From Boolean to Probabilistic Boolean Networks as Models of Genetic Regulatory Networks

ILYA SHMULEVICH, MEMBER, IEEE, EDWARD R. DOUGHERTY, AND WEI ZHANG

Invited Paper

Mathematical and computational modeling of genetic regulatory networks promises to uncover the fundamental principles governing biological systems in an integrative and holistic manner. It also paves the way toward the development of systematic approaches for effective therapeutic intervention in disease. The central theme in this paper is the Boolean formalism as a building block for modeling complex, large-scale, and dynamical networks of genetic interactions. We discuss the goals of modeling genetic networks as well as the data requirements. The Boolean formalism is justified from several points of view. We then introduce Boolean networks and discuss their relationships to nonlinear digital filters. The role of Boolean networks in understanding cell differentiation and cellular functional states is discussed. The inference of Boolean networks from real gene expression data is considered from the viewpoints of computational learning theory and nonlinear signal processing, touching on computational complexity of learning and robustness. Then, a discussion of the need to handle uncertainty in a probabilistic framework is presented, leading to an introduction of probabilistic Boolean networks and their relationships to Markov chains. Methods for quantifying the influence of genes on other genes are presented. The general question of the potential effect of individual genes on the global dynamical network behavior is considered using stochastic perturbation analysis. This discussion then leads into the problem of target identification for therapeutic intervention via the development of several computational tools based on first-passage times in Markov chains. Examples from biology are presented throughout the paper.

Keywords*—*Attractor, best-fit extension, Boolean network, cell differentiation, coefficient of determination, consistency problem, gene, genetic network, influence, Markov chain, microarray, nonlinear filter, probabilistic Boolean network, root signal.

I. INTRODUCTION

A central focus of genomic research concerns understanding the manner in which cells execute and control the enormous number of operations required for normal function and the ways in which cellular systems fail in disease. Biological systems function in an exceedingly parallel and extraordinarily integrated fashion. Feedback and damping are routine even for the most common of activities. Thus, in this era of genomic biology, single gene perspectives are becoming increasingly limited for gaining insight into biological processes. Global, systemic, or network perspectives are becoming increasingly important for making progress in our understanding of the manner in which genes and molecules collectively form a biological system and for harnessing this understanding in educated intervention for correcting human diseases. Such approaches inevitably require computational and formal methods to process massive amounts of data, understand general principles governing the system under study, and make useful predictions about system behavior in the presence of known conditions.

The development of high-throughput genomic and proteomic technologies is empowering researchers in the collection of broad-scope gene information. The advent of cDNA microarrays and oligonucleotide chips [1]–[5], which facilitate large-scale surveys of gene expression, has incited much interdisciplinary scientific activity. The diagnostic potential of gene expression data has already been observed. For example, cancer classification using a variety of methods has been used to exploit the class-separating power of expression data: leukemias [6], various cancers [7], small, round, blue-cell cancers [8], and hereditary breast cancer [9]. The next step is to dig deeper and understand the underlying mechanisms and the functions of genes in health and disease.

One approach is to model the genetic regulatory system and infer the model structure and parameters from real gene expression data. There are two main objectives. First, we aim to discover and understand the underlying gene regulatory mechanisms by means of inferring them from data. This generally falls within the scope of computational learning theory [10] or system identification [11]. Second, by using the inferred model, we endeavor to make useful predictions by mathematical analysis and computer simulations. There

Manuscript received March 15, 2002; revised July 15, 2002.

I. Shmulevich and W. Zhang are with the Cancer Genomics Laboratory, University of Texas M.D. Anderson Cancer Center, Houston, TX 77030 USA (e-mail: is@ieee.org; wzhang@mdanderson.org).

E. R. Dougherty is with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: edward@ee.tamu.edu).

Digital Object Identifier 10.1109/JPROC.2002.804686

is a natural order to these two objectives in that the inference must precede the analysis and simulation. The potential clinical impact is tremendous as this type of model-based analysis not only can open up a window on the physiology of an organism and disease progression, but also translate into accurate diagnosis, target identification, drug development, and treatment.

