2

Inferring Gene Regulatory Networks from Expression Data

Lars Kaderali[1] and Nicole Radde[2]

[1] University of Heidelberg, Viroquant Research Group Modeling, Bioquant BQ26, Im Neuenheimer Feld 267, 69120 Heidelberg, Germany, lars.kaderali@bioquant.uni-heidelberg.de

[2] University of Leipzig, Institute for Medical Informatics, Statistics and Epidemiology, Härtelstr. 16-18, 04107 Leipzig, Germany, nicole.radde@imise.uni-leipzig.de

Summary. Gene regulatory networks describe how cells control the expression of genes, which, together with some additional regulation further downstream, determines the production of proteins essential for cellular function. The level of expression of each gene in the genome is modified by controlling whether and how vigorously it is transcribed to RNA, and subsequently translated to protein. RNA and protein expression will influence expression rates of other genes, thus giving rise to a complicated network structure.

An analysis of regulatory processes within the cell will significantly further our understanding of cellular dynamics. It will shed light on normal and abnormal, diseased cellular events, and may provide information on pathways in dire diseases such as cancer. These pathways can provide information on how the disease develops, and what processes are involved in progression. Ultimately, we can hope that this will provide us with new therapeutic approaches and targets for drug design.

It is thus no surprise that many efforts have been undertaken to reconstruct gene regulatory networks from gene expression measurements. In this chapter, we will provide an introductory overview over the field. In particular, we will present several different approaches to gene regulatory network inference, discuss their strengths and weaknesses, and provide guidelines on which models are appropriate under what circumstances. In addition, we sketch future developments and open problems.

2.1 Introduction

Biology has undergone a seminal shift in the last decade, with a transition from focusing on simple, small components of cells, such as DNA, RNA and proteins, to the analysis of relationships and interactions between various parts of a biological system. The traditional approach to much of molecular biology breaks up a system into its various parts, analyzes each part in turn, and hopes to reassemble the parts back into a whole system. In contrast, the systems biology approach aims at understanding and modeling the entire system quantitatively, proposing that the system is more than the sum of its parts and can only be understood as a whole.

L. Kaderali and N. Radde: *Inferring Gene Regulatory Networks from Expression Data*, Studies in Computational Intelligence (SCI) **94**, 33–74 (2008)
www.springerlink.com

Gene regulatory networks control a cell at the genomic level, they orchestrate which genes and how vigorously these genes are transcribed to RNA, which in turn functions as a template for protein synthesis. Genes and proteins do not act independently. Instead, they interact with each other and form complicated regulatory networks. Proteins which function as transcription factors can positively or negatively influence the expression of another gene, and thus the production of other proteins. Some proteins act independently, others only become active in a complex. Gene regulatory networks describe these regulatory processes, and thus the molecular reaction of a cell to various stimuli. High throughput experimental techniques to measure RNA and protein concentrations enable new approaches to the analysis of such networks. The analysis of these data requires sophisticated techniques particularly tailored to the task. New statistical, qualitative and quantitative methods are being developed for this purpose.

At the modeling side, several levels of detail have traditionally been used to describe gene regulation. Starting with very simple models which allow for qualitative statements only, in recent years there is a tendency to describe the dynamic response of a system in more detail. Also, besides the analysis of given network models, the inference of parameters of a gene regulatory network from experimental data has become one of the big challenges in computational biology. As the number of parameters usually far exceeds the number of measurements available for this purpose, leading to under-determined problems, modelers have begun to use heterogeneous data sources for network inference and to include biological knowledge into the parameter estimation.

In the following, we will give an overview over different models and describe the challenges and current developments, with a focus on mathematical and computational techniques. In addition, we will present a novel method particularly suitable for the typical setting where one has only a low number of data points to estimate model parameters, but when still quantitative modeling is desired. We will show how inference from data can be carried out using the models discussed, and we will present algorithms for the computations involved.

Modeling of gene regulatory networks is a quickly evolving field, with new developments and algorithms being published almost daily. We can thus only provide an introduction to the subject matter with a rough overview, and in no way cover the field exhaustively. We will provide links to further literature where appropriate throughout the chapter, providing the reader with references for additional and more detailed information.

Before going into detail with the mathematical modeling of regulatory networks, we will briefly review the biological background in the following section. For more details see for example Alberts *et al*. [5], Cooper [28], Berg and Singer [13], or Collado-Vides and Hofestädt [27].

2.1.1 Biological Background

To understand the role regulatory networks play, we will start with the main players in a cell, the *proteins*. They consist of long folded chains of amino acids and

attend various tasks essential for survival of the cell. For example, they function as transporters, induce chemical reactions as enzymes, take part in metabolic pathways, recognize and transmit external signals, or act as ion channels in the cell membrane [5]. Proteins are permanently produced, this process is called *gene expression*. It consists of two stages, *transcription* and *translation*, and is highly regulated at different levels.

The information which proteins a cell can generally produce is encoded in its genome, the entirety of *genes* located on the DNA. During transcription, information from a gene is transcribed into an intermediate product called *messenger RNA*, or shortly *mRNA*. It serves as a template to produce a protein in the second step, the translation. The velocity and rate of this process is highly regulated and can vary in a wide range, making the organism flexible to adapt to external influences such as nutrition supply and to changes in environmental conditions such as temperature or salinity. It also enables the cell to respond to various stimuli and to maintain basic metabolic processes necessary for survival [27].

Regulation happens at different levels in the cell. We start with probably the most important mechanism, the regulation of transcription initiation. This is the main regulatory mechanism in prokaryotes. In eukaryotic cells, regulation is complicated by other effects such as alternative splicing or transport processes, we will neglect this here for simplicity. In transcription, an enzyme called *RNA-polymerase* (RNAP) is needed to catalyze the production of mRNA from an existing DNA template. This is initiated by binding of RNAP to the *promoter*, a regulatory region in front of the gene's coding region. Promoters contain specific binding sites for *transcription factors*, that is, for proteins regulating gene expression. Binding of RNAP and thus transcription initiation are facilitated by these transcription factors. *Operators* are DNA regions with binding sites for *repressors*, transcription factors which inhibit binding of the polymerase. A repressor-operator complex can influence the expression rates of multiple genes simultaneously. Some genes which encode for proteins involved in the same regulatory process are organized in *operons*, they are located side by side and are regulated by one single promoter. Their expression patterns are thus highly correlated. Transcription factors can also affect the process of RNA production by inducing conformational changes of the DNA, which can either activate or inhibit the polymerase [5].

Transcription factors do not always act independently, they can influence each other. When this influence is positive, one says that the transcription factors cooperatively enhance each other, their collective influence exceeds the sum of single influences. For example, some transcription factors are inactive until they form an active complex with other proteins. A transcription factor bound to DNA can facilitate the binding of another transcription factor by electrostatic attraction. Transcription factors can also inhibit each other. This is the case, for example, when several transcription factors compete for the same binding site, or when an occupied binding site prevents binding at another binding site, because the sequences of both sites overlap or because two transcription factors repel each other.

Regulation also happens after the gene is transcribed to mRNA. This is called *post-transcriptional regulation*. An example is the binding of a protein to mRNA, thus changing the secondary structure of the molecule, and hence stabilizing it or marking it for degradation. Analogously, regulation of protein concentration after translation is called *post-translational modification*. Mostly, a chemical group is appended to the protein, which induces a conformational change and activates or inactivates the protein. Many transcription factors taking part in signal transduction pathways have to be chemically modified to become active. These chemical modifications happen at a much faster time scale than the time scale for gene expression, which has consequences for quantitative models.

In addition to the production of RNA and protein, chemical degradation also affects concentrations of these molecules. RNA is quite unstable, and proteins are also degraded after some time. This is usually described as a first order decay process, thus degradation is assumed to be proportional to the component's concentration. Degradation rates are sometimes measured, and may then be included in models of gene regulation.

Figure 2.1 shows an example for regulation of gene expression at different levels. The four genes X, Y, Z_1 and Z_2 encode proteins which function as transcription factors. Protein X and the chemically modified protein Z_2 compete for the same binding site within an operator O. The repressor-operator complex inhibits transcription of the genes X and Y. Proteins Y and Z_1 form a complex that acts as a transcription factor for the operon Z containing the genes Z_1 and Z_2.

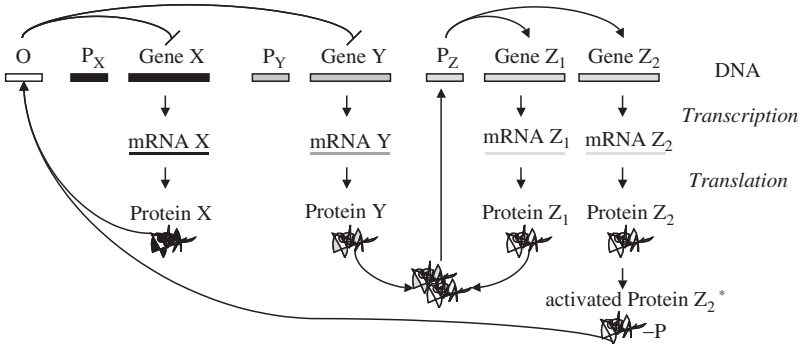

Fig. 2.1. Sample regulatory network consisting of four genes X, Y, Z_1 and Z_2. Regulation of gene expression happens at different levels: Protein X binds to an operator O and has thus a negative influence on the transcription rates of genes X and Y. Protein X and the chemically modified protein Z_2 compete for the same binding site. The proteins Y and Z_1 form an active complex, this complex acts as a transcription factor promoting expression of the operon Z, which in turn contains the genes Z_1 and Z_2

2.1.2 Modeling Gene Regulatory Networks

How can gene regulatory processes be represented in a computer? Our aim is twofold — inference of regulatory networks from data on the one hand, but also the simulation of a network's behavior on the other hand.

Recent advances in high-throughput biological techniques provide the basis for large scale analysis, which gives new insight into activities of cellular components under various biochemical and physiological conditions. DNA chips make the simultaneous measurement of concentrations of thousands of different RNA molecules possible, fermentation experiments yield data series of hundreds of metabolites, and large-scale measurements of protein concentrations are gradually becoming feasible. Moreover, the amount of protein-protein interaction and transcription factor binding site data is rapidly growing.

In computer models, gene regulatory networks are usually represented as directed graphs, with nodes corresponding to genes, and edges indicating interactions between the genes. In this chapter, we will discuss four different classes of models. In each section, we introduce a specific model or model class, and treat the inference problem. Subsequently, advantages and limitations of the models as well as possible extensions are discussed.

Boolean networks, described in Section 2.2, are probably the simplest models conceivable for regulatory networks. They assume that each gene is in one of two states, either *active* or *inactive*. Interactions between genes are modeled through Boolean logic functions, and updates are carried out simultaneously for all genes in discrete time steps. The updates are deterministic, and Boolean networks provide only a qualitative description of a system.

Relevance networks are described in Section 2.3. These approaches are based on pairwise distances (or similarities) between gene expression measurements, and try to reconstruct the networks using a threshold on the distance between genes.

Bayesian networks, discussed in Section 2.4, are probabilistic models. They model the conditional independence structure between genes in the network. Edges in a network correspond to probabilistic dependence relations between nodes, described by conditional probability distributions. Distributions used can be discrete or continuous, and Bayesian networks can be used to compute likely successor states for a given system in a known state.

Finally, *differential equation models*, described in Sections 2.5 to 2.7, provide a quantitative description of gene regulatory networks. Models used here range from simple linear differential equation models to complicated systems of nonlinear partial differential equations and stochastic kinetic approaches. In Section 2.5, we describe ordinary differential equation models. In Section 2.6, we present a novel method combining Bayesian networks and differential equations, and show first results on data from the yeast cell cycle network. Differential equation models going beyond ordinary differential equations are described in Section 2.7.

Finally, the last Section 2.8 gives a summary and an outlook, and provides a comparison between the model classes introduced.

2.2 Boolean Networks

Boolean networks offer a binary, discrete-time description of a system. They can be seen as a generalization of Boolean cellular automata [102], and have been introduced as models of genetic regulatory networks by Kauffman [52] in 1969. Let us start by stating a formal definition of a Boolean network:

Definition 1 (Boolean Network). *A* Boolean network *is defined as a tuple* $G = (X,B)$*, where* $X = (x_1,x_2,...,x_n) \in \{0,1\}^n$ *is a vector of Boolean variables, and* B *is a set of Boolean functions* $B = \{f_1,f_2,...,f_n\}$*,* $f_i : \{0,1\}^n \mapsto \{0,1\}$.

In gene expression networks, the x_i correspond to the genes and the f_i describe the interactions between them. In Boolean network models, one assumes that each gene can be modeled as being in one of two states, *on* (expressed, 1) or *off* (not expressed, 0). The functions B are used to update the nodes at discrete time-steps, all nodes X are updated synchronously using the Boolean functions B, that is, $x_i(t+1) = f_i(x_1(t),...,x_n(t))$. We call a snapshot of the values of the nodes $x(t) = (x_1(t),x_2(t),...,x_n(t))$ at time t the *expression pattern* or *state* of the network at the respective time point.

A Boolean network can be graphically represented in several ways, emphasizing different aspects of the network. An example is shown in Figure 2.2 for a small sample network consisting of three nodes A, B and C. The graph representation in Figure 2.2A shows how the nodes influence each other. Pointed arrows indicate an activation, see for example the positive regulation of node A by node B with the corresponding Boolean logic rule $A' = B$. In this example, the next value of node A, denoted A', will be equal to the current value of node B. Flat arrows indicate an inhibition, see for example the rule $B' = \neg A$. Here, the next value of node B will be the negation of node A, that is, $B' = 1$ if $A = 0$ and $B' = 0$ if $A = 1$. The value C' is computed from the current values of A and B together using the logical "OR" operation, hence $C' = 1$ if $A = 1$ or $B = 1$, and $C' = 0$ otherwise. The corresponding logical Boolean rules are given in Figure 2.2B. Figure 2.2C shows the *state transition table* of the network, it is a tabular representation of all possible "input" states of the network and, for each input, the resulting "output" or subsequent state. Figure 2.2D shows this table in a graph representation, visualizing the networks *state space* and its dynamics by connecting each input state with its corresponding output state. In this latter graph, it can be seen that the particular network in this example converges to a cycle of size four from any initial state.

