A Discrete Approach to Top-Down Modeling of Biochemical Networks

Chapter 12

Reinhard Laubenbacher and Pedro Mendes
Virginia Bioinformatics Institute, Blacksburg, Virginia

ABSTRACT

Mathematical and statistical network modeling is an important step toward uncovering the organizational principles and dynamic behavior of biological networks. This chapter focuses on methods of constructing discrete dynamic models of biochemical networks from high-throughput experimental data sets, also sometimes referred to as top-down modeling or reverse-engineering. Time-discrete dynamical systems models have long been used in biology, particularly in population dynamics. The models we mainly focus on here are also assumed to have a finite set of possible states for each variable. That is, the modeling framework discussed in this chapter is that of time-discrete dynamical systems over a finite state set.

After a brief survey of Boolean network and multi-state models, we discuss a modeling method using tools from computer algebra and the theory of Groebner bases. The method provides a compact description of the entire space of possible models and chooses from that space a model that is minimal in the sense that it contains no components that vanish on the data set used to construct the model. We also discuss the requirements of a mathematical program for the identification of biological systems.

I. INTRODUCTION

"All processes in organisms, from the interaction of molecules to the complex functions of the brain and other whole organs, strictly obey these physical laws. Where organisms differ from inanimate matter is in the organization of their systems and

especially in the possession of coded information (Mayr 1988, p. 2)." It is the task of systems biology to elucidate those differences. This process has barely begun and many researchers are testing computational tools that have been used successfully in other fields for their efficacy in helping to understand many biological systems. Here we are concerned with cellular biochemical networks. Mathematical and statistical network modeling is an important step toward uncovering the organizational principles and dynamic behavior of such networks.

This chapter focuses on methods of constructing discrete dynamic models of biochemical networks from high-throughput experimental data sets, also sometimes referred to as top-down modeling or reverse-engineering. Time-discrete dynamical systems models have long been used in biology, particularly in population dynamics. The models we mainly focus on here are also assumed to have a finite set of possible states for each variable. Boolean networks are an example, using only two possible states for each variable. This assumption requires that all experimental measurements, which are real-valued, be first discretized into a finite number of classes.

Because we need to use time series of measurements to make dynamic models and might want to use heterogeneous data sets, great care must be taken during this step so as not to lose too much information. The resulting models will have a lower resolution than, say, ODE models. However, in exchange they are sometimes easier to analyze. We see an important role for discrete models to provide constraints on the structure and dynamics of higher-resolution continuous models. In the language of Ideker and Lauffenburger (2003), discrete models are more high-level than ODE and PDE models.

After a short survey of discrete finite-state modeling frameworks and methods, we present a detailed description of a multi-state modeling technology that has a strong mathematical underpinning, providing mathematical and computational tools for model selection and analysis. We then discuss the issue of linking discrete high-level models with continuous low-level ones. Finally, we exploit the analogy of top-down modeling to the process of system identification in engineering and applied mathematics to outline some steps in a modeling program for cellular pathways.

II. TOP-DOWN MODELING

Traditionally, models of molecular regulatory systems in cells have been created bottom-up, where the model is constructed piece-by-piece by adding new components and characterizing their interactions with other molecules in the model. This process requires that the molecular interactions have been well characterized, usually through quantitative numerical values for kinetic parameters. Note that the construction of such models is biased toward molecular components that have already been associated with the phenomenon. Still, modeling can be of great help

in this bottom-up process, by revealing whether the current knowledge about the system is able to replicate its *in vivo* behavior.

There are many good examples of this process. Teusink et al. (2000) have built a comprehensive model of yeast glycolysis based on detailed kinetics of 15 enzymes of carbohydrate catabolism. Arkin et al. (1998) studied stochastic switching between lysis and lysogeny in a model of lambda phage infection. In a landmark paper, Bray et al. (1993) studied the regulation of chemotactic swimming of *E. coli* cells, correlating the model to the phenotypes of dozens of mutants. For an example of bottom-up modeling of a problem involving spatial distributions of signaling molecules, we refer to a study of calcium waves in neuroblastoma cells by Fink et al. (2000).

Bottom-up modeling is essentially a process of synthesis by which models of isolated cellular components (enzymes, and so on) are merged to become part of a larger model. Note that without applying other steps models built bottom-up are mechanistic (i.e., represent one level of organization with all of the details of the level below). For example, the model of ethanol catabolism mentioned previously contains details of enzyme action of each of its 15 component enzymes.

This modeling approach is well suited to complement experimental approaches in biochemistry and molecular biology, in that models thus created can serve to validate the mechanisms determined *in vitro* by attempting to simulate the behaviors of intact cells. Although this approach has been dominant in cellular modeling, it does not scale very well to genome-wide studies because it requires that proteins be purified and studied in isolation. This is not a practical endeavor due to its large scale, but especially because a large number of proteins act on small molecules that are not available in purified form, as would be required for *in vitro* studies.

With the completion of the human genome sequence and the accumulation of other fully sequenced genomes, research is moving away from the molecular biology paradigm to an approach characterized by large-scale molecular profiling and *in vivo* experiments (or if not truly *in vivo* at least carried out with intact cells). Technologies such as transcript profiling with microarrays, protein profiling with 2-D gels and mass spectrometry, and metabolite profiling with chromatography and mass spectrometry produce measurements that are large-scale characterizations of the state of the biological material probed.

Other new large-scale technologies are also able to uncover groups of molecules that interact (bind), allowing inference of interaction networks. All of these experimental methods are data rich, and some people have recognized (Loomis and Sternberg 1995; Brenner 1997; Kell 2004) that modeling is necessary to transform these data into knowledge. A new modeling approach is needed to best suit large-scale profiling experiments. Such a top-down approach will start with little knowledge about the system, capturing at first only a coarse-grained image of the system with only a few variables. Then, through iterations of simulation and experiment, the number of variables in the model is increased. At each iteration, novel experiments will be suggested by simulations of the model, which when carried out will

provide data to improve the model further, leading to a higher resolution in terms of mechanisms.