An important and fundamental question is: what class of models should be chosen? The selection of a model class should be made in view of the data requirements and the goals of the modeling and analysis. Such a choice involves classical engineering tradeoffs. For instance, a "fine" model with many parameters may be able to capture detailed "low-level" phenomena, such as protein concentrations and kinetics of reactions, but will require very large amounts of data for the inference, lest the model be "overfit" to the data. At the same time, a "coarse" model with fewer parameters and lower complexity will succeed in capturing "high-level" phenomena, such as whether a gene is ON or OFF at a given time, but will require much smaller amounts of data. Such considerations should drive the selection of the model class. Needless to say, within a chosen model class, Occam's Razor Principle, which underlies all scientific theory building, dictates that the model complexity should never be made higher than what is necessary to faithfully "explain the data."

There is a rather wide spectrum of approaches for modeling gene regulatory networks, each with its own assumptions, data requirements, and goals. The gamut runs from linear models, Bayesian networks, neural networks, nonlinear ordinary differential equations, and stochastic models, to Boolean models, logical networks, Petri nets, graph-based models, grammars, and process algebras. There have been a number of excellent survey papers on modeling and simulation of genetic regulatory networks [12]–[14] as well as a recent book [15].

II. Why Boolean?

The model system that has received, perhaps, the most attention, not only from the biology community, but also in physics, is the *Boolean Network* model, originally introduced by Kauffman [16]–[19]. Good reviews of this model can be found in [20]–[22]. In this model, gene expression is quantized to only two levels: ON and OFF. The expression level (state) of each gene is functionally related to the expression states of some other genes, using logical rules. Computational models that reveal these logical interrelations have since then been successfully constructed [23]–[26]. Before we go into the formal definitions of the model, it may be useful to pause and ask several general, but fundamental, questions.

1) To what extent do such models represent reality?
2) Do we have the "right" type of data to infer these models?
3) What do we hope to learn from them?

The first question pertains more to modeling in general. All models only approximate reality by means of some formal representation. It is the degree to which we hope to approximate reality and, more importantly, our goals of modeling, namely, to acquire knowledge about some physical phenomenon, that determines what class of models should be chosen. Viewed in the framework of learning theory, we are simply selecting an appropriate *hypothesis space* and then proceed to actually select a hypothesis from this space by observing data. In the context of Boolean networks as models of genetic regulatory networks, there is no doubt that the binary approximation of gene expression is only, as Huang puts it [21], a "logical caricature." However, even though most biological phenomena manifest themselves in the continuous domain, we often describe them in a binary logical language such as "on and off," "upregulated and downregulated," and "responsive and nonresponsive." Before embarking on modeling gene regulatory networks with a Boolean formalism, it is prudent to test whether or not meaningful biological information can be extracted from gene expression data entirely in the binary domain. This question was taken up in [27]. We reasoned that if the genes, when quantized to only two levels (1 or 0), would not be informative in separating known subclasses of tumors, then there would be little hope for Boolean modeling of realistic genetic networks based on gene expression data. Fortunately, the results were very promising. By using binary gene expression data, generated via cDNA microarrays, and the Hamming distance as a similarity metric, we were able to show a clear separation between different subtypes of gliomas as well as between different sarcomas, using multidimensional scaling. This seems to suggest that a good deal of meaningful biological information, to the extent that it is contained in the measured continuous-domain gene expression data, is retained when it is binarized.

This leads to the second question. In the case of cDNA microarray data, it is widely recognized that reproducibility of measurements and between-slide variation is a major issue [28], [29]. Furthermore, genetic regulation exhibits considerable uncertainty on the biological level. Indeed, evidence suggests that this type of "noise" is in fact advantageous in some regulatory mechanisms [30]. Thus, from a practical standpoint, limited amounts of data and the noisy nature of the measurements can make useful quantitative inferences problematic and a coarse-scale qualitative modeling approach seems to be justified. To put it another way, if our goals of modeling were to capture the genetic interactions with fine-scale quantitative biochemical details in a global large-scale fashion, then the data produced by currently available technology would not be adequate.