2.2.1 Inferring a Boolean Network from Data

We will now discuss the problem of inferring a Boolean network from time series data. To formalize this, we define the *Consistency Problem:*

Definition 2 (Consistency Problem). *Let (I,O) be a pair of observed expression patterns of an unknown network* $G = (X,B)$*, such that* $O = B(I)$*, that is,* O *is the expression pattern of the network* G *after one time step when starting at state* I*.*

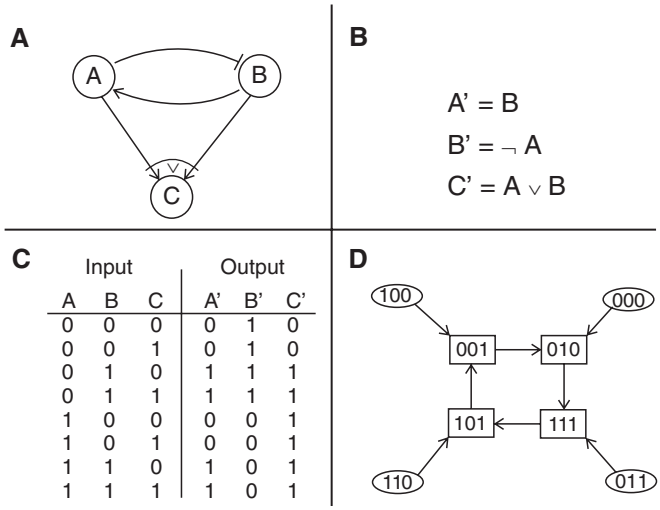

Fig. 2.2. Different representations of a sample Boolean network consisting of three nodes. (A) Graph representation, (B) logical Boolean rules, (C) state transition table and (D) state transition graph. In (A), pointed arrows indicate an activation, for example, gene A will be activated if gene B is active. Flat arrows indicate an inhibition, for example, gene B will be deactivated if gene A is active. Gene C is activated if either gene A or gene B is active, as denoted by the "or" symbol "$\vee$" in the figure. In Figure 2.2B, the same relationship is expressed in boolean logical rules. Figure 2.2C shows a tabular representation of all possible input states and the resulting next states of the network. Figure 2.2D visualizes the state space in a graphical form, showing how the eight possible states of the network are interconnected. For example, if the network is in state $(A = 1, B = 0, C = 0)$, then the next state of the network will be $(A = 0, B = 0, C = 1)$

Then, a network $G' = (X', B')$ *is* consistent *with* (I, O)*, if* $O = B'(I)$*.* G' *is consistent with a set of expression pairs* $D = \{(I_j, O_j)\}_{j=1,\ldots,m}$*, if it is consistent with each pair* (I_j, O_j) *in D. The* Consistency Problem *is the problem to decide, whether a Boolean network consistent with given data D exists, and output one if it exists [2].*

The *Identification Problem* for Boolean networks in addition asks whether the network is unique:

Definition 3 (Identification Problem). *Given the number of genes n and a set of m input-output pairs D, the* Identification Problem *is the problem to decide whether a* unique *Boolean network consistent with D exists, and output it if it does [2].*

The number of possible networks with a given number of nodes n is huge, hence exhaustive search is usually prohibitive. For a network of n nodes, for each node, there will be 2^{2^n} possible functions of n inputs. Even if we restrict the Boolean functions to functions with at most $k < n$ inputs, there will be 2^{2^k} possible functions, and each node has $n!/(n-k)!$ possible ordered combinations of k different inputs.

The number of possible networks for given n and k will thus be

$$\left(2^{2^k}\frac{n!}{(n-k)!}\right)^n, \tag{2.1}$$

which grows exponentially with the number n of nodes in the network. However, if the indegree k of the network is fixed, the following can be shown:

Theorem 1 (Akutsu, 1999). *The consistency problem and the identification problem can be solved in polynomial time for Boolean networks with maximum indegrees bounded by a constant k.*

If k is close to n, the consistency and identification problems are NP-hard [3].

Also the number of data points required to estimate the Boolean functions from data grows exponentially with the network size. Surprisingly, for networks of fixed indegree k, $O(\log n)$ input/output patterns are sufficient on average for the network identification, with constant around $k2^k$ in front of the $\log n$ [2]. This is why much effort has been spent on devising learning algorithms for Boolean networks with fixed maximum indegree. Several algorithms have been proposed for network inference, for example [2, 3, 54]. In the following, we will sketch the REVEAL algorithm by Liang, Fuhrmann and Somogyi [58], which is based on information theoretic principles.

The REVerse Engineering ALgorithm REVEAL

The strategy employed in the REVEAL algorithm is to infer regulatory interactions between nodes from measures of *mutual information* in state transition tables. The observed data D is considered a random variable, and information theoretic properties are then used to derive the network topology.

Given a random variable X with k possible, discrete outcomes $x_1, \ldots, x_k$, the *Shannon entropy* H of X is defined in terms of the probabilities $p(x_i)$ of the possible outcomes as

$$H(X) = -\sum_{i=1}^{k} p(x_i)\log p(x_i), \tag{2.2}$$

where the sum is over the different outcomes x_i with associated probabilities $p(x_i)$ [80]. The entropy is a measure of the *uncertainty* associated with a random variable. In a system with two binary random variables X and Y, the *individual* and *combined entropies* are defined as

$$H(X) = -\sum_{x\in\{0,1\}} p(x)\log p(x) \tag{2.3}$$

$$H(Y) = -\sum_{y\in\{0,1\}} p(y)\log p(y) \tag{2.4}$$

$$H(X,Y) = -\sum_{\substack{(x,y)\in\\ \{0,1\}\times\{0,1\}}} p(x,y)\log p(x,y), \tag{2.5}$$

where $p(x)$, $p(y)$ and $p(x,y)$ are the individual and combined probability distributions of the random variables X and Y, respectively. Note that, for sets $a = \{X_1, X_2, \ldots, X_n\}$ of random variables X_i, we will use the notation $H(a)$ to denote the joint entropy $H(X_1, X_2, \ldots, X_n)$, derived by naturally extending equation (2.5) for more than two variables. Similarly, for two sets a and b of random variables, $H(a,b) = H(a \cup b)$.

The *conditional entropy* $H(X|Y)$ is a measure of the remaining uncertainty associated with a random variable X, given that the value of a second random variable Y is known. The conditional entropies $H(X|Y)$ and $H(Y|X)$ are related to the individual and combined entropies through

$$H(X,Y) = H(Y|X) + H(X) = H(X|Y) + H(Y), \tag{2.6}$$

or, in words, the combined entropy of X and Y is the sum of the individual entropy of a single variable and the information contained in the second variable that is not shared with the first. The *mutual information* is then defined as

$$M(X,Y) = H(X) - H(X|Y) = H(Y) - H(Y|X), \tag{2.7}$$

it is a measure of the information about one variable, that is shared by the second variable. Mutual information measures, how much knowing one of the variables X and Y reduces our uncertainty about the other.

REVEAL extracts relationships between genes from mutual information in gene expression measurements. The idea is, that when $M(X,Y) = H(X)$, then Y completely determines X. Rewriting $M(X,Y)$ according to equation (2.7), it follows that

$$Y \text{ completely determines } X \Longleftrightarrow H(Y) = H(X,Y), \tag{2.8}$$

hence the computation of $M(X,Y)$ is not even necessary.

Now let a set of m input-output patterns $D = \{(I_1, O_1), (I_2, O_2), \ldots, (I_m, O_m)\}$ be given. REVEAL then estimates the entropies from the data, and compares the single and combined entropies $H(b)$ and $H(a,b)$ for each node a and each subset of the genes b. If b exactly determines a, that is, if $H(b) = H(a,b)$, then a corresponding rule is added to the network. The pseudocode for REVEAL is given in Algorithm 1.

The worst-case running time of REVEAL is $O(mn^{k+1})$: Time $O(m)$ to estimate the entropies from the input data, and this must be done for each node and all subsets of the nodes of size up to k (lines 1–3).

2.2.2 Advantages and Disadvantages of the Boolean Network Model

As we have seen, the Boolean network model provides a straightforward model of regulatory networks, and under the condition of bounded indegree, efficient algorithms for network inference exist. Boolean networks are attractive due to their simplicity, they are easily applied and quickly implemented. The underlying assumptions

Algorithm 1 REVEAL

```
for each node a do
  for i = 1 to k do
    for each subset b of size i of the nodes X do
      compute the entropy H(b) from the inputs
      compute the joint entropy H(a,b) from the inputs b and outputs a
      if H(b) = H(a,b) then
        b exactly determines a, add a corresponding rule to the inferred network
        proceed with the next node a
      end if
    end for
  end for
end for
```

however seem very strict, in particular, modeling genes as being in one of only two states, either *on*, or *off*, certainly is an oversimplification of true biological networks. Similarly, true networks are time-continuous and asynchronous, whereas Boolean networks assume time-discrete, synchronous updates.

Still, recent research results indicate that many biologically relevant phenomena can be explained by this model, and that relevant questions can be answered using the Boolean formalism [82]. Focusing on fundamental, generic principles rather than quantitative biochemical detail, Boolean networks can capture many biological phenomena, such as switch-like behavior, oscillations, multi-stationarity, stability and hysteresis [48, 94, 96], and they can provide a qualitative description of a system [92]. Recent modeling results combined with the first experimental techniques to validate genetic models with data from living cells show that models as simple as Boolean networks can indeed predict the overall dynamic trajectory of a biological genetic circuit [16]. It seems that for understanding the general dynamics of a regulatory network, it is the wiring that is most important, and often detailed dynamic parameters are not needed [103]. For example, Albert and Othmer [4] have predicted the trajectory of the segment polarity network in *Drosophila melanogaster* solely on the basis of discrete binary models. Similarly, Li *et al*. [57] have modeled the genetic network controlling the yeast cell cycle using a binary model.

A serious limitation of the Boolean network approach is that, although a steady state of a Boolean network will qualitatively correspond to a steady state of an equivalent continuous model based on differential equations, not all steady states of the continuous model will necessarily be steady states of the Boolean model [40]. Conversely, periodic solutions in the Boolean model may not occur in the continuous model. This problem limits the utility of Boolean modeling of gene networks [85].

Clearly, Boolean networks are not suitable when detailed kinetic parameters are desired, and the focus is on the quantitative behavior of a system. Their key advantage and limitation at the same time is their simplicity, enabling them to capture the overall behavior of a system, but limiting the analysis to qualitative aspects. On the other

hand, this simplicity allows the model to be applied to relatively large regulatory networks, when more detailed methods would be infeasible simply due to the lack of sufficient experimental data. At the same time, the simple two-state structure of each node in the Boolean network poses the problem that experimental data, which are usually measured on a continuous scale, need to be binarized, requiring delicate decisions about how this is best done.

Another shortcoming of Boolean networks is that they are deterministic in nature. However, true biological networks are known to have stochastic components, for example, proteins are produced from an activated promoter in short bursts that seem to occur at random time intervals, and probabilistic outcomes in switching mechanisms can be observed [65]. Furthermore, in realistic situations, we are usually dealing with noisy inputs and experimental measurement errors, which may lead to inconsistent data.

Finally, the dynamics of gene networks strongly depends on whether and how intra-cellular transport and diffusion of RNA and protein are modeled [60,61], which seems to play a particularly important role in eukaryotic cells [85]. The incorporation of such processes in Boolean network models is difficult, if not impossible [85].

2.2.3 Extensions of the Boolean Model

Several extensions of the Boolean network model have been proposed to overcome some of its limitations. To overcome the problems stemming from noisy and inconsistent data, from a learning-theoretic perspective, one relaxes the consistency problem to find a network that makes as few errors as possible. The resulting problem is known as the best-fit problem [17, 82] and is underlying many algorithms in machine learning.

To deal with the probabilistic nature of gene expression data, a popular extension of Boolean networks are the so-called *Probabilistic Boolean Networks* (PBN) [81]. The basic idea of PBNs is to aggregate several Boolean functions together, so that each can make a prediction of the target genes. One then randomly selects one of the functions, with probability being proportional to some weights assigned to the functions. PBNs can be interpreted as several Boolean networks operating in parallel, and one gets selected at random for a given time step. Thereafter, all networks are synchronized to the new state, so that each can make the next transition should it be selected [81].

Silvescu and Honavar [83] describe a generalization of Boolean networks to address dependences of genes that span over more than one time unit. Their model allows each gene to be controlled by a Boolean function of expression levels of at most k genes at T different time points, and they describe an algorithm for the inference of such networks from gene expression data. Other generalizations allow multi-valued networks, where each gene can be in one of several discrete states, and not just on or off [86].

2.3 Relevance Networks and Information Theoretic Approaches

While Boolean network models are based on the assumption that genes can only be in one of two states, *expressed* or *not expressed*, *relevance network* approaches [20] look at similarity or dissimilarity between pairs of genes on a continuous scale. Two steps are involved in network reconstruction using a relevance network approach:

1. All pairs of genes are compared using some measure of similarity or dissimilarity. For example, all genes can be compared against each other using pairwise correlation coefficients, or information theoretic measures such as mutual information can be used.
2. The complete set of pairwise comparisons is filtered to determine the relevant connections, corresponding to either positive or negative associations between genes.

The resulting network can then be represented in a graphical form. We will only briefly present one representative algorithm based on the relevance network approach, the ARACNe algorithm by Basso *et al.* [11, 63].