Although the processes of bottom-up and top-down modeling are distinct, both have as an objective the identification of molecular mechanisms responsible for cell behavior. The main difference between the two is that the construction of top-down models is biased by the data of the large-scale profiles, whereas bottom-up models are biased by the pre-existing knowledge of particular molecules and mechanisms.

Note that although top-down modeling makes use of genome-wide profiling data it is conceptually very different from other genome-wide data analysis approaches. Top-down modeling needs data produced in experiments that lend themselves to the approach—most likely those designed with that purpose in mind. One should not expect that a random combination of arbitrary molecular snapshots would be of much use for the top-down modeling process. Sometimes they may serve some purpose (e.g., variable selection), but overall, top-down modeling requires perturbation experiments that are carried out with appropriate controls. In the face of modern experimental research methods, the development of an effective top-down modeling strategy is crucial. In addition, we believe that a combination of top-down and bottom-up approaches will eventually have to be used.

III. DISCRETE MODELING METHODS

A. Boolean networks

The most common approach to the modeling of biochemical regulatory networks is through systems of ordinary differential equations; that is, time-continuous dynamical systems. In 1969, S. Kauffman proposed to model regulatory networks as logical switching networks, described as Boolean networks (Kauffman 1969). Boolean network models have the advantage of being more intuitive than ODE models, and might be considered as a coarse-grained approximation of the "real" network. They differ from ODE models in that molecules are considered present or absent, rather than ranging over a continuum of values. There is increasing evidence that certain types of regulatory networks have key features that can indeed be represented well through Boolean models (Davidson 2002; Wang et al. 2002; Fischle et al. 2003). Kauffman's early work has generated a substantial literature on the subject (Raeymaekers 2002; Sabatti et al. 2002; Albert and Othmer 2003; Kauffman et al. 2004).

Top-down modeling methods using the Boolean framework have been proposed by Liang et al. (1998), Akutsu et al. (1999), and Akutsu et al. (2000). To include stochastic features of gene regulation, probabilistic Boolean networks have been introduced by Shmulevich et al. (2002). The issue of how the Boolean framework can deal with experimental and biological noise was also addressed by Akutsu et al. (2000).

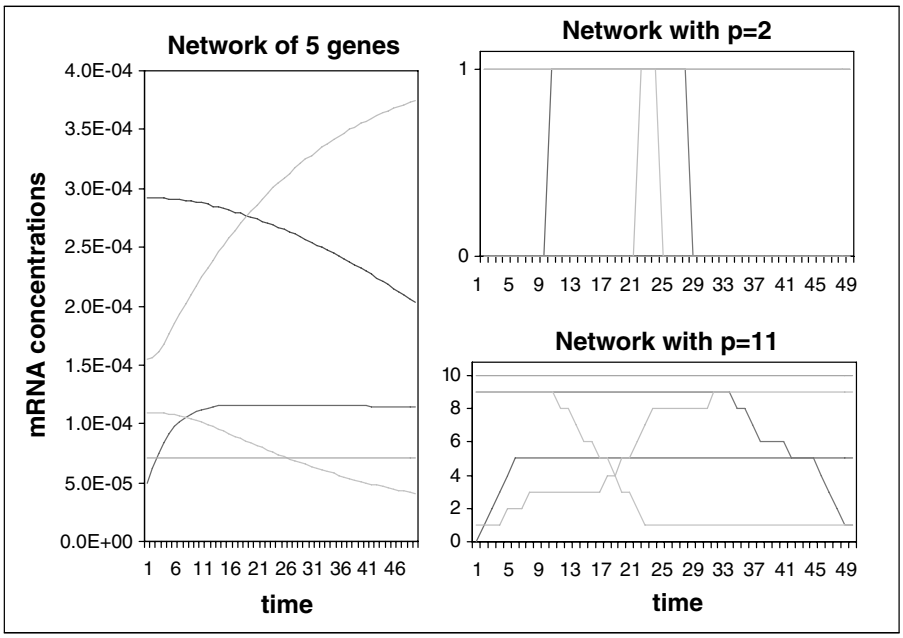


Figure 12.1. Different levels of data discretization.

B. Multi-state discrete models

One of the disadvantages of the Boolean modeling framework is the need to discretize real-valued expression data into an ON/OFF scheme, which loses a large amount of information. Figure 12.1 shows mRNA concentrations of a gene regulatory network simulated with the biochemical network simulator Gepasi (Mendes 1997) on the left. The right side of Figure 12.1 shows three different discretizations: one Boolean and the others allowing five (respectively 13) possible states.

This example makes it is clear that in many cases a finer data discretization is needed in order for a model to capture the essential dynamic features contained in a multivariate data set. Partly in response to this deficiency, multi-state discrete modeling frameworks and hybrid models have been developed. One of the most complex ones (Thomas 1991; Thieffry and Thomas 1998) uses multiple states for the genes in the network corresponding to certain thresholds of gene expression that make possible multiple gene actions. The model includes a mixture of multi-valued logical and real-valued variables, as well as the possibility of asynchronous updating of the variables. A top-down modeling method for this type of model was proposed by Thomas et al. (2004). A software package for analyzing this type of multi-state model is also available (de Jong et al. 2003).

Multiple discrete expression levels were also used in the reverse-engineering method of Repsilber et al. 2002), which uses a genetic algorithm to explore the

parameter space of multistage discrete genetic network models. Although this modeling framework is more effective than Boolean networks in capturing the many characteristics of gene regulatory networks, it also introduces substantially more computational complications from a top-down modeling point of view. A hybrid modeling framework was introduced by Brazma and Schlitt (2003) that tries to capture discrete as well as continuous aspects of gene regulation. The authors' finite-state linear model has a Boolean-network-type of control component, as well as linear functions that represent substances that change their concentrations continuously. For a more comprehensive review of modeling methods, see de Jong (2002).