Thus, the third question is concerned with what type of knowledge we hope to acquire with the chosen models and the available data. As a first step, we may be interested in discovering qualitative relationships underlying genetic regulation and control. That is, we wish to emphasize fundamental generic coarse-grained properties of large networks rather than quantitative details, such as kinetic parameters of individual reactions [21]. Furthermore, we may wish to gain insight into the dynamical behavior of such networks and how it relates to underlying biological phenomena, such as cellular state dynamics, thus providing the potential for the discovery of novel targets for drugs. Recent research indicates that many realistic biological questions may be an-

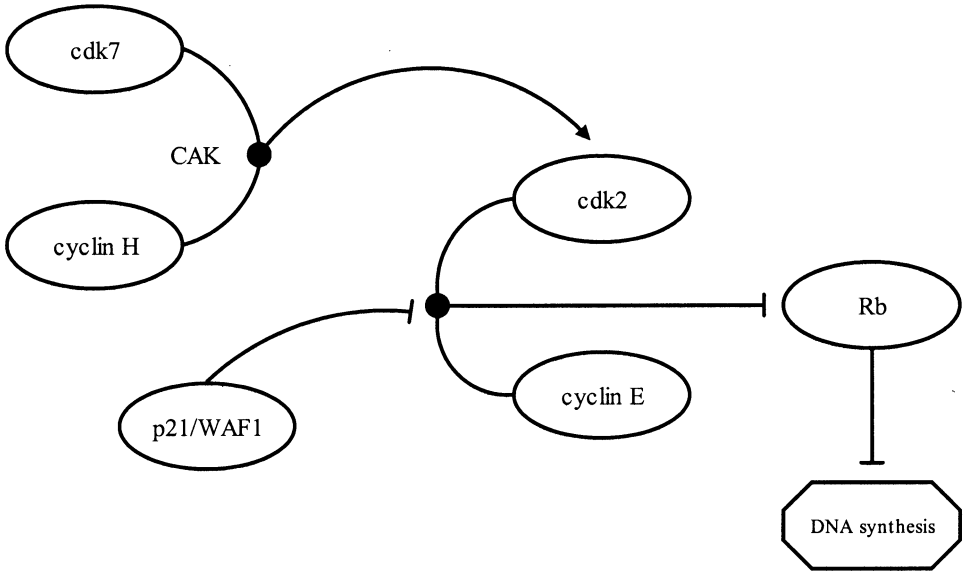


Fig. 1. A diagram illustrating the cell cycle regulation example. Arrowed lines represent activation and lines with bars at the end represent inhibition.

swered within the seemingly simplistic Boolean formalism. Boolean networks are structurally simple, yet dynamically complex, and have yielded insights into the overall behavior of large genetic networks [22], [31]–[33].

Let us give an example borrowed from [34], showing the logical representation of cell cycle regulation. This process of cellular growth and division is highly regulated. A disbalance in this process results in unregulated cell growth in diseases such as cancer. In order for cells to move from the G1 phase to the S phase, when the genetic material, DNA, is replicated for the daughter cells, a series of molecules such as cyclin E and cyclin-dependent kinase 2 (cdk2) work together to phosphorylate the retinoblastoma (Rb) protein and inactivate it, thus releasing cells into the S phase. Cdk2/cyclin E is regulated by two switches: the positive switch complex called cdk activating kinase (CAK) and the negative switch p21/WAF1. The CAK complex can be composed of two gene products: cyclin H and cdk7. When cyclin H and cdk7 are present, the complex can activate cdk2/cyclin E. A negative regulator of cdk2/cyclin E is p21/WAF1, which in turn can be activated by p53. When p21/WAF1 binds to cdk2/cyclin E, the kinase complex is turned off [35]. Further, p53 can inhibit cyclin H, a positive regulator of cyclin E/cdk2 [36]. This negative regulation is an important defensive system in the cells. For example, when cells are exposed to mutagen, DNA damage occurs. It is to the benefit of cells to repair the damage before DNA replication so that the damaged genetic materials do not pass onto the next generation. Extensive amount of work has demonstrated that DNA damage triggers switches that turn on p53, which then turns on p21/WAF1. p21/WAF1 then inhibits cdk2/cyclin E, thus Rb becomes activated and DNA synthesis stops. As an extra measure, p53 also inhibits cyclin H, thus turning off the switch that turns on cdk2/cyclin E. Such delicate genetic switch networks in the cells are the basis for cellular homeostasis—the ability of an organism to maintain equilibrium.

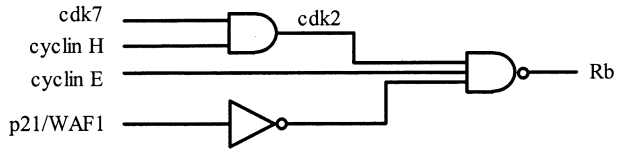


Fig. 2. The logic diagram describing the activity of Rb protein in terms of 4 inputs: cdk7, cyclin H, cyclin E, and p21. The gate with inputs cdk7 and cyclin H is an AND gate, the gate with input p21/WAF1 is a NOT gate, and the gate whose output is Rb is a NAND (negated AND) gate.