2.3.1 The ARACNe Algorithm

Similar to REVEAL, ARACNe (Algorithm for the Reconstruction of Accurate Cellular NEtworks) [11, 63] is based on *mutual information* to identify regulations between genes. In a first step, it also identifies statistically significant gene-gene coregulation by mutual information. ARACNe can do this for discrete and continuous random variables, mutual information is estimated using Gaussian kernel estimators [12]. The algorithm is hence not limited to Boolean networks such as REVEAL. A statistical test is then used to determine relevant edges in the network, Monte Carlo randomization of the data is used for the computation of p-values, and edges are filtered based on a p-value threshold.

In a further step, ARACNe then prunes the network to eliminate indirect relationships, in which two genes are coregulated by one or more intermediary genes. This is done using the *data processing inequality (DPI)*, which essentially states that if three random variables X, Y and Z depend from one another in a linear fashion $X \to Y \to Z$, then the mutual information $M(X,Z) \leq \min[M(X,Y), M(Y,Z)]$. This is used to find and remove indirect edges $X \to Z$ from the network.

The authors of ARACNe claim that relationships in the final reconstructed network have a high probability of representing direct regulatory interactions or interactions mediated by post-transcriptional modifiers. They show results on microarray gene expression data from human B cells, reconstructing a network with approximately $129,000$ interactions from 336 expression profiles [11].

2.3.2 Advantages and Disadvantages of Relevance Network Approaches

Similar to Boolean networks, relevance networks are relatively simple models of gene regulatory networks. They use straightforward and easy to compute measures

of pairwise similarity or dissimilarity between genes to reconstruct the network, such as correlation coefficients or information theoretic measures. In contrast to Boolean networks however, they are continuous models, that is, genes can have expression values on a quantitative scale.

One of the disadvantages of these approaches is, that they do not consider time, and thus disregard dynamic aspects of gene expression. Hence, these models can not infer causality, and it is not clear how to carry out simulations with an inferred network. Although algorithms such as ARACNe operate on a continuous scale for the gene expression levels, the method does not return any kinetic parameters, and is not based on chemical reaction kinetics. Furthermore, the relevance network approach is based on pairwise similarity only, and it may thus miss interactions between multiple genes. Finally, the choice of threshold for the inclusion of edges is somewhat arbitrary, and varying threshold parameters slightly may change the network considerably.

On the other hand, depending on the similarity measure used, relevance network approaches are less sensitive to noise than differential equations models. Although the data processing inequality used in ARACNe is not sufficient to identify indirect regulations, and hence the algorithm may sometimes remove direct relations as well, the pruning step helps the algorithm to derive sparse networks.

The simplicity of the relevance network approach makes it applicable to large networks. ARACNe, for example, is an algorithm with polynomial time complexity, and the authors report its use on networks with several hundred genes [11, 64]. It remains to be seen, how reliable the inferred interactions are for such large-scale applications.

2.4 Bayesian Networks

While Boolean networks assume a fixed functional dependence between different nodes, *conditional models* look at statistical correlation between genes. Conditional models try to explain the correlation between two genes by other genes in the network. These models are particularly simple in the Gaussian setting, since in this case networks can be learned from data using classical statistical tests [33, 84]. The most popular conditional model is the Bayesian network model, which is widely used to model and infer gene regulatory networks [69].

Definition 4 (Bayesian Network). *A* Bayesian Network *is a directed, acyclic graph $G = (X, A)$, together with a set of local probability distributions P. The vertices $X = \{X_1, \ldots, X_n\}$ correspond to variables, and the directed edges A represent probabilistic dependence relations between the variables. If there is an arc from variable X_i to X_j, then X_j depends probabilistically on X_i. In this case, X_i is called a* parent *of X_j. A node with no parents is* unconditional. *P contains the local probability distributions of each node X_i conditioned on its parents, $p(X_i|parents(X_i))$.*

Figure 2.3 shows a simple Bayesian example network with three nodes A, B and C, each assumed to be in one of two states, either *on* or *off*. The conditional

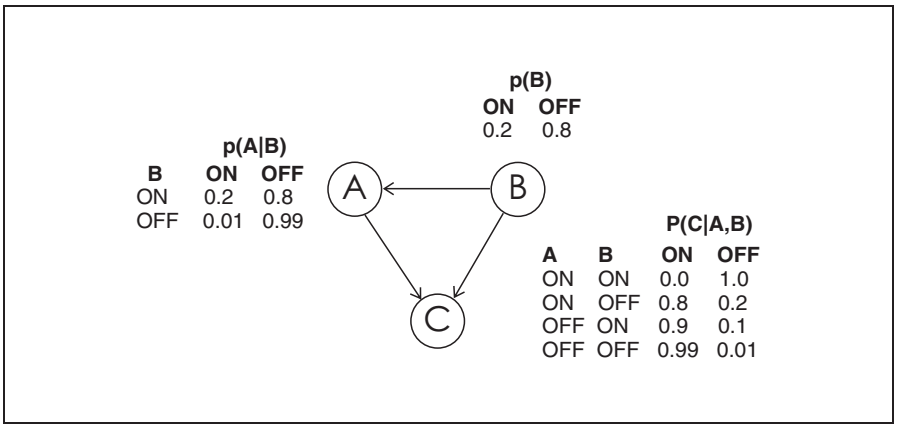

Fig. 2.3. Sample Bayesian Network with three nodes with two possible states each (ON and OFF). Given next to each node are the conditional distributions for the node, conditioned on its parents, as indicated by the arcs. For example, the probability that A is *off* given that B is *on*, $p(A = \text{off}|B = \text{on})$ is 0.8

probabilities $p(A|B)$, $p(C|A,B)$ and the unconditional probability $p(B)$ in this binary case are easily tabulated, as shown in the figure.

Note that the probability distributions of the nodes in Bayesian networks can be of any type, and need not necessarily be restricted to discrete or even binary values as in our example.

Given a Bayesian network, it is easy to compute the joint probability distribution of all variables in the network:

Definition 5 (Joint Distribution). *The* joint distribution *of a set of variables* $X_1, X_2, ..., X_n$ *is the product of the local distributions,*

$$p(X_1, X_2, ..., X_n) = \prod_{i=1}^{n} p(X_i|parents(X_i)). \tag{2.9}$$

In our example, the joint probability distribution is given by

$$p(A,B,C) = p(B)p(A|B)p(C|A,B), \tag{2.10}$$

and, for example, the joint probability that all nodes are *on* is $p(A = \text{on}, B = \text{on}, C = \text{on}) = p(B = \text{on})p(A = \text{on}|B = \text{on})p(C = \text{on}|A = \text{on}, B = \text{on}) = 0.2 \times 0.2 \times 0.0 = 0.0$. It is important to note at this point that the joint probability distribution can only be resolved this way if the network does not contain any directed cycles.

Bayesian networks provide a graphical representation of statistical dependences between variables, but more importantly, they also visualize independence relations among variables. Conditional independence of variables is represented in the graph by the property of *d-separation*, for *directional separation* [70]:

Definition 6 (d-separation). *Let a Bayesian network $G = (X,A)$ with local probability distributions P be given. Two nodes X_i and X_j, $i \neq j$, are* d-separated *in the graph G by a given set $S \subseteq X \setminus \{X_i, X_j\}$ of the nodes X, if and only if the variables X_i and X_j are independent given the values of the nodes in S.*

Informally, d-separation means that no information can flow between nodes X_i and X_j, when the nodes in S are observed. X_i and X_j are independent conditional on S if knowledge about X_i yields no extra information about X_j, once the values of the variables in S are known.

Two Bayesian network structures may actually represent the same constraints of conditional independence – the two networks are equivalent. For example, the structures $X_1 \rightarrow X_2 \rightarrow X_3$ and $X_1 \leftarrow X_2 \leftarrow X_3$ both represent the assertion that X_1 and X_3 are conditionally independent given X_2 [46]. When inferring networks from data, we cannot distinguish between equivalent networks, that is, causal relationships cannot be derived. This should be kept in mind when working with Bayesian networks – the best we can hope for is to recover a structure that is in the same equivalence class as the true network. Formally, Bayesian networks in the same equivalence class can be characterized as having the same underlying undirected graph, but may disagree on the direction of some edges. See for example [72] for details.

With these definitions and precautions at hand, we now come to the problem of learning a Bayesian network from given data.

2.4.1 Learning a Bayesian Network from Data

Learning a Bayesian network from given data requires estimating the conditional probability distributions and independence relations from the data. In order to do this, we would have to test independence of a given gene pair from every subset of the other genes. Examples for such *constraint based learning* approaches are given, for example, in [70] for networks involving only a few genes. For bigger networks, this approach quickly becomes infeasible, simply because of the number of tests that would be required.

The difficult issue is the decomposition of the joint probability distribution into conditional distributions among the relevant variables. This decomposition yields the network topology, estimating the distinct conditional probability distributions given the dependence structure is then relatively easy. In fact, given that the identification problem for Boolean networks is NP-hard, it is probably no surprise that inferring the dependence structure of a Bayesian network from given data is NP-hard as well. For this reason, the inference problem is usually tackled using heuristic approaches. Methods used include Bayesian and quasi-Bayesian approaches [19, 46, 55, 88, 91] as well as non-Bayesian methods [71, 89]. In the following, we will focus on the Bayesian approach. We will discuss the problems of structure learning and parameter learning for Bayesian networks in turn, starting with the easier parameter learning problem.

Learning Probabilities for a given Network Topology

Assume we are given the graph $G = (X, A)$, and all we ask for is details of the conditional distributions P. Let us furthermore assume that the conditional distributions are parameterized by some parameter vector ω, and that the general form of the distribution is known. Hence, we are asking for the values of ω given example data $\mathscr{D}$ assumed to have been generated by an underlying Bayesian network with topology G. The Bayesian approach then is to ask for the *posterior* distribution of the parameters, given the network topology and the data. Using Bayes' theorem,

$$p(\omega|\mathscr{D}, G) = \frac{p(\mathscr{D}|\omega, G)p(\omega|G)}{p(\mathscr{D}|G)}. \tag{2.11}$$

The *evidence* $p(\mathscr{D}|G) = \int p(\mathscr{D}|\omega, G)p(\omega|G)\,d\omega$ averages over all possible parameters ω and normalizes equation (2.11). It can be neglected when scoring parameter values relative to one another, since it is independent of ω. The *likelihood* $p(\mathscr{D}|\omega, G)$ describes the probability that a network with given structure G and parameters ω has generated the data $\mathscr{D}$, and will depend on the functional form of the local distributions P used in the Bayesian network, for example normal distributions or discrete distributions. Finally, $p(\omega|G)$ is a *prior* distribution on the network parameters ω, and is often chosen to be conjugate to the likelihood for computational reasons. If prior knowledge is available here, this can easily be included in the Bayesian network framework through $p(\omega|G)$.

Heckerman [45] gives the example of multinomial distributions $p(\mathscr{D}|\omega, G)$, hence each node is assumed to be discrete, having r_i possible values $x_i^1, \ldots, x_i^{r_i}$. Under the assumption that there are no missing data in $\mathscr{D}$ and furthermore assuming that the parameters of the multinomial distributions for the different nodes are independent from one another, the computation of the posterior distribution is easy when a Dirichlet prior is used. In this case, the posterior distribution can be shown to be a Dirichlet distribution as well. One can then maximize the posterior to find the most likely parameters ω of the network, or average over possible configurations of ω to obtain predictions for the next state of the network.

Learning the Network Topology

Let us now consider the problem of learning the structure of a Bayesian network from given data. To evaluate different structures, we consider the posterior probability of a network topology G given the data $\mathscr{D}$:

$$p(G|\mathscr{D}) = \frac{p(\mathscr{D}|G)p(G)}{p(\mathscr{D})}. \tag{2.12}$$

The term $p(\mathscr{D})$ is the evidence, and can be written as an average $p(\mathscr{D}) = \int p(\mathscr{D}|G)p(G)\,dG$ over all possible model structures. Again, when scoring network structures relative to one another, we need not compute it and can neglect this term.

The likelihood $p(\mathscr{D}|G)$ can be computed by marginalizing $p(\mathscr{D}|\omega,G)$ over all possible parameters ω of the local distributions,

$$p(\mathscr{D}|G) = \int p(\mathscr{D}|\omega,G)p(\omega|G)\,d\omega, \tag{2.13}$$

hence the local parameters ω are treated as nuisance parameters and are integrated out.

Finally, $p(G)$ is a prior distribution over network structures. In principle, this prior can be used to encode any biological knowledge that is available on the system under consideration. The simplest conceivable structure prior is to assume that every structure is equally likely. Alternatively, a structure prior can be defined by assigning confidences $0 < w(x,y) \leq 1$ to the edges (x,y) of the fully connected graph, and scoring structures using the prior

$$p(G) = \frac{1}{N} \prod_{(x,y)\in A} w(x,y), \tag{2.14}$$

where N is a normalizing constant to make the right hand side a proper distribution, and A is the set of directed edges (arcs) of the network. Many alternative structure prior distributions have been proposed in the literature. For example, Heckerman *et al.* [47] suggest using a prior network and penalizing the prior probability of any structure according to some measure of deviation between the prior network and the topology of interest. Madigan *et al.* [59] describe an approach to elicit prior knowledge from experts and encode it into the prior. Bernard *et al.* [14] use transcription factor binding site information to define a prior distribution, thus including knowledge from other data sources into the network inference.

Different strategies can then be employed to search the model space for the network topology with highest posterior probability given the data. Exhaustive search is usually prohibitive, since the number of possible network topologies with n variables is equal to the number of acyclic directed graphs with n nodes, which is growing exponentially with n [76]. This is why researchers have used heuristic search algorithms, such as greedy search, simulated annealing, gradient descent procedures, genetic algorithms and Monte Carlo methods [25].