C. Finite-state polynomial models

We now describe a multi-state discrete model approach that leverages existing algorithmic methods from symbolic computation and computational algebraic geometry (Laubenbacher and Stigler 2004). It models a regulatory network as a time-discrete multi-state dynamical system, synchronously updated. The method shares many features with a recently developed continuous top-down method (Yeung et al. 2002), which we first describe in some detail. According to the authors, the method is intended to generate a "first draft of the topology of the entire network, on which further, more local, analysis can be based." The authors make two assumptions. First, the system is assumed to be operating near a steady state, so that the dynamics can be approximated by a linear system of ordinary differential equations:

$$\frac{dx_i}{dt} = -\lambda_i x_i(t) + \sum_{j=1}^{N} w_{ij}(t) x_i(t) + b_i(t) + \xi_i(t),$$

For $i = 1, \ldots, N$. Here, $x_1, \ldots, x_N$ are mRNA concentrations, the λi are the self-degradation rates, the b_i are the external stimuli, and the ξI represent noise. The (unknown) w_{ij}, which are assumed to be constant over time, describe the type and strength of the influence of the jth gene on the ith gene. They assemble to a square matrix W of real numbers. The output of the reverse-engineering algorithm is this matrix W. The input is a series of data points obtained by applying the stimulus $(b_1, \ldots, b_N)^T$ and measuring the concentrations $x_1, \ldots, x_N{}^M$ times. Assembling these measurements into a matrix X, neglecting noise, and absorbing self-degradation into the coupling constants w_{ij}, we obtain a matrix equation

$$\frac{d}{dt}(X) = WX + B.$$

Here, X is an $(N \times M)$-matrix, W an $(N \times N)$-matrix, and B an $(N \times M)$-matrix. Using singular value decomposition (SVD), one obtains

$$X^T = UWV^T,$$

where U and V are orthogonal to each other. The first step is to obtain a particular solution W_0 to the reverse-engineering problem. One then obtains all possible solutions to the problem as

$$W = W_0 + CV^T,$$

where C ranges over the space of all square $(N \times N)$-matrices whose entries are equal to 0 for a certain range of j and arbitrary otherwise. Equivalently, CV^T ranges over all matrices that vanish on the given time points. The second assumption made in the paper is that gene regulatory networks are sparse. This provides a selection criterion on which to base a particular choice for C, and hence for W. The method selects the sparsest connection matrix W. This is accomplished through a particular choice of norm and robust regression. The algorithm was validated by way of simulated data from three networks.

The modeling framework for the discrete analog of this method is that of time-discrete dynamical systems over a finite state set X. Here, X is to be thought of as the set of discretized experimental values. For instance, in the Boolean case we have $X = \{0, 1\}$. To be precise, a dynamical system of dimension n over X is a function

$$f : X^n \to X^n$$

with dynamics generated by iteration of f. We will call f a *finite dynamical system*. Here, X^n denotes the set of all n-tuples with entries in X. Abbreviate an n-tuple $(x_1, \ldots, x_n)$ by $\mathbf{x}$. The function f is determined by its coordinate functions $f_i : X^n \to X$; that is,

$$f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})).$$

Suppose that we are given one or more time series of state transitions, measuring concentrations of mRNA, proteins, or metabolites. Our goal is to choose a finite dynamical system $f : X^n \to X^n$, which fits the data and "best describes" the network that generated the data. To be precise, we assume that we are given sequences of states

$$s_1 = (s_{11}, s_{21}, \ldots, s_{n1}), \ldots, s_m = (s_{1m}, \ldots, s_{nm})$$

$$t_1 = (t_{11}, t_{21}, \ldots, t_{n1}), \ldots, t_r = (t_{1r}, \ldots, t_{nr})$$

$$\ldots$$

These satisfy the property that if the unknown transition function of the network is f then

$$f(s_i) = s_{i+1}, \text{ for } i = 1, \ldots, m-1$$

$$f(t_j) = t_{j+1}, \text{ for } j = 1, \ldots, r-1$$

$$\ldots$$

Typically, there will be more than one possible choice. In fact, unless *all* state transitions of the system are specified there will always be more than one model that

fits the given data set. Because this much information is hardly ever available in practice, any top-down modeling method has to choose from a large set of possible models. As with most methods, ours will also choose the simplest model, in a certain sense. Before describing the selection principle used, we first need to describe the computational framework.

If we do not impose any further mathematical structure, we are left with a problem about set functions. No systematic computational tools for finding dynamical systems that fit the data (and for choosing a particular one) are available in this general setting. The standard mathematical solution is to endow the model space with a suitable additional mathematical structure. One way to do this is by a process analogous to the imposition of a coordinate system onto an affine space, resulting in an algebraic structure on the set of points in the space. Precisely, we assume that our set X is equipped with the structure of a finite number system; that is, a *finite field*.

It is well-known that this can be done whenever the number of elements in X is a power of a prime number p. This assumption is a straight-forward generalization of the Boolean case, where we can take advantage of Boolean arithmetic (e.g., $1 + 1 = 0$). Because the cardinality of X depends on the resolution of the discretization we choose, this is an easy assumption to satisfy in practice by refining the resolution, if needed. One possible approach is to choose a prime number p of possible variable states, in which case the number system can be taken to be $\mathbf{Z}/p$, the integers modulo p.

An important consequence of this assumption is the well-known fact (Lidl and Niederreiter 1997, p. 369) that each of the coordinate functions of f can be expressed as a polynomial function in n variables, with coefficients in X, and so that the degree of each variable is less than the number of elements in X. For instance, each Boolean function can be expressed as a polynomial, via the correspondence $x \wedge y = xy$, $x \vee y = x + y + xy$, and $\neg x = x + 1$. In other words, polynomial dynamical systems can serve as a computational model for all finite dynamical systems over a finite field. We are now in a position to use the rich algorithmic theory of polynomial algebra that has been developed over the last 20 years (Cox et al. 1997), including sophisticated symbolic computation software. Thus, we can overcome one disadvantage that discrete models have compared to ODE models, for which there is a mature mathematical theory available.

