termination”. In: *Nucleic Acids Res* 26.23 1998. Pp. 5456–63.
428. T. Washio, M. Wada, and M. Tomita. *Sequence Analysis of Short Tandem Repeats in the Genomes of H. influenzae and M. genitalium*. 1996.
429. R. Wehelie. “Mycoplasma pyrimidine deoxynucleotide biosynthesis”. PhD thesis. Acta Universitatis agriculturae Sueciae, 2006.
430. J. 3rd Weiner, R. Herrmann, and G. F. Browning. “Transcription in Mycoplasma pneumoniae”. In: *Nucleic Acids Res* 28.22 2000. Pp. 4488–96.
431. J. 3rd Weiner, C. U. Zimmerman, H. W. Göhlmann, and R. Herrmann. “Transcription profiles of the bacterium Mycoplasma pneumoniae grown at different temperatures”. In: *Nucleic Acids Res* 31.21 2003. Pp. 6306–20.

432. M. R. Wilkins et al. "Protein identification and analysis tools in the ExPASy server". In: *Methods Mol Biol* 112 1999. Pp. 531–52.
433. A. C. Wilson and M. Tan. "Stress response gene regulation in Chlamydia is dependent on HrcA-CIRCE interactions". In: *J Bacteriol* 186.11 2004. Pp. 3384–91.
434. G. G. Wilson and N. E. Murray. "Restriction and modification systems". In: *Annu Rev Genet* 25 1991. Pp. 585–627.
435. G. Witte, S. Hartung, K. Büttner, and K. P. Hopfner. "Structural biochemistry of a bacterial checkpoint protein reveals diadenylate cyclase activity regulated by DNA recombination intermediates". In: *Mol Cell* 30.2 2008. Pp. 167–78.
436. U. Wittig et al. "In proceedings of the 3rd International workshop on Data Integration in the Life Sciences 2006". In: *Lecture Notes in Bioinformatics* 4075 2006. Pp. 94–103.
437. R. L. Wurdeman, K. M. Church, and B. Gold. "DNA methylation by N-methyl-N-nitrosourea, N-methyl-N'-nitro-N-nitrosoguanidine, N-nitroso(1-acetoxyethyl)methylamine, and diazomethane. The mechanism for the formation of N7-methylguanine in sequence-characterized 5'-[32P]-end-labeled DNA". In: *J Am Chem Soc* 111.16 1989. Pp. 6408–6412.
438. X. S. Xie, P. J. Choi, G. W. Li, N. K. Lee, and G. Lia. "Single-molecule approach to molecular biology in living bacterial cells". In: *Annu Rev Biophys* 37 2008. Pp. 417–44.
439. Y. Xu, N. D. Grindley, and C. M. Joyce. "Coordination between the polymerase and 5'-nuclease components of DNA polymerase I of Escherichia coli". In: *J Biol Chem* 275.27 2000. Pp. 20949–55.
440. Harada Y et al. "Single-molecule imaging of RNA polymerase-DNA interactions in real time". In: *Biophys J* 76 1999. Pp. 709–715.
441. Setty Y, Mayo AE, Surette MG, and Alon U. "Detailed map of a cis-regulatory input function". In: *Proc Natl Acad Sci USA* 100 2003. Pp. 7702–7707.
442. K. Yoda and T. Okazaki. "Specificity of recognition sequence for Escherichia coli primase". In: *Mol Gen Genet* 227.1 1991. Pp. 1–8.
443. R. Young and H. Bremer. "Polypeptide-chain-elongation rate in Escherichia coli B/r as a function of growth rate". In: *Biochem J* 160.2 1976. Pp. 185–94.
444. J. Yu, J. Xiao, X. Ren, K. Lao, and X. S. Xie. "Probing gene expression in live cells, one protein molecule at a time". In: *Science* 311.5767 2006. Pp. 1600–3.

445. E. Yus et al. "Impact of genome reduction on bacterial metabolism and its regulation". In: *Science* 326.5957 2009. Pp. 1263–8.
446. E. L. Zechiedrich et al. "Roles of topoisomerases in maintaining steady-state DNA supercoiling in *Escherichia coli*". In: *J Biol Chem* 275.11 2000. Pp. 8103–13.
447. H. Zhao, U. Dreses-Werringloer, P. Davies, and P. Marambaud. "Amyloid-beta peptide degradation in cell cultures by mycoplasma contaminants". In: *BMC Res Notes* 1 2008. P. 38.
448. Y. Zou, T. M. Liu, N. E. Geacintov, and B. Van Houten. "Interaction of the UvrABC nuclease system with a DNA duplex containing a single stereoisomer of dG-(+)- or dG-(-)-anti-BPDE". In: *Biochemistry* 34.41 1995. Pp. 13582–93.
449. U. Zuber and W. Schumann. "CIRCE, a novel heat shock element involved in regulation of heat shock operon dnaK of *Bacillus subtilis*". In: *J Bacteriol* 176.5 1994. Pp. 1359–63.