2.4.2 Advantages and Disadvantages of the Bayesian Network Model

Bayesian networks are attractive models for gene regulatory networks since they are stochastic in nature. They can thus deal with noisy measurements and stochastic aspects of gene expression in a natural way [29, 65], and they are easily extended to deal with missing data [45]. Furthermore, they provide an intuitive and easy to grasp visualization of the conditional dependence structure in given data, and are much easier for humans to understand than full conditional distributions. At the same time, depending on the probability distributions used (continuous or discrete), they can model quantitative aspects of gene regulatory networks.

Still, the level of detail they provide on the system modeled is rather coarse [29]. Furthermore, learning Bayesian networks from data is NP-hard, hence heuristic

search methods have to be used, which do not guarantee that the globally optimal solution is found [29]. Probably their main disadvantage is that they disregard dynamical aspects completely, and that they require the network structure to be acyclic, since otherwise the joint distribution cannot be decomposed as in equation (2.9). However, feedback loops are known to play key roles in causing certain kinds of dynamic behavior such as oscillations or multi-stationarity [44, 48, 85, 94, 96], which cannot be captured by the Bayesian network model. In spite of these limitations, Bayesian networks have been used for example to infer regulatory interactions in the yeast cell cycle [36, 87].

2.4.3 Extensions of Bayesian Networks

Efforts have been made to overcome the mentioned limitations. Bayesian networks can be extended to capture the dynamic aspects of regulatory networks by assuming that the system evolves over time. Thus, gene expression is modeled as a time series, and one considers different vectors $X(1), ..., X(T)$ at T consecutive time points. One then assumes that a variable $X_i(t)$ of a particular gene i at time t can have parents only at time $t-1$. The cycles in the Bayesian network then unroll, and the resulting network is acyclic and the joint probability in equation (2.9) becomes tractable again. The resulting networks are called *Dynamic Bayesian Networks* [37, 67, 106].

Dynamic Bayesian Networks have been combined with *hidden variables* to capture non-transcriptional effects [73]. Similarly aiming at the inclusion of information from additional data sources into the Bayesian network learning process, Bernard and Hartemink [14] include transcription factor binding location data through the prior distribution, while evidence from gene expression data is considered through the likelihood.

Other extensions of Bayesian networks try to deal with the typical setting encountered with microarray data – where many genes are measured, but only few time points are available. *Regularization* approaches are used to avoid overfitting in this situation, different methods have been proposed for Bayesian networks. For example, Steck and Jaakkola [90] discuss parameter choices for a Dirichlet prior for the marginal likelihood (2.13), and show that sparse networks are learned for specific choices of parameters. Bulashevska and Eils [18] achieve regularization by constraining the form of the local probability distributions, they restrict interactions to Boolean logic semantics, and utilize Gibbs sampling to learn the model from the data.

2.5 Quantitative Models using Ordinary Differential Equations

We have seen that Bayesian networks highlight the stochastic nature of gene regulation, but are static models since they comprise no explicit time dependence in their definition. In contrast, we will now turn to *ordinary differential equations* (ODEs), which provide a deterministic, quantitative description of the time evolution of a system. ODEs are used in many scientific fields to describe a system's dynamic behavior. They provide a detailed time- and state continuous description of

the system under consideration. In recent years, ODEs have also been established as models for gene regulatory networks, ranging from simple linear models to complex nonlinear systems.

We start with a formal definition of a continuous dynamical system:

Definition 7 (Continuous dynamical system). *A* continuous dynamical system *is a triple* (U, Φ^t, T). *The* state space U *is an open subset of* $\mathbb{R}^n$ *and the set* $T \in \mathbb{R}$ *is the set of time points. The function* Φ^t *is called* evolution function *and maps for every time point* $t \in T$ *a state* $x \in U$ *onto a state* $x \in U$, *hence* $\Phi^t : T \times U \to U$. Φ^t *is assumed to be a smooth function.*

In our models, $T = \mathbb{R}$, and then Φ^t is called a *flow*. It is assumed to be the solution of an autonomous first order differential equation of the form

$$\dot{x}(t) = f(x(t)), \qquad \text{where } x(t) \in U,\ f \in \mathscr{C}^1. \tag{2.15}$$

We assume the vector field $f(x(t))$ to be continuously differentiable, that is, $f(x(t)) \in \mathscr{C}^1$, since this guarantees uniqueness of a solution of equation (2.15), given an initial state $x(t_0)$. In gene regulatory network models, the state vector $x(t)$ contains concentrations of all n network components at time t. Hence, the state space U is often restricted to the positive quadrant $U = \mathbb{R}^n_+$.

Several suggestions have been made how to choose the function $f(x(t))$, we will highlight the main models in the following.

Linear Models

Chen *et al.* [24] in 1999 were among the first to use ordinary differential equations to model gene expression networks. They used a simple linear function $f(x(t)) = Ax(t)$ with an $n \times n$-matrix A with constant entries. Here, every regulation in the network is described by one single parameter a_{ij}, one thus has to estimate n^2 parameters to infer the corresponding network structure. Linear ODEs have the advantage of being analytically tractable, thus time-consuming numerical integration can be avoided. On the other hand, systems of the form $\dot{x}(t) = Ax(t) + b$ do not show a rich variety of dynamic behavior. They only have one isolated *stationary state* $x_s = -A^{-1}b$ in which the temporal change of x vanishes. Once reaching this state, the concentrations of the network components remain constant. (This is the usual case when A is invertible. If A^{-1} does not exist, the situation is more complicated, since the equation $\dot{x}(t) = 0$ then either has no solution or many non-isolated stationary states). If x_s is stable in the sense that small perturbations of the system at rest in x_s disappear and the system returns to x_s after some time, it is globally stable, that is, the system eventually approaches this state from any initial concentration vector $x(t_0)$. If x_s is not stable, then the solution x is not bounded, leading to infinitely increasing or decreasing concentrations. For these reasons, linear models are not well suited for regulatory networks, in which the concentrations are expected to be bounded and should not become negative. Furthermore, *oscillations* or *multi-stationarity*, which are both important properties of true biological networks, are nonlinear phenomena and cannot be captured with linear models.

Nevertheless, linear models are still used to reverse engineer gene regulatory networks from experimental data, in particular for large systems including a lot of genes (see, for example, [10, 43, 56, 77, 99]). Gustafsson *et al.* [43] try to infer the regulatory network of the yeast cell cycle using microarray measurements of the whole yeast genome, which contains about 6000 genes. They argue, that even if the nature of interactions between genes is nonlinear, it can be approximated around a specific working point with its linearization, which then provides a good starting point for further considerations.

Additive Models based on Chemical Reaction Kinetics

For smaller networks, more detailed and complex models with more free parameters are feasible. Thus instead of linear models, for networks containing only a few components, ODEs of the form

$$\dot{x}_i(t) = \sum_{j=1}^{n} f_{ij}(x_j(t)) - \gamma_i x_i(t) \qquad i = 1, \ldots, n \tag{2.16}$$

with nonlinear *regulation functions* $f_{ij} : \mathbb{R} \to \mathbb{R}$ and a first order degradation term $\gamma_i x_i(t)$ are frequently used (see, e.g. [31]).

Like linear models, these are *additive models*, where the influences of different regulators are added and are thus assumed to act independently. This is often a necessary simplification to keep the number of variables tractable, but in fact numerous effects within a cell are non-additive. For example, some proteins form multimers and only become functional in these complexes, several different transcription factors can compete for a single binding site, or they act in cooperation and amplify each other. Efforts have been made to overcome these limitations of additive models, and cooperative effects are described as logical AND and OR gates, respectively [6, 79]. However, including interactions between different regulators makes the model far more complicated since the regulation functions $f_{ij}(x_j)$ then become multi-dimensional.

The regulation functions $f_{ij}(x_j)$ describe the effect of a regulator j on the temporal change of the concentration of component i. According to equation (2.16), a gene regulatory network is characterized when all individual dependences between regulated components and regulators, that is, between $\dot{x}_i$ and x_j, are known. Many efforts have therefore been made to derive an appropriate parameterization of a regulation function. These approaches are often based on chemical reaction kinetics, in which the binding process of a transcription factor TF to a specific binding site BS of the DNA is considered a reversible chemical reaction with reaction constant K:

$$TF + BS \overset{K}{\rightleftharpoons} \underbrace{TF\text{-}BS\text{-complex}}_{C} \tag{2.17}$$

The temporal changes of concentrations over time are expressed with differential equations:

$$\frac{d}{dt}[TF] = -k_1[TF][BS] - k_2[C] \tag{2.18}$$

$$\frac{d}{dt}[BS] = \frac{d}{dt}[TF] \tag{2.19}$$

$$\frac{d}{dt}[C] = -\frac{d}{dt}[TF] \tag{2.20}$$

Here, $[\cdot]$ denote concentrations, and k_1 and k_2 are rates for complex formation and dissociation, respectively. Solving for the reaction constant $K = k_1/k_2$ in equilibrium leads to the following relation between K and the steady state concentrations of all components involved in the reaction, known as the *law of mass action*:

$$K = \frac{[C_s]}{[TF_s][BS_s]} \tag{2.21}$$

Rewriting and substituting the difference between the total concentration of binding sites and that of the free binding sites for the complex concentration $[C_s]$, that is, inserting $[C_s] = [BS_t] - [BS_s]$, leads to

$$1 - \frac{[BS_s]}{[BS_t]} = \frac{[BS_b]}{[BS_t]} = \frac{[TF_s]}{K^{-1} + [TF_s]}. \tag{2.22}$$

The fraction of occupied binding sites $[BS_b]$ thus increases hyperbolically with the transcription factor concentration.

For one single binding site, the left hand side of equation (2.22) can be interpreted as the probability of this site to be occupied by a transcription factor. Therefore, when the number of free transcription factors far exceeds the number of bound ones, and thus the number of free transcription factors can be approximated with the total amount of transcription factors, $[TF_s] \approx [TF_t]$, the probability P_C of the binding site to be occupied can be written in terms of $[TF_t]$ as

$$P_C([TF_t]) = \frac{[TF_t]}{[TF_t] + K^{-1}}. \tag{2.23}$$

This probability is proportional to the effect on the transcription rate of x_i and also to the amount of protein, provided that mRNA lifetime and translation rates are constant, leading to the following parameterization:

$$\dot{x}_i(x_j) = k_{ij}\frac{x_j}{x_j + K^{-1}}. \tag{2.24}$$

Here, we have changed the notation according to equation (2.15). Relation (2.24) is known as the *Michaelis Menten kinetics* [6]. Taking also cooperative effects between several transcription factors x_j into account, we can write the regulation function as a *Hill function*

$$f_{ij}(x_j) = k_{ij}\frac{x_j^{m_{ij}}}{x_j^{m_{ij}} + \theta_{ij}^{m_{ij}}} \tag{2.25}$$

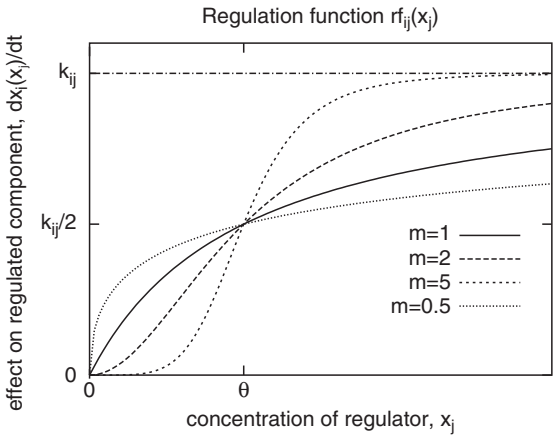

Fig. 2.4. Sigmoidal regulation function according to equation (2.25)

with *Hill coefficients* m_{ij} accounting for cooperativity between transcription factors, and with threshold values θ_{ij} which are related to the reaction constant K in equation (2.24) (see [31]). Function (2.25) is monotonically increasing or decreasing, and approaches the *regulation strength* k_{ij} for large concentrations x_j. The coefficient k_{ij} is positive when j activates i, zero when the concentration of j has no influence on i, and negative in case of an inhibition. Figure 2.4 illustrates equation (2.25) for the case of an activation with different values for the Hill coefficients m. A coefficient $m = 1$ corresponds to independent regulation (according to equation (2.24)). An exponent $m > 1$ indicates cooperative interaction between transcription factors x_j, causing a sigmoidal shape. Compared to $m = 1$, the effect on the regulated component is lower for small regulator concentrations, but increases quickly around the threshold θ, so that it exceeds the curve for $m = 1$ for concentrations $x_j > \theta$. When transcription factors influence each other negatively, for example they compete for a single binding site, this is expressed by an exponent $m < 1$, and the corresponding curve shows a steep slope for low regulator concentrations. It rapidly flattens for higher concentrations due to mutual inhibition.

To our knowledge, equation (2.25) was first proposed by Jacob and Monod in the year 1961 [49], and experiments carried out by Yagil and Yagil in 1971 supported the theory [105]. The latter estimated values of Hill coefficients and dissociation constants for different enzymes in *Escherichia coli*, one of the best studied bacterial model organisms, which is found, for example, in the human gut.

Let us stop here for a moment and reconsider the modeling approach according to equation (2.15). Although it looks rather general, it implies that there is a functional relation between the state of the system at time t, that is, the concentration vector $x(t)$ in our case, and the temporal change of this state at time t. This is a strong assumption which underlies all approaches used for network inference from expression data. For models based on chemical reaction kinetics, it implies that regulating reactions are in chemical equilibrium, otherwise there would be no unique relation between $x(t)$ and

$\dot{x}(t)$. This assumption is feasible for gene regulatory networks, when one considers the time scales in the system: Regulation via binding of a transcription factor to DNA happens at a time scale of seconds, and is thus much faster than the whole gene expression process, which lasts several minutes or even hours [6].