Thus, assume now that our state set X is a finite field. The model $f : X^n \to X^n$ we are searching for is determined by its coordinate functions $f_i : X^n \to X$. We can reverse engineer each coordinate function independently and thus reconstruct the system one variable at a time. The strategy of the method is to first compute the space of all systems that are consistent with the given time series data. The core of this computation is an interpolation algorithm. The method then chooses a particular system $f = (f_1, \ldots, f_n)$ that satisfies the following property.

Minimality: For each i, f_i is minimal in the sense that there is no non-zero polynomial g such that $f = h + g$ and g is identically equal to zero on the given time

points. That is, we exclude terms in the polynomials f_i that vanish identically on the data. In other words, we do not include interactions in the model that are not manifest in the given data set.

Suppose that f_i and f_i' are two models that fit the given data set. Then, $f_i(\mathbf{x}) = f_i'(\mathbf{x})$ for all data points $\mathbf{x}$. That is, $(f_i - f_i')(\mathbf{x}) = 0$ for all $\mathbf{x}$. Therefore, the set of all such models can be described as $f_i + I$, where f_i is a particular model and I is the set of all models that vanish identically on the given data set. In other words, the situation is very similar to the case of solving a nonhomogeneous system of linear equations, where f_i represents a particular solution to the system and I represents the solution space of the corresponding homogeneous system. The correspondence with the ODE modeling method described by Yeung et al. (2002) is that f_i corresponds to W_0 and I corresponds to the space C. Thus, we need to compute f_i and I.

The particular solution f_i can be computed using a standard formula for Lagrange interpolation (see Laubenbacher and Stigler (2004) for details). To compute Iwe use mathematical algorithms from computer algebra based on the theory of Groebner bases (Cox et al. 1997). What allows us to do this is the fact that the set of polynomials that vanish on a given data set has the algebraic structure of an *ideal* in the algebraic system $X[x_1, \ldots, x_n]$ of all polynomials in n variables with coefficients in X. These algorithms are implemented using the computer algebra system Macaulay2 (Grayson and Stillman, 2003). An important aspect of this computation is that the set of all possible models is described not by enumeration but in terms of a small set of generators, similar to describing a vector space by giving a basis for it. The algorithm to select the simplest model from the set $f_i + I$ uses another fundamental procedure in computer algebra: dividing a polynomial by all polynomials in the ideal I.

One can prove that there is in fact a unique simplest model to choose. However, the algorithm of Laubenbacher and Stigler (2004) depends on an up-front choice of a total ordering of the variables $x_1, \ldots, x_n$. This choice has the effect that the algorithm uses the "cheapest" (smallest, in this ordering) variables preferentially. On the one hand, this feature allows the incorporation of biological knowledge in the case where certain interactions are already known. On the other hand, it arbitrarily biases the model output in the case where such information is absent.

In the work of Laubenbacher and Stigler (2004), several variable orders were used and common terms in the polynomial models for each order were extracted to circumvent this problem. We briefly describe the validation of this approach. Albert and Othmer (2003) presented a Boolean model for a well-characterized network of segment polarity genes in *Drosophila melanogaster*. The network, consisting of five genes and their products, is responsible for pattern formation in the *Drosophila* embryo. The network is a ring of 12 interconnected cells, in which the genes are expressed in patterns resembling stripes. The genes represented in the Albert-Othmer model are *wingless*, *engrailed*, *hedgehog*, *patched*, and *cubitus interruptus*.

The proposed model is a collection of Boolean functions, representing the genes and proteins in the network. Each function governs the state transitions of a single compound. The following are four of the functions defined in the model.

$$f_6 = hh_i^{t+1} = EN^t{}_i \wedge \neg CIR^t{}_i$$

$$f_7 = HH_i^{t+1} = hh_i^t$$

$$f_8 = ptc_i^{t+1} = CIA_i^{t+1} \wedge \neg EN_i^{t+1} \wedge \neg CIR_i^{t+1}$$

$$f_9 = PTC_i^{t+1} = ptc_i^t \vee (PTC_i^t \wedge \neg HH_{i-1}^t \wedge \neg HH_{i+1}^t)$$

Representing each biochemical with a variable, the Boolean functions may be translated into polynomial functions, shown below.

$$f_6 = x_5(x_{15} + 1)$$

$$f_7 = x_6$$

$$f_8 = x_{13}((x_{11} + x_{20} + x_{11}x_{20}) + x_{21} + (x_{11} + x_{20} + x_{11}x_{20})x_{21}) \\ (x_4 + 1)(x_{13}(x_{11} + 1)(x_{20} + 1)(x_{21} + 1) + 1)$$

$$f_9 = x_8 + x_9(x_{18} + 1)(x_{19} + 1) + x_8x_9(x_{18} + 1)(x_{19} + 1)$$

Treating this Boolean model as "reality," wild-type and simulated knock-out experiments were generated, creating knock-outs by setting a function representing a gene equal to 0. As the algorithm relies on the choice of an ordering of the variables, causing some variables to have greater weight than the rest, four variable orders were used to counteract this preferential ranking.

Not surprisingly, algorithm performance improved greatly with knock-out data rather than just wild-type data. The algorithm is able to reconstruct approximately 84% of the interactions in the Boolean model, versus only 32% when only wild-type data were used. Furthermore, it correctly identified 92% of the additive interactions and 10% of the nonadditive interactions, whereas none of the nonadditive interactions were identified in the model constructed with only wild-type data.

A more elegant solution was proposed by Allen et al. (2005). Using a large number of randomly generated variable orders to generate models, the authors then rank the variables according to their frequency of appearance in the models for each of these variable orders. This ranking then determines a variable ordering to be used for the final model construction.