For the purposes of illustration, let us consider a simplified diagram, shown in Fig. 1, illustrating the effects of cdk7/cyclin H, cdk2/cyclin E, and p21/WAF1 on Rb. Thus, p53 and other known regulatory factors are not considered. While this diagram represents the above relationships from a pathway perspective, we may also wish to represent the activity of Rb in terms of the other variables in a logic-based fashion. Fig. 2 contains a logic circuit diagram of the activity of Rb ("on" or "off") as a Boolean function of four input variables: cdk7, cyclin H, cyclin E, and p21/WAF1. Note that cdk2 is shown to be completely determined by the values of cdk7 and cyclin H using the AND operation and thus, cdk2 is not an independent input variable. Also, in Fig. 1, p21/WAF1 is shown to have an inhibitive effect on the cdk2/cyclin E complex, which in turn regulates Rb, while in Fig. 2, we see that from a logic-based perspective, the value of p21/WAF1 works together with cdk2 and cyclin E to determine the value of Rb.

The representation containing logical gates in Fig. 2 should be familiar to most electrical engineers who have studied digital logic design.

III. Boolean Networks

We now describe the structure of Boolean networks and how their dynamics relate to functional cellular states. We also show some relationships to invariant signal sets

or root signals in nonlinear digital filtering. A Boolean network $G(V, F)$ is defined by a set of nodes (genes) $V = \{x_1, \ldots, x_n\}$ and a list of Boolean functions $F = (f_1, \ldots, f_n)$. Each $x_i \in \{0, 1\}$, $i = 1, \ldots, n$ is a binary variable and its value at time $t + 1$ is completely determined by the values of some other genes $x_{j_1(i)}, x_{j_2(i)}, \ldots, x_{j_{k_i}(i)}$ at time t by means of a Boolean function $f_i \in F$. That is, there are k_i genes assigned to gene x_i and the mapping j_k: $\{1, \ldots, n\} \rightarrow \{1, \ldots, n\}$, $k = 1, \ldots, k_i$ determines the "wiring" of gene x_i. Thus, we can write

$$x_i(t+1) = f_i\left(x_{j_1(i)}(t), x_{j_2(i)}(t), \ldots, x_{j_{k_i}(i)}(t)\right). \quad (1)$$

Each x_i represents the state (expression) of gene i, where $x_i = 1$ represents the fact that gene i is expressed and $x_i = 0$ means it is not expressed. The list of Boolean functions F represents the rules of regulatory interactions between genes. That is, any given gene transforms its inputs (regulatory factors that bind to it) into an output, which is the state or expression of the gene itself. The *maximum connectivity* of a Boolean network is defined by $K = \max_i k_i$. All genes are assumed to update synchronously in accordance with the functions assigned to them and this process is then repeated. The artificial synchrony simplifies computation while preserving the qualitative, generic properties of global network dynamics [21], [20], [32]. It is clear that the dynamics of the network are completely determined by (1). Let us give an example.

Consider a Boolean network consisting of five genes $\{x_1, \ldots, x_5\}$ with the corresponding Boolean functions given by the truth tables shown in Table 1. The maximum connectivity is $K = 3$, although we allow some input variables to duplicate, essentially reducing the connectivity. For example, consider f_4, which is the truth table of the well-known majority function. We see that, since $j_1(4) = 3$ and $j_2(4) = j_3(4) = 4$, $f_4(x_3, x_4, x_4) = x_4$, which is a function of only one (*essential*) variable.[1]

The dynamics of this Boolean network are shown in Fig. 3. Since there are five genes, there are $2^5 = 32$ possible states that the network can be in. Each state is represented by a circle and the arrows between states show the transitions of the network according to the functions in Table 1. It is easy to see that, because of the inherent deterministic directionality in Boolean networks as well as only a finite number of possible states, certain states will be revisited infinitely often if, depending on the initial starting state, the network happens to transition into them. Such states are called *attractors* and the states that lead into them comprise their *basins of attraction*. For example, in Fig. 3, the state (00000) is an attractor and the seven other (transient) states that eventually lead into it are its basin of attraction.