From a mathematical point of view, one of the main requirements on the regulation functions is that they should be bounded (concentrations should not rise to infinity) and monotone. Different parameterizations are used in the literature to guarantee these properties. Equation (2.25) is the direct result from chemical reaction kinetics, but exponents such as the Hill-coefficient m_{ij} are often hard to estimate from a numerical point of view. Thus other parameterizations such as $k_{ij}(1+e^{-x_j})^{-1}$ [22, 104] or $k_{ij}\tanh(x_j)$ [32] can be found. Several authors use step functions of the form

$$f_{ij}(x_j) = \begin{cases} 0 & \text{if } x_j \leq \theta_{ij} \\ k_{ij} & \text{otherwise} \end{cases} \tag{2.26}$$

to approximate equation (2.25) [30, 31, 34, 40, 66, 92]. This is the limit function for large Hill coefficients $m \to \infty$, and these models are known as *piecewise linear differential equations* (PLDEs). Equation (2.26) provides a partition of the state space into cuboids, separated by the threshold values θ_{ij}. Within each cuboid, the model is linear and thus analytically tractable. On the other hand, problems concerning the behavior of the system at the thresholds θ_{ij} can occur and may lead to additional steady states or limit cycles [30,31]. Note also that a step function is not differentiable at the thresholds and therefore does not satisfy the conditions in system (2.15).

In contrast to simple linear models, systems of the form (2.16) with bounded regulation functions are stable in the sense that there exists a trapping region in state space which eventually attracts all trajectories. This is an important feature in order to provide a global description of the biological system. Furthermore, monotonicity of the regulation function leads to a Jacobian matrix with constant signs (It should be noted at this point that positive self-regulation might lead to exceptions from this rule and must be treated carefully in this context – It can lead to changing signs of the Jacobian matrix depending on the location in state space, and thus statements about systems with constant J hold only for the parts of the state space in which J has constant signs). For ODE systems with positive Jacobian matrix, important statements about their dynamic behavior can be made. For example, Gouzé and Thomas emphasized the role of feedback circuits in the corresponding interaction graph [41, 93]. A positive circuit is required for multi-stationarity or hysteresis, and a negative feedback loop with at least two components is needed for periodic behavior. Thus feedback mechanisms in regulatory networks are fundamental network substructures which are related to certain dynamic behavior.

S-Systems

A further widely used class of ordinary differential equation models are *S-systems* [100], in which regulatory influences are described by power law functions:

$$\frac{dx_i(t)}{dt} = \alpha_i \prod_{j=1}^{n} x_j(t)^{g_{ij}} - \beta_i \prod_{j=1}^{n} x_j(t)^{h_{ij}} \tag{2.27}$$

The *kinetic orders* g_{ij} and $h_{ij} \in \mathbb{R}$ and the *rate constants* α_i and $\beta_i \geq 0$ have to be estimated in these models, these are $2n^2 + 2n$ parameters. The first term describes the effect of positive regulators, the second one refers to inhibitors, respectively. In contrast to additive models, here, single influences are multiplied. S-systems have been shown to capture many relevant types of biological dynamics [53]. A hyperbolic regulation such as described by equation (2.24) can be well approximated with exponents 0.5 and -0.5, respectively [95]. Steady states of (2.27) can be determined analytically, making these models attractive for network inference. Nevertheless, most of the model parameters are exponents, which are typically hard to estimate numerically. Cho *et al.* [26] and Kikuchi *et al.* [53] have used S-systems to reconstruct regulatory networks from gene expression data with genetic algorithms. Thomas *et al.* [95] developed an algorithm to estimate the rate constants from experimental data. They evaluated their approach with a simulated three gene system.

2.5.1 Network Inference

We now turn to the network inference problem for ordinary differential equation models, which is usually formulated as an optimization problem with an objective function that is minimized with respect to the network parameters ω. A common choice for this objective function is the sum of squared errors between measurements and model predictions. The corresponding optimization problem has the form

$$\min_{\omega} \left(F_1(\omega) = \sum_{t=1}^{T} \sum_{i=1}^{n} \| x_{i,model}(\omega,t) - x_{i,exp}(t) \|^2 \right). \tag{2.28}$$

Here, $x_{i,model}(\omega,t)$ denotes the model prediction for the concentration of network component i at time t, which is compared with the corresponding experimental result $x_{i,exp}(t)$. In order to minimize F_1 with respect to the parameter vector ω, numerical integration of the system is required to calculate $x_{i,model}(\omega,t)$. Usually, optimization of equation (2.28) can not be carried out analytically, and one has to apply heuristic methods such as gradient descent or genetic algorithms. This means that the numerical integration has to be carried out several times, and computing time quickly becomes the limiting factor [101]. This can be avoided by optimizing the sum of squared errors of time derivatives rather than of the concentrations directly:

$$\min_{\omega} \left(F_2(\omega) = \sum_{t=1}^{T} \sum_{i=1}^{n} \| \dot{x}_{i,model}(\omega,t) - \dot{x}_{i,exp}(t) \|^2 \right) \tag{2.29}$$

In this formula, $\dot{x}_{i,model}(\omega,t)$ is obtained from the model equations, and $\dot{x}_{i,exp}(t)$ is the corresponding slope estimate from the experimental data. Contrary to the minimization problem (2.28), solving problem (2.29) does not require numerical integration of the ordinary differential equations. Instead, one needs an appropriate method to

estimate the slopes $\dot{x}_{i,exp}(t)$ from the data. For this purpose, it can be useful to smooth the data in a preprocessing step, in particular in case of high levels of noise in the data.

Since quantitative models such as ordinary differential equation models depend on many parameters, but the number of samples available for parameter estimation is usually small in comparison, the main problem in this setting is *overfitting*. This means that the model is overtuned to specific characteristics of the training data, which do not reflect actual properties of the true underlying model, but are noise. Such overfit models will show bad performance on validation data which has not been used for training.

Different algorithms have been proposed to counter overfitting. *Early stopping* divides the data into three classes. The training data are used for learning, and this process is interrupted by testing performance of the learned model on the validation set. The procedure is stopped when performance on the validation data does not improve any further. As the result depends on both, training- and validation data, a third dataset is required to validate the inferred model.

Another method, called *weight decay* in the context of neural networks, regularizes the objective function by adding a term which penalizes models with many degrees of freedom. Popular criteria used for this purpose are *Akaike's information criterion* (AIC) [1]

$$F_{AIC} = -2\ln\mathscr{L} + 2k \tag{2.30}$$

and the *Bayesian information criterion* (BIC)

$$F_{BIC} = -2\ln\mathscr{L} + k\ln(n), \tag{2.31}$$

where in both equations k is the number of free model parameters, $\mathscr{L}$ the value of the error function and n the sample size. These criteria were used as objective functions in the inference of the yeast cell cycle network in Nachman *et al.* [68] and Chen *et al.* [22], respectively.

More biologically motivated approaches restrict the search space by including biological knowledge into the learning process. This can be done by introducing constraints to the optimization problem, such as upper limits for the number of regulators for every gene, or ranges for model parameters. Alternatively, similar to the criteria introduced above, one can modify the objective function by adding a term penalizing networks with a large number of strong regulatory interactions (see, for example, [98]). In Section 2.6, we will introduce an inference method which uses this latter approach.

2.5.2 Advantages and Disadvantages of ODE Models

Continuous dynamical systems provide a very detailed quantitative description of a network's dynamic, as they are time- and state-continuous models. They can show a rich variety of dynamic behaviors, such as multi-stationarity, switch-like behavior, hysteresis or oscillations. For nonlinear systems based on chemical reaction kinetics, parameters can directly be interpreted as kinetic rates of a chemical or physical

reaction, for example, as degradation rates or velocities of binding reactions [44]. Some of these rates are experimentally accessible, which provides either a possibility to evaluate the model afterwards, or to restrict the range of the corresponding parameter values prior to the inference process. For example, binding coefficients between macromolecules can often be measured in vitro, and they differ only slightly in vivo [85]. Other rate constants, such as rates of phosphorylation of a transcription factor subsequent to a stimulus, are hard to verify experimentally [15, 29, 85].

When analyzing ODEs, one can exploit the well-established theory of differential equations [42]. To examine, for example, the long-term behavior of an ODE system, methods have been developed to calculate steady states or limit cycles and to determine their basins of attraction. *Bifurcation analysis* aims at predicting parameter values for which the qualitative behavior of the system changes, because the stability of steady states or periodic solutions changes when varying parameters, or solutions appear and disappear. Many tools have been developed to conduct such an analysis numerically [35].

A drawback of differential equation models is the relatively large number of parameters which have to be estimated in the network inference process. Time courses with many time points are needed for this purpose, but such data is rarely available. Many high-throughput techniques aim at measuring a lot of components simultaneously, but good time resolution is hard to obtain. This is the main reason why inference of ODEs from experimental data is currently restricted to small networks with only few components.

Another problem lies in the quality of experimental data. Microarray data are mostly used to infer gene regulatory networks. They contain a lot of noise, and may not be appropriate to make quantitative statements. Thus, when modeling regulatory networks with differential equations, it is often inevitable to include prior biological knowledge or to make simplifying assumptions. Of course, this often makes the approach specific for a certain biological system and not ad hoc applicable to other organisms or subsystems.

2.6 Bayes Regularized Ordinary Differential Equations

We would now like to give an example from our own work, combining ordinary differential equations with a (dynamic) Bayesian network approach. The underlying model used is a system of differential equations, but we embed the differential equations in a probabilistic framework with conditional probabilities as in Bayesian networks, and use Bayes' theorem for the inference. In our approach, the differential equations are used to specify the mean of the conditional probability distributions for the genes at a given time point, conditioned on the expression pattern at a previous time point. We then estimate the parameters of the differential equations, and thus determine the conditional probability distributions and the network topology. This topology is assumed to be fully connected initially, but we will show how to drive the solution to sparse networks using a specifically designed prior distribution on the ODE parameters.

Two aspects make such a Bayesian approach attractive for the inference of gene regulatory networks from gene expression data. The stochastic approach captures the stochastic nature of biological processes within a cell and the noise due to the experimental procedure. Moreover, prior knowledge can be included into the posterior distribution by choosing appropriate prior distributions reflecting our knowledge of the system. Furthermore, the probabilistic nature of the approach makes it possible to compute confidences for model parameters and also predictions of the network [50], work on this is ongoing in our groups.

To become more concrete, we consider an additive ODE model with sigmoidal regulation functions of the form

$$\dot{x}_i(t) = s_i - \gamma_i x_i(t) + \sum_{j=1}^{n} k_{ij} \frac{x_j^{m_{ij}}}{x_j^{m_{ij}} + \theta_{ij}^{m_{ij}}}. \tag{2.32}$$

The parameters s_i and γ_i are basic *synthesis-* and *degradation* rates, they determine the dynamics of component i when all regulators of i are absent. Coupling of the differential equations is due to the sum of regulation functions, compare equation (2.25). The sum in (2.32) is over all genes in the network and reflects the influence of the j-th gene on gene i. The network is thus assumed to be fully connected, unless the corresponding parameters k_{ij} become zero. More details on this model can be found in [38, 74].

We discretize this equation with a simple *Euler discretization*, that is, we approximate the time derivatives on the left hand side by difference quotients, and we furthermore add a noise term $r_i(t)$ to the output. We then get

$$x_i(t+\Delta t) = \underbrace{x_i(t) + \Delta t \left[s_i - \gamma_i x_i(t) + \sum_{j=1}^{n} k_{ij} \frac{x_j(t)^{m_{ij}}}{x_j(t)^{m_{ij}} + \theta_{ij}^{m_{ij}}} \right]}_{h_i(\omega)} + r_i(t). \tag{2.33}$$

The noise term $r_i(t)$ is assumed to be normally distributed with mean 0 and variance $\sigma_i(t)^2$. The assumption of normally distributed noise corresponds to assuming that the noise stems from many small, independent sources, which is arguably a reasonable approximation at least for the experimental noise. Δt is a discretization parameter, the smaller the time step Δt, the better does equation (2.33) approximate the continuous system (2.32). Biological data sets usually comprise large time steps, when using differential equations models one therefore interpolates over time in order to get sufficiently small time resolution.

Assuming independence of all noise terms for every time point and all network components from one another, the likelihood $\mathscr{L}$ decomposes into a product over all time points $t_1, \ldots, t_T$ and all n network components:

$$\mathscr{L} = p(\mathscr{D} \mid \omega) = \prod_{z=1}^{T} \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma_i(t_z)^2}} \exp\left[-\frac{1}{2\sigma_i(t_z)^2} \left(h_i(\omega) - x_i(t_z) \right)^2 \right] \tag{2.34}$$

Clearly, the independence and normality assumptions are a simplification. Noise on $x(t)$ will lead to correlated, non-normal noise on $x(t+\Delta t)$. Furthermore, modeling

errors will accumulate over time, and are certainly correlated. On the other hand, the assumptions are probably reasonable for experimental noise, and clearly they are a trade-off between model complexity/computational tractability and a detailed and realistic model, and similar assumptions are frequently used in Bayesian learning approaches.

If no prior knowledge is included into the learning process, an optimal parameter vector ω can be computed by maximization of equation (2.34) with respect to ω. This is known as *maximum likelihood estimation* (MLE):

$$\hat{\omega}_{MLE} = \arg\max_{\omega} \mathscr{L}(\omega) \tag{2.35}$$

In case all variances are equal, that is, $\sigma_i(t) = \sigma$ for all $i = 1,\ldots,n$ and for all $t = t_1,\ldots,t_T$, $\hat{\omega}_{MLE}$ is equivalent to the result one gets when minimizing the sum of squared errors between measurements and model predictions with respect to ω. This is easily seen when taking the negative logarithm of (2.34), and dropping terms independent of ω.