Another shortcoming of the algorithm of Laubenbacher and Stigler (2004) is that it relies on exact fitting of data. This makes the method very sensitive to noise that is known to be present in DNA microarray and other "-omics" data. To avoid models that are overly complex due to fitting of noise, the Laubenbacher group is presently developing a genetic algorithm that optimizes between data fit and model complexity. An important feature of the algorithm is that its performance is substantially improved by supplying as initialization the output of the exact data-fitting algorithm described previously versus a random initialization. The key theoretical ingredient in the algorithm is a mathematical characterization of the

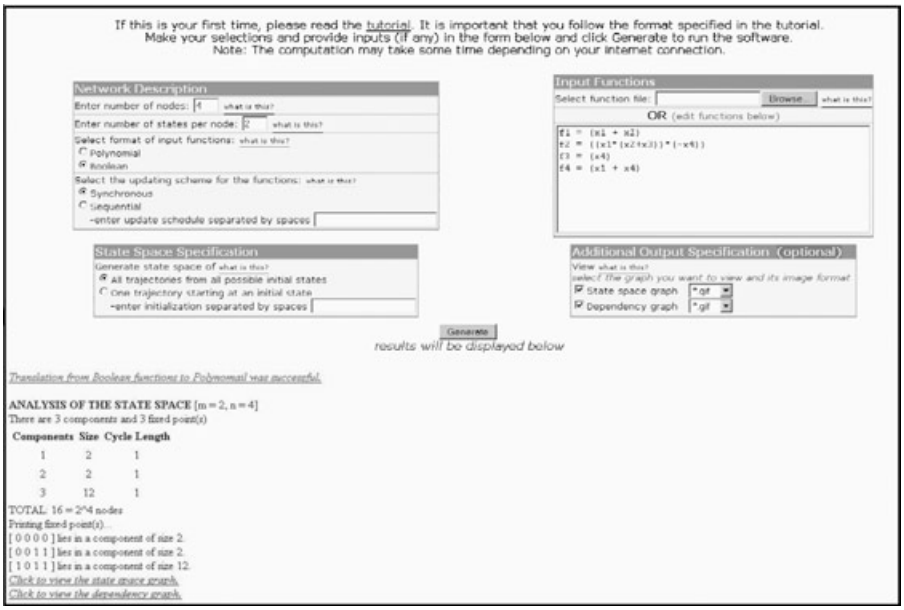


Figure 12.2. Snapshot of DVD interface.

evolution rules to guarantee that each mutation still satisfies the minimality criterion imposed.

An important tool for working with polynomial models over finite fields is the software package DVD (available at *http://dvd.vbi.vt.edu* as a web interface or for download). The program takes a polynomial system as input. For binary systems, one can also input Boolean functions, which are then translated into polynomial functions. DVD then computes the phase space of the system and outputs statistics such as the number of components, length of limit cycles, and so on. It also outputs the wiring diagram of the system. For small systems, it visualizes the phase space. Figure 12.2 shows the DVD interface.

IV. DATA DISCRETIZATION

The very important issue of data discretization has been studied from the points of view of Bayesian network applications and machine learning (Dougherty et al. 1995; Friedman and Goldszmidt 1996). The first important choice to make is the number of discrete states to use. The second choice is the method by which to map real-valued data to discrete states. There are various ways of labeling real-valued data using finite-state sets. Thresholds with biological relevance are one type of labeling that can be used. This is typically referred to as binning. For example, up-regulation, no regulation, and down-regulation of a gene may be used as thresh-

olds for partitioning the raw data into three groups, labeled 1, 0, and –1, respectively. For binary states, the choice of threshold is particularly crucial, in that even a relatively small change can result in very different discrete time series profiles (Sabatti et al. 2002). Another method of discretization is to normalize the expression of each gene or protein and use the deviation from the mean to discretize the data.

Any discretization method suitable for our purposes must preserve information about the dynamic relationship between the different variables, and must accommodate several heterogeneous time series simultaneously (e.g., transcription data as well as protein and metabolite concentrations). We have developed a method based on a graph theoretic approach that has the important advantage that the algorithm chooses an optimal number of states, based on the given data (Dimitrova et al. 2005). Most discretization methods require such a choice as part of the input. The algorithm has been implemented in C++ and is freely available. We illustrate it with an example.

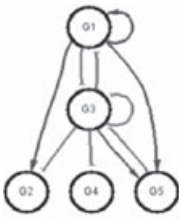
Consider the simulated gene regulatory network shown in Figure 12.3 (five genes, whose wiring diagram is given in Figure 3a). The network was generated with the artificial gene network system AGN (Mendes et al. 2003). After simulating the network with the biochemical network simulator Gepasi (Mendes 1997), one finds that it has the positive stable steady state (1.99006, 1.99006, 0.000024814, 0.997525, 1.99994). From the model, we generate six time series, each of length 20, including one wild-type time series and five deletion mutant time series. The discretization algorithm chooses the number system $X = \{0, 1, 2, 3, 4\}$, consisting of five different states for the combined data set.

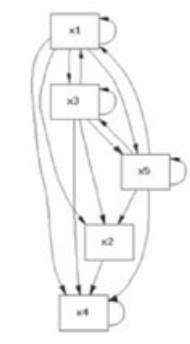
After using the multivariate interpolation algorithm, we obtain a "best" polynomial model $f : X^5 \to X^5$ in five variables. Its phase space consists of a directed graph whose nodes are the 5^5 possible states for the five variables, and there is a directed edge from state $\mathbf{a}$ to state $\mathbf{b}$ if $f(\mathbf{a}) = \mathbf{b}$. The model also has a fixed point, like the continuous "real-world" system. Figure 3c shows a particular initialization of the network, simulated in Gepasi, reaching the previously cited steady state. Figure 3d shows a sample of the time series obtained by initializing the discrete model f with the discretization of this initialization. It converges to the discretization (4, 4, 0, 4, 2) of the steady state cited previously.