The attractors represent the *fixed points* of the dynamical system that capture its long-term behavior. The attractors are always cyclical and may consist of more than one state. The number of transitions needed to return to a given state in an attractor is called the *cycle length*. For example, the attractor (00000) has cycle length 1 while the states (11010) and (11110) comprise an attractor with cycle length 2.

[1]The majority of x_3, x_4, and x_4 is always x_4. Other variables are called *fictitious.*

Table 1
Truth Tables of the Functions in a Boolean Network With Five Genes

	f_1	f_2	f_3	f_4	f_5
	0	0	0	0	0
	1	1	1	0	0
	1	1	1	0	0
	1	0	0	1	0
	0	0	1	0	0
	1	1	1	1	0
	1	1	0	1	0
	1	1	1	1	1
j_1	5	3	3	3	5
j_2	2	5	1	4	4
j_3	4	4	5	4	1

The indices j_1, j_2, and j_3 indicate the input connections for each of the functions.

A. Relationship to Nonlinear Digital Filters

The attractors in Boolean networks are very closely related to so-called *root signals* of nonlinear digital filters. A root signal of a given filter is a signal that is invariant to applications of that filter; i.e., the signal remains unchanged. Root signals are important for characterizing nonlinear filters because they represent the "pass-band" characteristics of a filter, very much like the frequencies that are passed by a linear filter. Root signals have been studied extensively for different types of filters, such as median filters, stack filters, and morphological filters [37]–[42]. A key instance of root signals occurs in the case of idempotent filters, which play a central role in mathematical morphology [43], [44].

Consider a binary-valued one-dimensional (1-D) signal of arbitrary length. Suppose a window of length $n = 2m + 1$ is sliding across this signal. At every location of the window, the contents inside the window are used as input variables to some fixed Boolean function f. That is,

$$y_i = f(x_{i-m}, \ldots, x_i, \ldots, x_{i+m}) \quad (2)$$

represents the output of the Boolean function corresponding to the window centered on the ith value of the input signal. The sequence of outputs y_i can be thought of as an output signal of the filter. For example, if this Boolean function is monotone (positive), meaning that it can be written without complemented variables in its disjunctive normal form, then the filter defined by such a Boolean function is a stack filter [38]. It also corresponds to a neural network in which the weights of all the threshold logic gates are nonnegative. A special property of such filters, known as *threshold decomposition* [52], allows us to generalize these filters to the real-valued domain while being able to analyze all their deterministic and statistical properties entirely in the binary domain. It is easy to see that for a finite-length signal,[2] (2)

[2]For the case of finite-length signals, various appending strategies can be used to augment the left- and right-hand sides of the signal so that the output signal is of the same length as the input signal.

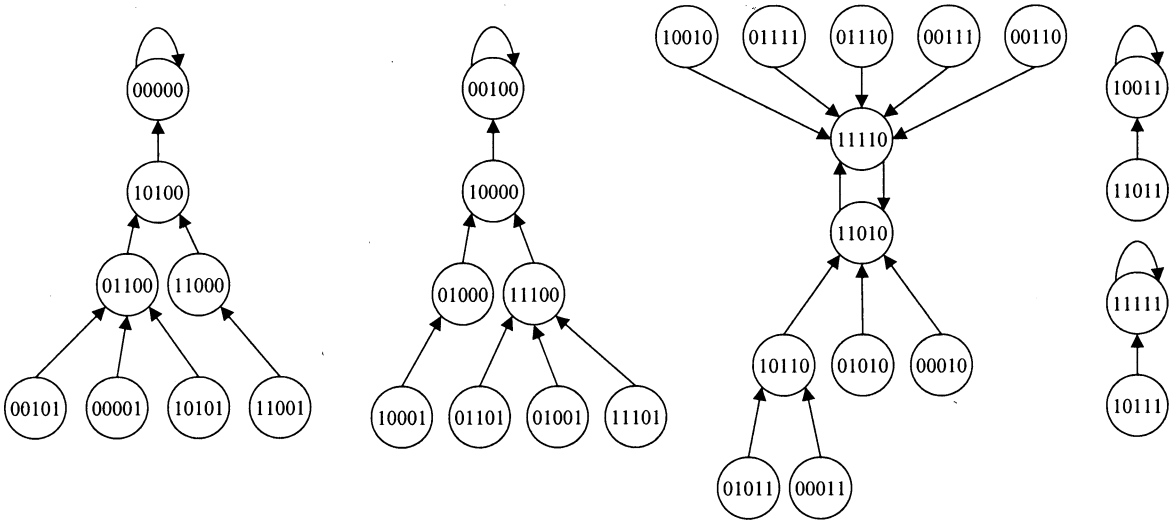


Fig. 3. The state-transition diagram for the Boolean network defined in Table 1.