To include prior knowledge into the inference process, we use Bayes' theorem to compute the posterior probability distribution, that is, the conditional distribution over the parameter vector ω, given the data:

$$p(\omega \mid \mathscr{D}) = \frac{p(\mathscr{D} \mid \omega)p(\omega)}{p(\mathscr{D})}. \tag{2.36}$$

Here, the right hand side includes a product of the likelihood $\mathscr{L}$ and the prior distribution $p(\omega)$ over the model parameters. Maximizing equation (2.36) with respect to ω once again leads to a point estimate for ω, this is known as *maximum a-posteriori* (MAP) estimation:

$$\hat{\omega}_{MAP} = \arg\max_{\omega} p(\omega \mid \mathscr{D}) \tag{2.37}$$

When no prior information about the system under consideration is available, the prior distribution $p(\omega)$ is often chosen to be an improper uniform distribution, and $\hat{\omega}_{MLE}$ then equals $\hat{\omega}_{MAP}$. In the following section, we will detail our choice of prior distribution over model parameters.

2.6.1 Prior Distributions for Network Parameters

We now need to specify prior distributions for the model parameters s_i, γ_i, k_{ij}, θ_{ij} and m_{ij}.

The parameters s_i and γ_i are basic synthesis and degradation rates for the network components. Both parameters should neither become negative nor too large. We therefore choose independent gamma distributions for these two parameters. The gamma distribution is given by

$$g(x) = \frac{a^r x^{r-1}}{\Gamma(r)} \exp[-ax]. \tag{2.38}$$

Here, $\Gamma(r)$ is the gamma function. The shape of the distribution depends on the shape parameter $r > 0$. The parameter $a > 0$ determines the scale of the distribution. The smaller a, the more spread out is the distribution. The parameters a and r must be carefully chosen depending on the numerical range of the experimental measurements, and prior knowledge on synthesis and degradation rates can be included through specific settings.

For the parameters k_{ij}, a hierarchical prior distribution is used, which has specifically been designed to favor *sparse* networks. Biologically motivated, most of the k_{ij} should be close to zero, and only few k_{ij} should differ significantly from zero – corresponding to only few significant edges in the network. This is achieved using independent mean-zero normal distributions as prior distributions on k_{ij}, with standard deviations distributed according to a gamma distribution. The idea here is that most of the normal distributions should be concentrated strongly around zero in order to keep the corresponding k_{ij} small, and should only in few cases be allowed to become wider, if the data indicates so. This expectation of narrow normal distributions is reflected by the gamma distribution on the standard deviations of the normal distributions. Combining these two distributions and marginalizing over the standard deviation s,

$$p(k) = \prod_{i=1}^{n} \int_0^\infty \mathcal{N}(k|\mu = 0, \sigma = s) g(s) ds, \tag{2.39}$$

where

$$\mathcal{N}(k|\mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2} \frac{(k-\mu)^2}{\sigma^2} \right] \tag{2.40}$$

is the normal density with mean μ and variance σ^2.

When drawing samples from the distribution $p(k)$, most of the values will be small, since they stem from normal distributions with mean zero and a small variance. Figure 2.5 shows the distribution $p(k)$ resulting from the integration (2.39) for the two-dimensional case ($n = 2$). As can be seen, this prior favors solutions were only one parameter k_i is distinct from zero over solutions where both k_1 and k_2 differ

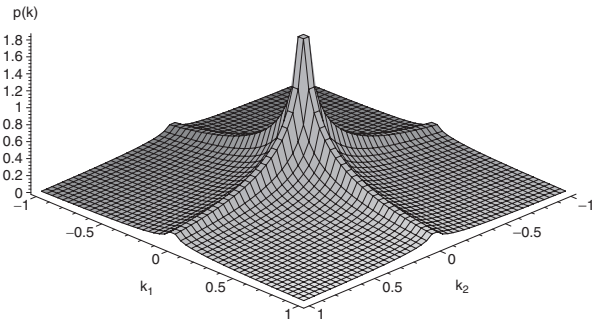

Fig. 2.5. The two dimensional hierarchical prior distribution with parameters $r = 1$ and $a = 1.0001$ for edge weights k_{ij}

significantly from zero. This is distinct from standard regularization schemes used such as the L_2 penalty, which would correspond to a Gaussian prior – and which would give the same penalty to points at equal distance from the origin, independent of the number of nonzero components. Note also that this prior is stronger than a Laplace prior on k.

At this point, we remark that the choice of prior distribution on network parameters clearly influences results of the computation, and it is not necessarily guaranteed, that this reflects biological reality. This is a classical example of the *bias-variance tradeoff*, where a stronger prior will lead to a stronger bias in learning, but less variance, and vice versa. In the setting of network learning described here, a strong prior driving the network to sparse solutions is needed to avoid overfitting of the model, this is discussed in more detail in [75], where we compare maximum likelihood and maximum a-posteriori under various settings on simulated data.

We use fixed values for the exponents m_{ij} and threshold parameters θ_{ij} for numerical reasons, this corresponds to assuming a delta distribution on these parameters. The reason for this decision is numerical instability of the optimization routine when m and θ are optimized, and insufficient experimental data to learn these parameters properly.

The negative logarithm of the posterior distribution (2.36) is then minimized using conjugate gradient descent. Alternatively, one could sample from the posterior distribution using a Markov chain Monte Carlo approach, work on this is presently ongoing and will be published elsewhere. For technical details on both approaches see [50, 51], where the same hierarchical prior distribution as the one used here on the k_{ij} is used in combination with a Cox regression model to predict survival times of cancer patients from gene expression measurements.

2.6.2 Application to the Yeast Cell Cycle Network

In this section, we will show results of an application of the Bayesian approach described above to a dataset on the yeast cell cycle. More details as well as an additional evaluation of the method on simulated data can be found in [75].

The yeast cell cycle is one of the best studied eukaryotic regulatory systems. A proper functioning of this regulatory mechanism is essential for the organism to survive. Core elements of its machinery are highly conserved in evolution among eukaryotes, making studies on a model organism such as budding yeast worthwhile, as many results can be transfered to higher organisms. Many publications on the yeast cell cycle exist, see, for example, [8].

We examined eleven genes from a publicly available dataset by Spellman *et al.* [87], these genes are known to be central to the cell cycle [57]. The dataset contains log ratios between synchronized cells and control experiments of the whole yeast genome, approximately 6000 genes were measured several times during the cell cycle, in total over 69 time points. The reference network we use for evaluation of our results is a reduction of the network described in Li *et al.* [57].

Results

Time series data of eleven genes was used, including cln1, cln2, cln3, clb5, clb6, cdc20, cdc14, clb1, clb2, mcm1 and swi5. Measurements corresponding to nodes in the reference network involving several genes were represented by the mean value of the genes, missing values were estimated by linear interpolation over time. Conjugate gradient descent was used to fit the model to the data, with prior distribution parameters $a = 0.1$ and $r = 0.01$ for the synthesis and degradation rates, and $a = 5$ and $r = 1.7$ for the prior on the k_{ij}. Fixed values of $\theta_{ij} = 1$ and $m_{ij} = 2$ were used for the threshold parameters and Hill coefficients of the ODE model. Since we expect sparse solutions, the gradient descent was started near the origin, see [75] and [50] for technical details.

To evaluate our results, we compared the inferred network structure with the reference network. Figure 2.6 shows the reference network (*left*) and the network inferred with the Bayesian approach (*right*). The 16 edges with highest weights are marked in bold, continuous bold lines indicate true positives, dashed bold lines correspond to false positives. Thin lines appear in the reference network, but are not revealed in our approach. 12 of 16 regulations are true positives, the remaining four interactions are not stated in the literature. Note that, in the latter case, it is not clear whether there is no such regulation or whether it exists but has not been described yet. The corresponding values for *specificity*, that is, the fraction of revealed true regulations, and *sensitivity*, the fraction of true negatives, are 0.55 and 0.85, respectively.

Receiver Operator Characteristics (ROC) curves can be used to assess the inferred network structure more quantitatively. By using a cutoff value c on the weights k_{ij} and including only edges with $|k_{ij}| > c$ in the network, one can compute sensitivity and $1 -$ specificity. Sensitivity and $1 -$ specificity can then be plotted against one another for different cutoff values c, assuming that the reference network is the correct underlying network. The resulting ROC curves provide a comprehensive overview over all combinations of sensitivity and specificity that can be achieved with a given model. ROC curves can further be summarized by computing the *Area Under the ROC Curve*, the AUC. The AUC is a numerical value between 0.5 and 1, where

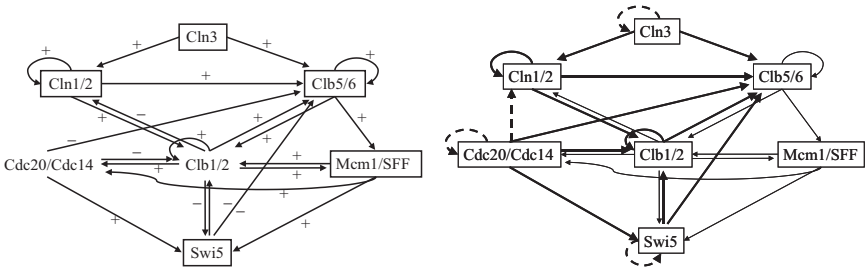

Fig. 2.6. Regulatory network of the yeast cell cycle (*left*) (see [75] and [57]) and the network inferred with the Bayesian approach (*right*). True positives are marked in bold, false positives are marked with bold dashed lines, false negatives correspond to thin lines

0.5 would be equivalent to guessing for each edge whether it is present or not, and an AUC of 1 would correspond to a prediction with perfect sensitivity and specificity. We computed AUC values for our approach, the corresponding AUC value is 0.68, indicating that the main regulatory interactions are revealed. The approach outperforms maximum likelihood estimation, which yields an AUC value of 0.61, showing that the sparsity constraint introduced through the prior distribution helps the learning process.

Computational demands of the approach are comparable to demands of other differential equations model approaches. For small networks with 5 to 10 nodes, running times are typically in the range of a few minutes, depending on the number of time points available. For large networks, the limiting factor is usually not computing time, but insufficient amounts of data to reliably estimate parameters of the differential equations.

2.7 Other Approaches

In this section, we will give an overview over models that go beyond ordinary differential equations. We will focus on three further model classes, delay differential equations (DDEs), partial differential equations (PDEs), and stochastic equations. DDEs are used to account for time delays in regulatory mechanisms, which is often necessary when the system consists of reactions taking place at different time scales. Spatial inhomogeneities within a cell are captured with PDEs, which contain derivatives of time and space, and include, for example, diffusion processes. Unfortunately, without further knowledge concerning diffusion coefficients and locations of transcription and translation, it is not possible to learn parameters for such models using only microarray data.

Stochastic equations try to model the stochastic nature of single reactions and provide the most detailed level of description. Here as well, far more information is needed than microarray expression data can provide. Thus, all three model classes are not ad hoc suitable for the inference of large scale regulatory networks from expression data, and have mostly been investigated only theoretically or used to model very specific regulatory mechanisms so far. No “standard method” exists to estimate parameters for these models, and we will therefore only point out some basic concepts and difficulties with these models rather than give a recipe on how to infer networks using them.

2.7.1 Delay Differential Equations

All modeling approaches discussed so far implicitly assume a local time dependence and spatial homogeneity. To include transport processes into the model, time-delay differential equations (DDEs) or partial differential equations (PDEs) are used (see, e.g. [23, 29, 85]). This is particularly interesting for eukaryotic organisms, where macromolecules such as mRNA have to be transported from one cell compartment into another prior to translation, or proteins are produced somewhere in a cell and

become active somewhere else. How important such factors may be can be seen from recent reports indicating that the spatial distribution of proteins within a cell seems to have an effect on the embryonic development in eukaryotes [29].

Active transport processes, which require energy and can be in opposition to the concentration gradient, are modeled with DDEs of the form

$$\dot{x}(t) = f(x(t), x_{delay}(t)), \tag{2.41}$$

with

$$x_{delay}(t) = \int_{-\infty}^{0} x(t-\tau)G(x(t-\tau))d\tau. \tag{2.42}$$

In these systems, the left hand side depends on the current state $x(t)$ and the state vector $x_{delay}(t)$, which is a weighted integral over past states. The sum of the weights is normalized, that is,

$$\int_{-\infty}^{0} G(x(t-\tau))d\tau = 1. \tag{2.43}$$

In the simple case where one can assume a fixed duration τ_0 between binding of a transcription factor to a binding site within a promoter of a gene and the effect it has on the amount of protein, the distribution over weights can be modeled using a delta distribution:

$$G(x(t-\tau)) = \delta(\tau_0) \tag{2.44}$$

and hence

$$\int_{-\infty}^{0} x(t-\tau)G(x(t-\tau))d\tau = x(t-\tau_0). \tag{2.45}$$

In equation (2.41), $f : \mathbb{R}^n \times C^1 \to \mathbb{R}^n$ is a functional operator which maps n continuously differentiable functions defined on $\mathbb{R}$ onto a vector in $\mathbb{R}^n$. This makes DDEs more difficult to analyze than ODEs, in which $f : \mathbb{R}^n \to \mathbb{R}^n$ is an ordinary function which maps a vector $x(t)$ onto another vector $\dot{x}(t)$.