This example illustrates the fact that the discrete model f exhibits the same qualitative dynamics as the continuous model we started with. Figure 3b shows the wiring diagram of the discrete model obtained with our algorithm. The main point of this example is to demonstrate that our discretization method preserves the essential dynamic features of the continuous system representing "reality" in this case, and our interpolation algorithm chooses a model that reflects these dynamic features as well as most of the causal dependencies among the variables.

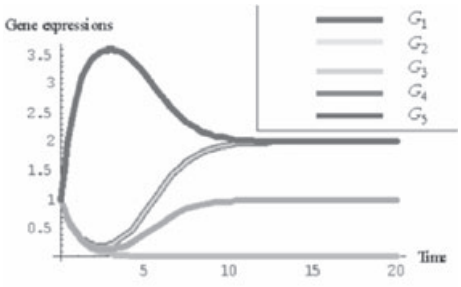
V. RELATIONSHIP BETWEEN DISCRETE AND CONTINUOUS MODELS

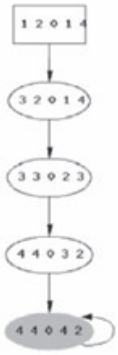
The relationship between discrete and continuous models has been studied extensively in population dynamics (Durrett and Levin 1994; Henson et al. 2001; Domokos





a. Wiring diagram of network. **b. Wiring diagram of model.**





c. Plot of time series of network. **d. State space of model.**

Figure 12.3. Graphs of a network and its associated models. (a) Wiring diagram of network, (b) wiring diagram of model, (c) plot of time series of network, and (d) state space of model.

and Scheuring 2004; Geritz and Kisdi 2004). For models of biochemical and other biological networks, this relationship was first explored by Glass and Kauffman (1973), with subsequent work by Edwards (2000), Edwards et al. (2001), and Glass et al. (2003). Within the modeling frameworks explored there, (bottom-up) discrete models can be a helpful tool to provide constraints and information about (bottom-up) continuous models of the same network. A good example of how a continuous and a discrete model of the same system can be used together is given by Muraille et al. (1996), where an ODE model of immune response to a replicating pathogen is studied via a discrete logical model using the technique of Thomas (1991). The dynamics of the discrete model, which are easy to analyze, are used to obtain a qualitative picture of the dynamics of the ODE model.

A corresponding mathematical theory for top-down modeling has yet to be developed. How can high-level information from discrete multi-state dynamic models of a network be incorporated into the model selection process for low-level ODE models? For the polynomial system framework described here, we are developing such a theory in parallel with an ODE framework based on a linearization of the dynamics (i.e., the Jacobian, a first-order truncation of the Taylor approximation to the dynamics).

Estimates of the elements of the Jacobian matrix are currently pursued through non-linear least squares. Our aim is to develop ways in which these top-down approaches become synergistic. In particular, we expect the results of the discrete model to be used as initial states for the parameter estimation needed to define a continuous model. We are currently carrying out experiments that will be used to validate both methods, using integrated transcriptomics, proteomics, and metabolomics time courses measuring oxidative stress response in *Sacchromyces cerevisiae*.

VI. A MATHEMATICAL THEORY FOR DISCRETE MODELS

Discrete models are not well understood at a theoretical level. In particular, the relationship between the structure of a model and its dynamics has remained elusive. There are no general results about the number of components of the state space of Boolean or multi-state discrete models or about the existence of steady states. Especially the question of steady states is an important one for biological models. Having fairly general results about the relationship between structure and dynamics for sufficiently large classes of models is an important problem.

Not surprisingly, these questions can be answered algorithmically for linear systems. Let X be a finite field and $f: X^n \to X^n$ a linear system. That is, the coordinate functions of f are linear polynomials without constant term. Then f can be represented by a matrix after making a choice of basis. It turns out that the structure of the phase space of f can be completely determined from the factorization of the characteristic polynomial of f, in particular the number of components and the length of all limit cycles (Hernandez Toledo 2003).

Very few results are available for nonlinear systems. A modest first step toward general results for sufficiently large classes of polynomial systems has been made by Colon-Reyes et al. (2004). Suppose that f is a Boolean polynomial system all of whose coordinate functions consist of monomials; that is, f is constructed using the *AND* operator. Let G be the directed graph whose vertices are the variables of f. There is a directed edge from x_i to x_j if x_j appears in f_i. Reversing the arrows of G, one obtains the wiring diagram of the network. One can define a positive integer, the *loop number* of G, which can be computed in polynomial time (relative to the number of vertices in G). The main result of Colon-Reyes et al. (2004) is that f has only steady states if and only if the loop number of all strongly connected components of G is equal to 1.

VII. TOWARD A MATHEMATICAL THEORY OF BIOLOGICAL SYSTEM IDENTIFICATION

The basic inverse problem we face in modeling biochemical networks is common in engineering and applied mathematics, known as system identification. Our goal is to make a phenomenological (and, ultimately, mechanistic) mathematical model of a multivariate system we can observe as well as perturb, and about which we may have partial knowledge. The major challenges, compared to typical engineered systems, are that the system is very often high-dimensional, the number of observations is small in comparison, and the information we have about the systems is very limited.

The basic procedure is to choose an appropriate modeling framework, use one or more time series of observations to identify some or all possible models within this framework, and choose "the best" one from the possible model space. For engineered systems there is a well-developed mathematical theory that helps in this process. (An important application is the development of controllers for systems.) In particular, there is a theory of system identifiability, which provides criteria for how good a given data set is for the system identification process (Ljung 1999) for a comprehensive treatment of system identification.

No corresponding mathematical theory exists yet for the identification of biological systems. In particular, there is no good understanding about the appropriate experimental design for a particular modeling framework that provides good data sets for top-down modeling. The most commonly studied type of systematic perturbation focuses on single genes in regulatory networks (Karp et al. 1999; Ideker et al. 2000; Rung et al. 2002; Shmulevich et al. 2002; Tegner et al. 2003). Genetic genomics provides another possible approach (Jansen 2003). Studies of the quantity of data needed have been done by Krupa (2002) and Selinger et al. (2003). The study of appropriate experimental designs for various modeling methods must be part of a long-term systems biology modeling program.