In order to solve equation (2.41), not only an initial state vector $x(t_0)$, but an entire interval of initial data is required. Thus, the state space is infinite dimensional. This also leads to infinitely many eigenvalues when linearizing the system in order to analyze the behavior of steady states. The characteristic equation is not a simple polynomial, but involves exponential functions. No standard method to solve such equations exists, and stability analysis of steady states can be a hard task. In general, not much is known about effects that are caused by time delays. Most work in this field examines the stability of a steady state depending on time delays for a certain system. Chen and Aihara [23], for example, consider an oscillating two-gene model and claim that time delays increase the stability region of oscillations in their model, making the oscillations robust against parameter changes. Santillán and Mackey [78] built a model of the tryptophan operon in *Escherichia coli*, one of the prototypic gene

control systems. They also included time delays into their nine differential equations, and they estimated 28 parameters. Simulations were carried out numerically with a fourth order Runge-Kutta method. A lot of specific knowledge about the operon as well as steady state approximations were included into the parameter estimation processes, hence the estimation method cannot ad hoc be generalized and used for arbitrary organisms.

2.7.2 Partial Differential Equations

PDEs describe spatial inhomogeneities and diffusion processes and distinguish between different cell compartments, for example nucleus and cytoplasm [29]. The corresponding differential equations consist of a sum of functions $f_i(x(t))$, which describe the regulatory network as in equation (2.16), and a term for the diffusion process:

$$\frac{\partial x_i}{\partial t} = f_i(x(t)) + \delta_i \frac{\partial^2 x_i}{\partial l^2}, \text{ with } 0 \leq l \leq \lambda, i = 1, \ldots, n. \tag{2.46}$$

In contrast to ODEs, this equation contains derivatives with respect to both time and space. The variable δ_i is the diffusion constant, and l is the position in the cell. Boundary conditions such as

$$\frac{\partial^2}{\partial l^2} x_i(0,t) = 0 \text{ and } \frac{\partial^2}{\partial l^2} x_i(\lambda,t) = 0 \tag{2.47}$$

ensure that components stay within the cell.

The lack of appropriate analysis methods and missing experimental data providing information on transport processes make both DDEs and PDEs currently inappropriate for the inference of regulatory networks from gene expression data. Also little is known about "typical durations" of mRNA or protein transport, and data about spatial distributions of cell components is only gradually becoming available with recent developments in live-cell imaging techniques.

2.7.3 Stochastic Kinetic Approaches

Finally, a *stochastic kinetics* modeling approach provides the by far most detailed level of description [29, 44, 85], but also has the highest computational cost [44].

Probabilistic models were developed to explain the observed variety in experiments, in particular when the number of molecules is small [62]. In these models, concentrations are discrete and change according to some probability distribution. The probability of a system to be in state X at time $t + \Delta t$ is given by

$$p(X, t+\Delta t) = p(X,t) \left(1 - \sum_{j=1}^{m} \alpha_j \Delta t \right) + \sum_{j=1}^{m} \beta_j \Delta t, \tag{2.48}$$

see, for example [29]. Here, X is a discrete concentration vector and $p(X,t)$ is a probability distribution. The term $\alpha_j \Delta t$ is the probability that a reaction j takes place

in the time interval Δt, and the sum runs over all m possible reactions. The second term is the probability that the system will be brought to state X from any other state via a reaction j. Taking the limit $\Delta t \to 0$ leads to the well known *Master equation*, a first order differential equation that describes the evolution of the probability to occupy a discrete set of states:

$$\frac{\partial p(X,t)}{\partial t} = \sum_{j=1}^{m} (\beta_j - \alpha_j p(X,t)) \tag{2.49}$$

Modeling gene regulatory networks with these equations requires much more information than with ordinary differential equations since every single reaction is considered. Moreover, the computational costs are very high since a large number of simulations is needed to approximate $p(X,t)$ [44]. The Master equation can in some cases be approximated with stochastic differential equations. These so called *Langevin equations* assume that internal fluctuations cancel out on average, and the system can be described by a deterministic ODE and a noise term. Numerical solutions are obtained for these equations using Monte Carlo simulations. Alternatively, a stochastic simulation approach provides information on individual behavior instead of examining the whole distribution $p(X,t)$. Gillespie developed an efficient algorithm to describe a spatially homogeneous chemical system with a stochastic approach, the *stochastic simulation algorithm* [39], which is equivalent to the spatially homogeneous Master equation. This algorithm was used by McAdams and Arkin [65], who examined the influence of statistical variations during regulatory cascades on cellular phenomena across cell populations, and by Arkin *et al.* [7], who considered the influence of fluctuations in the rate of gene expression on the choice between lytic and lysogenic growth in phage λ. The latter is the pioneering work on the role of fluctuations in gene expression.

2.8 Conclusion

In this chapter, we have attempted to give an overview over a number of different models used for gene regulatory network reconstruction. We started with simple binary models, which assume that each gene is in one of two possible states, *expressed* or *not expressed*. We then extended the scope all the way to complex quantitative models, which can capture kinetic properties of the chemical reactions underlying gene regulation. All these models have their own specific strengths and weaknesses. So, when faced with an actual biological system to be analyzed or simulated, what is the appropriate model to use?

The answer is – it depends. It depends on the biological question we are interested in, and it will also depend on the experimental data we have at our disposition or can measure on the system considered. Furthermore, it will depend on the kind of biological knowledge we already have on the system under consideration.

There are three central questions that should be considered when choosing a model. These are:

1. What do we hope to learn from the model?
2. How large is the system we need to model?
3. Do we have the right data, and is there additional knowledge we can use?

The first question asks for the ultimate objective driving our modeling attempts. Occam's razor is the principle to choose the simplest model that can explain the data we have, and very similarly, we should choose the simplest model that can answer the questions we ask. So, if our interest is in qualitative properties such as *"Does component j interact with component i in the network?"* or *"Do two components have the same regulators?"*, then qualitative models such as Boolean networks probably provide the appropriate framework. If the questions are of a quantitative nature, such as *"What happens when changing the affinity of a transcription factor to a specific binding site?"* or *"How does a change in concentration of component i affect the dynamic behavior of the system?"*, then obviously quantitative models are required.

At the same time, one should be highly alert to the complexity of the modeling task. This brings us to the second question above. Large genetic systems are extremely difficult to model, and extrapolating a detailed differential equations model for a single gene with its several kinetic parameters to larger systems will quickly render the model prohibitively complicated [16]. The sheer quantity of parameters in such models will make their application impossible to networks involving more than just a few genes. So, there also is a tradeoff here. When the complexity of the biological system modeled is low, thus single genes or only few genes are of interest, computer modeling can go into much detail and quantitative differential equation models or even stochastic molecular simulations are feasible, permitting simulations of detailed single gene dynamics and time courses of gene activity. On the other hand, when mid-size to large genetic networks are desired, models must focus on less detail and concentrate on the overall qualitative behavior of the system. This may still allow inference about the state dynamics of a system in terms of a flow pattern grasping the qualitative aspects of state transitions, but quantitative models for the entire system are usually impossible, simply because of the lack of sufficient data to estimate all parameters in those models.

In our experience, differential equations models quickly reach their limit when more than a handful of genes are modeled, and while additional constraints such as the sparsity constraint introduced in Section 2.6 can extend the feasible network size slightly, these approaches are not useful for large-scale network inference with several hundred to thousands of components. However, they provide a very detailed, quantitative model for small networks. Bayesian networks permit slightly larger network models, but here too, one needs to be cautious about overfitting and insufficient data when more than a few dozen genes are modeled. Boolean models and relevance network approaches finally permit the largest number of genes to be included in network models, and application involving thousands of genes have been reported, see, for example, [21]. It remains to be seen how reliable such large-scale networks are.

For a numerical evaluation and comparison of different approaches on simulated data see, for example, [9].

This brings us to the third question, concerning the available data for the modeling task. The large bulk of work on transcriptional network reconstruction has concentrated on deterministic, coarse-grained models. Even when quantitative models are used, the conclusions drawn from them are usually of a qualitative nature. This is mainly due to the incomplete knowledge on the chemical reactions underlying gene regulation, and the lack of detailed kinetic parameters and concentration measurements required for these models [29]. Often, the lack of suitable data is the limiting factor in network inference. However, this can sometimes be alleviated by the inclusion of additional biological knowledge in the learning process. For example, if information on transcription factor binding sites is available, this may be used to reduce the search space for model topologies. The inclusion of such prior knowledge is an ongoing research problem. If quantitative data of good quality is available, maybe supported by additional data sources such as measurements of kinetic parameters and prior biological knowledge on interactions in the network, detailed quantitative models are often feasible [7].

Even though large scale techniques such as DNA microarrays can provide genome-wide expression measurements, microarrays provide effectively only more or less qualitative data at present. In addition, they measure many genes under few different conditions or time points, whereas for network inference, one would rather have few (relevant) genes under many conditions and time points.

However, this data bottleneck can reasonably be expected to be relieved in the near future. With the advent of novel experimental techniques to measure RNA and protein concentrations, accompanied by large databases providing access to this and other published and unpublished data, quantitative models will increasingly be used in the future, bringing us closer to the ultimate goal, the simulation of whole cells [97].

References

1. H. Akaike, *A new look at the statistical model identification*, IEEE Trans. Automatic Control **19** (1974), 716–723.
2. T. Akutsu, S. Miyano, and S. Kuhara, *Identification of genetic networks from a small number of gene expression patterns under the boolean network model*, Pac Symp Biocomput **4** (1999), 17–28.
3. ———, *Algorithms for identifying boolean networks and related biological networks based on matrix multiplication and fingerprint function*, RECOMB'00: Proceedings of the fourth annual international conference on Computational molecular biology (New York, NY, USA), ACM Press, 2000, pp. 8–14.
4. R. Albert and H.G. Othmer, *The topology of the regulatory interactions predict the expression pattern of the segment polarity genes in Drosophila melanogaster*, J Theor Biol **223** (2003), 1–18.
5. B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter (eds.), *Molecular biology of the cell*, 4 ed., Garland Publishing, New York, 2002.

6. U. Alon, *An introduction to systems biology - design principles of biological circuits*, Chapman & Hall/CRC Mathematical and Computational Biology Series, New York, 2007.
7. A. Arkin, J. Ross, and H.H. McAdams, *Stochastic kinetic analysis of developmental pathway bifurcation in phage λ - infected Escherichia coli cells*, Genetics **149** (1998), no. 4, 1633–1648.
8. J. Bähler, *Cell-cycle control of gene expression in budding and fission yeast*, Annu. Rev. Genet. **39** (2005), 69–94.
9. M. Bansal, V. Belcastro, A. Ambesi-Impiombato, and D. di Bernardo, *How to infer gene networks from expression profiles*, Molecular Systems Biology **3** (2007), 78.
10. M. Bansal, G.D. Gatta, and D. di Bernardo, *Inference of gene regulatory networks and compound mode of action from time course gene expression profiles*, Bioinformatics **22** (2006), no. 7, 815–822.
11. K. Basso, A.A. Margolin, G. Stolovitzky, U. Klein, R. Dalla-Favera, and A. Califano, *Reverse engineering of regulatory networks in human B-cells*, Nature Genetics **37** (2005), 382–390.
12. J. Beirlant, E. Dudewicz, L. Gyorfi, and E. van der Meulen, *Nonparameteric entropy estimation: An overview*, Int J Math Stat Sci **6** (1997), no. 1, 17–39.
13. P. Berg and M. Singer (eds.), *Dealing with genes*, University Science books, 1992.
14. A. Bernard and J. Hartemink, *Informative structure priors: Joint learning of dynamic regulatory networks from multiple types of data*, Pac Symp Biocomput (2005), 459–70.
15. H. Bolouri and E.H. Davidson, *Modeling transcriptional regulatory networks*, BioEssays **24** (2002), 1118–1129.
16. S. Bornholdt, *Less is more in modeling large genetic networks*, Science **310** (2005), no. 5747, 449–450.
17. E. Boros, T. Ibaraki, and K. Makino, *Error-free and best-fit extension of partially defined boolean functions*, Information and Computation **140** (1998), 254–283.
18. S. Bulashevska and R. Eils, *Inferring genetic regulatory logic from expression data*, Bioinformatics **21** (2005), no. 11, 2706–2713.
19. W. Buntine, *Theory refinement on bayesian networks*, Proceedings of the 7th Conference on Uncertainty in Artificial Intelligence (Los Angeles, CA, USA) (B. D'Ambrosio, P. Smets, and P. Bonissone, eds.), Morgan Kaufmann Publishers, 1991, pp. 52–60.
20. A. Butte and I. Kohane, *Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements*, Pac Symp Biocomput, 2000, pp. 418–429.
21. A.J. Butte, P. Tamayo, D. Slonim, T.R. Golub, and I.S. Kohane, *Discovering functional relationships between rna expression and chemotherapeutic susceptibility using relevance networks*, Proc Natl Acad Sci U S A **97** (2000), no. 22, 12182–12186.
22. K.-C. Chen, T.-Y. Wang, H.-H. Tseng, C.-Y.F. Huang, and C.-Y. Kao, *A stochastic differential equation model for quantifying transcriptional regulatory network in Saccharomyces cerevisiae*, Bioinformatics **21** (2005), no. 12, 2883–2890.
23. L. Chen and K. Aihara, *A model for periodic oscillation for genetic regulatory systems*, IEEE Trans. Circuits and Systems I **49** (2002), no. 10, 1429–1436.
24. T. Chen, H.L. He, and G.M. Church, *Modeling gene expression with differential equations*, Pac Symp Biocomput, 1999, pp. 29–40.
25. D.M. Chickering, D. Geiger, and D. Heckerman, *Learning bayesian networks: Search methods and experimental results*, Proceedings of the Fifth Conference on Artificial Intelligence and Statistics (Ft. Lauderdale), Society for Artificial Intelligence and Statistics, 1995, pp. 112–128.

26. D.-Y. Cho, K.-H. Cho, and B.-T. Zhang, *Identification of biochemical networks by S-tree based genetic programming*, Bioinformatics **22** (2006), no. 13, 1631–1640.
27. J. Collado-Vides and R. Hofestädt (eds.), *Gene regulations and metabolism - postgenomic computational approaches*, MIT Press, 2002.
28. G.M. Cooper and R.E. Hausman (eds.), *The cell: A molecular approach*, 4 ed., ASM Press and Sinauer Associates, 2007.
29. H. de Jong, *Modeling and simulation of genetic regulatory systems: A literature review*, J Comput Biol **9** (2002), no. 1, 67–103.
30. H. de Jong, J.-L. Gouzé, C. Hernandez, M. Page, T. Sari, and J. Geiselmann, *Qualitative simulation of genetic regulatory networks using piecewise-linear models*, Bull Math Biol **66** (2004), no. 2, 301–340.
31. H. de Jong and M. Page, *Qualitative simulation of large and complex genetic regulatory systems*, Proceedings of the 14th European Conference on Artificial Intelligence (W. Horn, ed.), 2000, pp. 141–145.
32. P. D'Haeseler, *Reconstructing gene networks from large scale gene expression data*, Ph.D. thesis, University of New Mexico, 2000.
33. M. Drton and M.D. Perlman, *Model selection for gaussian concentration graphs*, Biometrika **91** (2004), no. 3, 591–602.
34. R. Edwards and L. Glass, *Combinatorial explosion in model gene networks*, Chaos **10** (2000), no. 3, 691–704.
35. B. Ermentrout, *Simulating, analyzing and animating dynamical systems: A guide to xppaut for researchers and students*, 1 ed., Soc. for Industrial & Applied Math., 2002.
36. N. Friedman, M. Linial, I. Nachman, and D. Pe'er, *Using bayesian networks to analyze expression data*, J Comput Biol **7** (2000), no. 3-4, 601–620.
37. N. Friedman, K. Murphy, and S. Russell, *Learning the structure of dynamical probabilistic networks*, Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (San Francisco, CA, USA), Morgan Kaufmann Publishers, 1998, pp. 139–147.
38. J. Gebert and N. Radde, *Modelling procaryotic biochemical networks with differential equations*, AIP Conference Proceedings, vol. 839, 2006, pp. 526–533.
39. D.T. Gillespie, *Exact stochastic simulation of coupled chemical reactions*, J Phys Chem **81** (1977), no. 25, 2340–2361.
40. L. Glass and S.A. Kauffman, *The logical analysis of continuous, non-linear biochemical control networks*, J Theor Biol **39** (1973), 103–129.
41. J.L. Gouze, *Positive and negative circuits in dynamical systems*, J Biological Systems **6** (1998), no. 21, 11–15.
42. J. Guckenheimer and P. Holmes, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, Springer Series, New York, 1983.
43. M. Gustafsson, M. Hörnquist, and A. Lombardi, *Constructing and analyzing a large-scale gene-to-gene regulatory network - lasso-constrained inference and biological validation*, IEEE Transaction on Computational Biology and Bioinformatics **2** (2005), no. 3, 254–261.
44. J. Hasty, D. McMillen, F. Isaacs, and J.J. Collins, *Computational studies of gene regulatory networks: in numero molecular biology*, Nature Review Genetics **2** (2001), no. 4, 268–279.
45. D. Heckerman, *A tutorial on learning with bayesian networks*, Technical Report MSR-TR-95-06, Microsoft Research, Redmond, WA, USA, 1995.
46. D. Heckerman, D. Geiger, and D.M. Chickering, *Learning bayesian networks: The combination of knowledge and statistical data*, Machine Learning **20** (1995), 197–243.

47. D. Heckerman, A. Mamdani, and M. Wellman, *Real-world applications of bayesian networks*, Communications of the ACM **38** (1995), no. 3, 24–30.
48. S. Huang, *Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery*, Journal of Molecular Medicine **77** (1999), 469–480.
49. F. Jacob and J. Monod, *Genetic regulatory mechanisms in the synthesis of proteins*, J Mol Biol **3** (1961), 318–356.
50. L. Kaderali, *A hierarchical bayesian approach to regression and its application to predicting survival times in cancer*, Shaker Verlag, Aachen, 2006.
51. L. Kaderali, T. Zander, U. Faigle, J. Wolf, J.L. Schultze, and R. Schrader, *Caspar: A hierarchical bayesian approach to predict survival times in cancer from gene expression data*, Bioinformatics **22** (2006), no. 12, 1495–1502.
52. S. Kauffman, *Metabolic stability and epigenesis in randomly constructed genetic nets*, J Theor Biol **22** (1969), 437–467.
53. S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita, *Dynamic modeling of genetic networks using genetic algorithm and S-systems*, Bioinformatics **19** (2003), no. 5, 643–650.
54. H. Lähdesmäki, I. Shmulevich, and O. Yli-Harja, *On learning gene regulatory networks under the boolean network model*, Machine Learning **52** (2003), 147–167.
55. W. Lam and F. Bacchus, *Using causal information and local measures to learn bayesian networks*, Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence (Washington, DC, USA), Morgan Kaufmann Publishers, 1993, pp. 243–250.
56. R. Laubenbacher and B. Stigler, *A computational algebra approach to the reverse engineering of gene regulatory networks*, J Theor Biol **229** (2004), no. 4, 523–537.
57. F. Li, T. Long, Y. Lu, Q. Ouyangm, and C. Tang, *The yeast cell-cycle network is robustly designed*, Proc. Natl. Acad. Sci. U. S. A **101** (2004), 4781–4786.
58. S. Liang, S. Fuhrman, and R. Somogyi, *Reveal, a general reverse engineering algorithm for inference of genetic network architectures*, Pac Symp Biocomput **3** (1998), 18–29.
59. D. Madigan, J. Garvin, and A. Raftery, *Eliciting prior information to enhance the predictive performance of bayesian graphical models*, Communications in Statistics: Theory and Methods **24** (1995), 2271–2292.
60. J.M. Mahaffy, D.A. Jorgensen, and R.L. van der Heyden, *Oscillations in a model of repression with external control*, J Math Biol **30** (1992), 669–691.
61. J.M. Mahaffy and C.V. Pao, *Models of genetic control by repression with time delays and spatial effects*, J Math Biol **20** (1984), 39–57.
62. L. Mao and H. Resat, *Probabilistic representation of gene regulatory networks*, Bioinformatics **20** (2004), no. 14, 2258–2269.
63. A.A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. Dalla-Favera, and A. Califano, *Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context*, BMC Bioinformatics **7 (Suppl 1)** (2006), S7.
64. A.A. Margolin, K. Wang, W.K. Lim, M. Kustagi, I. Nemenman, and A. Califano, *Reverse engineering cellular networks*, Nature Protocols **1** (2006), 663–672.
65. H.H. McAdams and A. Arkin, *Stochastic mechanisms in gene expression*, Proc. Natl. Acad. Sci. U. S. A. **94** (1997), 814–819.
66. T. Mestl, E. Plahte, and S.W. Omholt, *A mathematical framework for describing and analyzing gene regulatory networks*, J Theor Biol **176** (1995), no. 2, 291–300.
67. K. Murphy and S. Mian, *Modelling gene expression data using dynamic bayesian networks*, Tech. report, Computer Science Division, University of California, Berkeley, CA, USA, 1999.

68. I. Nachman, A. Regev, and N. Friedman, *Inferring quantitative models of regulatory networks from expression data*, Bioinformatics **20** (2004), no. 1, i248–i256.
69. S. Ott, S. Imoto, and S. Miyano, *Finding optimal models for small gene networks*, Pac Symp Biocomput **9** (2004), 557–567.
70. J. Pearl, *Causality: Models, reasoning and inference*, Cambridge University Press, Cambridge, 2000.
71. J. Pearl and T. Verma, *A theory of inferred causation*, Knowledge Representation and Reasoning: Proceedings of the Second International Conference (New York) (J. Allen, R. Fikes, and E. Sandewal, eds.), Morgan Kaufmann Publishers, 1991, pp. 441–452.
72. D. Pe'er, *Bayesian network analysis of signaling networks: A primer*, Science STKE **281** (2005), p 14.
73. B.-E. Perrin, L. Ralaivola, A. Mazurie, et al., *Gene networks inference using dynamic bayesian networks*, Bioinformatics **19 Suppl. II** (2003), i138–i148.
74. N. Radde, J. Gebert, and C.V. Forst, *Systematic component selection for gene network refinement*, Bioinformatics **22** (2006), 2674–2680.
75. N. Radde and L. Kaderali, *Bayesian inference of gene regulatory networks using gene expression time series data*, BIRD 2007, LNBI **4414** (2007), 1–15.
76. R.W. Robinson, *Counting labeled acyclic graphs*, New Directions in the Theory of Graphs (F. Harary, ed.), Academic Press, New York, 1973, pp. 239–273.
77. C. Sabatti and G.M. James, *Bayesian sparse hidden components analysis for transcription regulation networks*, Bioinformatics **22** (2006), no. 6, 739–746.
78. M. Santillán and M.C. Mackey, *Dynamic regulation of the tryptophan operon: A modeling study and comparison with experimental data*, Proc. Natl. Acad. Sci. U. S. A. **98** (2001), no. 4, 1364–1369.
79. M.J. Schilstra and H. Bolouri, *Modelling the regulation of gene expression in genetic regulatory networks*, Document for NetBuilder, a graphical tool for building logical representations of genetic regulatory networks.
80. C.E. Shannon and W. Weaver, *The mathematical theory of communication*, University of Illinios Press, 1963.
81. I. Shmulevich, E.R. Dougherty, and W. Zhang, *From boolean to probabilistic boolean networks as models of genetic regulatory networks*, Proceedings of the IEEE **90** (2002), no. 11, 1778–1792.
82. I. Shmulevich, A. Saarinen, O. Yli-Harja, and J. Astola, *Inference of genetic regulatory networks under the best-fit extension paradigm*, Proceedings of the IEEE EURASIP Workshop on Nonlinear Signal and Image Proc. (W. Zhang and I. Shmulevich, eds.), 2001.
83. A. Silvescu and V. Honavar, *Temporal boolean network models of genetic networks and their inference from gene expression time series*, Complex Systems **13** (1997), no. 1, 54–75.
84. P.W.F. Smith and J. Whittaker, *Edge exclusion tests for graphical gaussian models*, Learning in Graphical Models (M. Jordan, ed.), MIT Press, 1999, pp. 555–574.
85. P. Smolen, D.A. Baxter and J.H. Byrne, *Modeling transcriptional control in gene networks*, Bull Math Biol **62** (2000), 247–292.
86. R.V. Solé, B. Luque, and S.A. Kauffman, *Phase transitions in random networks with multiple states*, Technical Report 00-02-011, Santa Fe Institute, 2000.
87. P.T. Spellman, G. Sherlock, M.Q. Zhang, et al., *Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization*, Mol Biol Cell **9** (1998), 3273–3297.
88. D. Spiegelhalter, A. Dawid, S. Lauritzen, and R. Cowell, *Bayesian analysis in expert systems*, Statistical Science **8** (1993), 219–282.

89. P. Sprites, C. Glymour, and R. Scheines, *Causation, prediction, and search*, Springer Verlag, New York, 1993.
90. H. Steck and T. Jaakkola, *On the dirichlet prior and bayesian regularization*, Advances in Neural Information Processing Systems 15 (Cambridge, MA, USA), MIT Press, 2002.
91. J. Suzuki, *A construction of bayesian networks from databases based on an mdl scheme*, Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence (Washington, DC, USA), Morgan Kaufmann Publishers, 1993, pp. 266–273.
92. D. Thieffry and R. Thomas, *Qualitative analysis of gene networks*, Pac Symp Biocomput **3** (1998), 77–88.
93. R. Thomas, *On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations*, Springer Series in Synergetics **9** (1981), 180–193.
94. R. Thomas and R. d'Ari, *Biological feedback*, CRC Press, Boca Raton, FL, USA, 1990.
95. R. Thomas, S. Mehrotra, E.T. Papoutsakis, and V. Hatzimanikatis, *A model-based optimization framework for the inference on gene regulatory networks from dna array data*, Bioinformatics **20** (2004), no. 17, 3221–3235.
96. R. Thomas, D. Thieffry, and M. Kauffman, *Dynamical behaviour of biological regulatory networks – I. biological role of feedback loops and practical use of the concept of the loop-characteristic state*, Bull Math Biol **57** (1995), 247–276.
97. M. Tomita, *Whole-cell simulation: A grand challenge for the 21st century*, Trends Biotechnol. **19** (2001), no. 6, 205–210.
98. E.P. van Someren, B.L.T. Vaes, W.T. Steegenga, A.M. Sijbers, K.J. Dechering, and J.T. Reinders, *Least absolute regression network analysis of the murine osteoblast differentiation network*, Bioinformatics **22** (2006), no. 4, 477–484.
99. E.P. van Someren, L.F.A. Wessels, and M.J.T. Reinders, *Linear modeling of genetic networks from experimental data*, ISMB 2000: Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, 2000, pp. 355–366.
100. E.O. Voit, *Computational analysis of biochemical systems*, Cambridge University Press, 2000.
101. E.O. Voit and J. Almeida, *Decoupling dynamical systems for pathway identification from metabolic profiles*, Bioinformatics **20** (2004), no. 11, 1670–1681.
102. J. von Neumann, *The theory of self-reproducing automata*, University of Illinois Press, 1966.
103. A. Wagner, *Circuit topology and the evolution of robustness in two-gene circadian oscillators*, Proc. Natl. Acad. Sci. U. S. A. **102** (2005), 11775–11780.
104. D.C. Weaver, *Modeling regulatory networks with weight matrices*, Pac Symp Biocomput, 1999, pp. 112–123.
105. G. Yagil and E. Yagil, *On the relation between effector concentration and the rate of induced enzyme synthesis*, Biophysical Journal **11** (1971), no. 1, 11–27.
106. M. Zou and S.D. Conzen, *A new dynamic bayesian network (dbn) approach for identifying gene regulatory networks from time course microarray data*, Bioinformatics **21** (2005), no. 1, 71–79.