VIII. CONCLUSIONS

We have discussed some top-down modeling methods resulting in time-discrete dynamical system models over finite-state sets. They serve to provide high-level information about systems that can be used as constraints for the construction of low-level models, either top-down or bottom-up. Our method using polynomial dynamical systems over finite fields has the advantageous feature that its mathematical underpinning provides access to a variety of mathematical algorithms and symbolic computation software. In particular, it provides a mathematical basis for the investigation of questions such as "goodness" measures on data sets. Ultimately, the performance of top-down modeling methods cannot be properly evaluated unless we understand what types of input data are required for optimal performance. That is, the data must fit the models.

Experimental data sets suitable for the various modeling methods are still difficult to obtain, and the biochemical networks producing the data are typically too poorly understood to truly test modeling performance. An important resource in the field would be a collection of benchmark synthetic biochemical networks and the ability to generate from them data sets covering various types of networks, providing wild-type and perturbation time series. One possible tool for generating such networks and data is described by Mendes et al. (2003).

We believe that the field of system identification can serve as a blueprint for a mathematical top-down modeling program in systems biology. Based on a well-defined collection of model classes, from high-level statistical models down to ODE and PDE models, such a program must include the development of appropriate system identification methods for each model class and quality measures on data sets that can be used to develop confidence measures for the resulting models.

ACKNOWLEDGMENTS

This work was partially supported by NIH grant RO1 GM068947-01. The authors thank E. Dimitrova, A. Jarrah, D. Potter, B. Stigler, J. Tyson, and P. Vera-Licona for help in preparing this manuscript.

REFERENCES

Akutsu, T., Miyano, S., et al. (1999). Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pac. Symp. Biocomput.* 17–28.

Akutsu, T., Miyano, S., et al. (2000). Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function. *J. Comput. Biol.* **7**(3/4):331–343.

Akutsu, T., Miyano, S., et al. (2000). Algorithms for inferring qualitative models of biological networks. *Pac. Symp. Biocomput.* 293–304.

Akutsu, T., Miyano, S., et al. (2000). Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics* **16**(8):727–734.

Albert, R., and Othmer, H. G. (2003). The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster. *J. Theo. Biol.* **223**(1):1–18.

Allen, E. E., Fetrow, J. S., et al. **(2005a)**. Algebraic dependency models of protein signal transduction networks from time-series data. under review.

Allen, E. E., Fetrow, J. S., et al. **(2005b)**. Heuristics for dependency conjectures in proteomic signaling pathways. 43rd ACM Southeast Conference.

Arkin, A., Ross, J., et al. (1998). Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells. *Genetics* **149**(4):1633–1648.

Bray, D., Bourret, R. B., et al. (1993). Computer simulation of the phosphorylation cascade controlling bacterial chemotaxis. *Mol. Biol. Cell* **4**(5):469–482.

Brazma, A., and T. Schlitt (2003). Reverse engineering of gene regulatory networks: a finite state linear model. *Genome Biology* **4**(6).
Brenner, S. (1997). Loose Ends. London, Current Biology 73.
Colon-Reyes, O., Laubenbacher, R., et al. (2004). Boolean monomial dynamical systems. *Annals of Combinatorics* **8**:426–439.
Cox, D., Little, J., et al. (1997). *Ideals, Varieties, and Algorithms.* New York: Springer-Verlag.
Davidson, E. H., et al. (2002). A genomic regulatory network for development. *Science* **295**:1669–1678.
de Jong, H. (2002). Modeling and simulation of genetic regulatory systems: A literature review. *J. Comput. Biol.* **9**(1):67–103.
de Jong, H., Geiselmann, J., et al. (2003). Genetic Network Analyzer: Qualitative simulation of genetic regulatory networks. *Bioinformatics* **19**(3):336–344.
Dimitrova, E., McGee, J., et al. (2005). A graph-theoretic method for the discretization of gene expression measurements.
Domokos, G., and Scheuring, I. (2004). Discrete and continuous state population models in a noisy world. *J. Theo. Biol.* **227**:535–545.
Dougherty, J., Kohavi, R., et al. (1995). Supervised and unsupervised discretization of continuous features. In *Proceedings of the 12th International Conference on Machine Learning*. San Francisco: Morgan Kaufmann.
Durrett, R., and Levin, S. (1994). The importance of being discrete (and spatial). *Theo. Population Biol.* **46**:363–394.
Edwards, R. (2000). Analysis of continuous-time switching networks. *Physica* **146**:165–199.
Edwards, R., Siegelmann, H. T., et al. (2001). Symbolic dynamics and computation in model gene networks. *Chaos* **11**:160–169.
Fink, C. C., Slepchenko, B., et al. (2000). An image-based model of calcium waves in differentiated neuroblastoma cells. *Biophys. J.* **79**(1):163–183.
Fischle, W., Wang, Y., et al. (2003). Binary switches and modification cassettes in histone biology and beyond. *Nature* **425**:475–479.
Friedman, N., and Goldszmidt, M. (1996). Discretization of continuous attributes while learning Bayesian networks. In *Proceedings of the 13th International Conference on Machine Learning*. San Francisco: Morgan Kaufmann.
Geritz, S. A. H., and Kisdi, E. (2004). On the mechanistic underpinning of discrete-time population models with complex dynamics. *J. Theo. Biol.* **228**:261–269.
Glass, K., Xia, Y., et al. (2003). Interpreting time-series analyses for continuous-time biological models: Measles as a case study. *J. Theo. Biol.* **223**(1):19–25.
Glass, L., and Kauffman, S. A. (1973). The logical analysis of continuous, nonlinear biochemical control networks. *J. Theo. Biol.* **39**:103–129.
Grayson, D. R., and Stillman, M. E. (••–2003). Macaulay2.
Henson, S. M., Costantino, R. F., et al. (2001). Lattice effects observed in chaotic dynamics of experimental populations. *Science* **294**:602–605.
Hernandez, ••, and Toledo, R. A. (2003). Linear finite dynamical systems.
Ideker, T. E., and Lauffenburger, D. (2003). Building with a scaffold: Emerging strategies for high- to low-level cellular modeling. *Trends in Biotechnology* **21**(6):256–262.
Ideker, T. E., Thorsson, V., et al. (2000). Discovery of regulatory interaction through perturbation: Inference and experimental design. *Pac. Symp. Biocomput.* **5**:302–313.
Jansen, R. C. (2003). Studying complex biological systems using multifactorial perturbation. *Nature Reviews Genetics* **4**:145–151.

12 Karp, R. M., Stoughton, R., et al. (1999). Algorithms for choosing differential gene expression experiments. RECOMB99.

Kauffman, S. A. (1969). Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theo. Biol.* **22**(3):437–467.

Kauffman, S. A., Peterson, C., et al. (2004). Genetic networks with canalyzing Boolean rules are always stable. *PNAS* **101**(49):17102–17107.

Kell, D. B. (2004). Metabolomics and systems biology: Making sense of the soup. *Curr. Opin. Microbiol.* **7**(3):296–307.

Krupa, B. (2002). On the number of experiments required to find the causal structure of complex systems. *J. Theo. Biol.* **219**(2):257–267.

Laubenbacher, R., and Stigler, B. (2004). A computational algebra approach to the reverse engineering of gene regulatory networks. *J. Theo. Biol.* **229**:523–537.

Liang, S., Fuhrman, S., et al. (1998). Reveal, a general reverse engineering algorithm for inference of genetic network architectures. *Pac. Symp. Biocomput.* 18–29.

Lidl, R., and Niederreiter, H. (1997). *Finite Fields.* New York: Cambridge University Press.

Ljung, L. (1999). *System Identification: Theory for the User.* Upper Saddle River, NJ: Prentice-Hall.

Loomis, W. F., and Sternberg, P. W. (1995). Genetic networks. *Science* **269**(5224):649.

Mayr, E. (1988). Toward a New Philosophy of Biology. Cambridge, MA: Harvard University Press.

Mendes, P. (1997). Biochemistry by numbers: Simulation of biochemical pathways with Gepasi 3. *Trends Biochem. Sci.* **22**(9):361–363.

Mendes, P., Sha, W., et al. (2003). Artificial gene networks for objective comparison of analysis algorithms. *Bioinformatics* **19**(2):II122–II129.

Muraille, E., Thieffry, D., et al. (1996). Toxicity and neuroendocrine regulation of the immune response: A model analysis. *J. Theo. Biol.* **183**:285–305.

Raeymaekers (2002). Dynamics of Boolean networks controlled by biologically meaningful functions. *J. Theo. Biol.* **218**:331–341.

Repsilber, D., Liljenstrom, H., et al. (2002). Reverse engineering of regulatory networks: Simulation studies on a genetic algorithm approach for ranking hypotheses. *Biosystems* **66**(1/2):31–41.

Rung, J., Schlitt, T., et al. (2002). Building and analysing genome-wide gene disruption networks. *Bioinformatics* **18**(2):S202–S210.

Sabatti, C., Karsten, S. L., et al. (2002). Thresholding rules for recovering a sparse signal from microarray experiments. *Mathematical Biosciences* **176**:17–34.

Selinger, D. W., Wright, M. A., et al. (2003). On the complete determination of biological systems. *Trends Biotech.* **21**(6):251–254.

Shmulevich, I., Dougherty, E. R., et al. (2002a). Gene perturbation and intervention in probabilistic Boolean networks. *Bioinformatics* **18**(10):1319–1331.

Shmulevich, I., Dougherty, E. R., et al. (2002b). Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics* **18**(2):261–274.

Tegner, J., Yeung, M. K., et al. (2003). Reverse engineering gene networks: Integrating genetic perturbations with dynamical modeling. *Proc. Natl. Acad. Sci. USA* **100**(10):5944–5949.

Teusink, B., Passarge, J., et al. (2000). Can yeast glycolysis be understood in terms of in vitro kinetics of the constituent enzymes? Testing biochemistry. *Eur. J. Biochem.* **267**(17):5313–5329.

A

Thieffry, D., and Thomas, R. (1998). Qualitative analysis of gene networks. *Pac. Symp. Biocomput.* 77–88.

Thomas, R. (1991). Regulatory networks seen as asynchronous automata: A logical description. *J. Theo. Biol.* **153**:1–23.

Thomas, R., Mehrotra, S., et al. (2004). A model-based optimization framework for the inference on gene regulatory networks from DNA array data. *Bioinformatics* **20**(17):3221–3235.

Wang, W., Cherry, J. M., et al. (2002). A systematic approach to reconstructing transcription networks in Saccharomycescerevisiae. *Proc. Natl. Acad. Sci. USA* **99**(26):16893–16898.

Yeung, M. K., Tegner, J., et al. (2002). Reverse engineering gene networks using singular value decomposition and robust regression. *Proc. Natl. Acad. Sci. USA* **99**(9):6163–6168.

AUTHOR QUERY FORM

Dear Author:
During the preparation of your manuscript for publication, the questions listed below have arisen. Please attend to these matters and return this form with your proof.
Many thanks for your assistance.

Query References	Query	Remarks
1.	Au: 2002a or b?	
2	(Au: 2002a or b?)	
3.	Au: In *Proceedings of* . . . ? Need title of publication, publisher, editor, city, etc.	
4.	Au: Is this reference complete?	
5.	Au: Need facts of publication.	
6.	Au: Editor?	
7.	Au: San Francisco? Editor?	
8.	Au: Need facts of publication or web contact.	
9.	Au: Initials?	
10.	Au: What does preprint mean? In publication? Need facts of publication.	
11.	Au: Proceedings of? Editor? Publisher?	
12.	Au: Journal? Book publication? Need facts of publication.	