is really a special case of (1): each Boolean function f_i is the same function f; $k_i = n$ for all i; and $j_1(i) = i - m$, $j_2(i) = i - m + 1, \ldots, j_n(i) = i + m$. So, the filter is really a Boolean network with simple fixed "wiring" defined by the neighborhood structure (window). Similarly, the entire signal corresponds to a state of a Boolean network and one filtering pass corresponds to a transition from that state to the next state. If the filtering process is repeated, the same cyclical phenomenon will occur. That is, either the signal will *converge* to a root signal after a finite number of filtering passes or periodic behavior will be observed.[3] Such a "sliding window" filtering process corresponds to a cellular automaton and can easily be extended to two- or higher-dimensional signals.

The attractors of a Boolean network represent a type of memory of the dynamical system. They also represent an abstract model of computation, which transforms a finite configuration (input) into another configuration (output). For example, the partitioning of the state space into attractors with their respective basins of attraction is a form of classification. Virtually the same idea appears in papers by Yu and Coyle [46], [47] on the classification and associative memory capability of stack filters, where the set of root signals represents the associative memory of the filter. Similar work showed that cellular automata, which are special cases of Boolean networks, can process information [48] and are able to perform computations, such as density classification [49], [50]. With the pioneering work of John von Neumann, biology was one of the first disciplines that considered using cellular automata for describing and simulating self-reproduction [51].

B. Cell Differentiation and Cellular Functional States

Boolean networks qualitatively reflect the nature of complex adaptive systems in that they are "systems composed of interacting agents described in terms of rules" [53]. A central concept in dynamical systems is that of *structural stability*, which is the persistent behavior of a system under perturbation. Structural stability formally captures the idea of behavior that is not destroyed by small changes to the system. This is most certainly a property of real genetic networks, since the cell must be able to maintain homeostasis in metabolism and its developmental program in the face of external perturbations and stimuli. Boolean networks naturally capture this phenomenon as the system usually "flows" back into the attractors when some of the genes are perturbed. Real gene regulatory networks exhibit spontaneous emergence of ordered collective behavior of gene activity. Moreover, recent findings provide experimental evidence for the existence of attractors in real regulatory networks [26]. At the same time, Wolf and Eeckman [54] showed that dynamical system behavior and stability of equilibria can be largely determined from regulatory element organization. This suggests that there must exist certain generic features of regulatory networks that are responsible for the inherent robustness and stability. In addition, since there are many different cell types in multicellular organisms, despite the fact that each cell contains exactly the same DNA content, the cellular "fate" is determined by which genes are expressed.

This was the insight pursued by Kauffman in his pioneering studies of genetic regulatory networks [16], [17], [20]. The idea was to generate random Boolean networks with certain properties and then systematically study the effects of these properties on the global dynamical behavior of the networks. For example, random Boolean networks were studied with varying average connectivity and different classes of Boolean functions, such as *canalizing* or *forcing* functions.[4] "Random" here means that the wiring is random as are the Boolean functions themselves. Kauffman's intuition was that the attractors in the Boolean networks should correspond to cellular types. This interpretation is quite reasonable if cell types are characterized by stable recurrent patterns of gene expression.

[3]It is interesting to note that similar periodic behavior exists even for some infinite networks (networks with an infinite number of nodes) [45], such as those in which every Boolean function is the majority function.

[4]A Boolean function $f\colon \{0, 1\}^n \to \{0, 1\}$ is called *canalizing* in its ith input if there exist y and z such that for all $(x_1, x_2, \ldots, x_n) \in \{0, 1\}^n$ with $x_i = y$, $f(x_1, x_2, \ldots, x_n) = z$.

It is widely believed that complex and adaptable systems such as the genome operate on the "edge of chaos." In the *ordered* regime, attractors are quite short, few, and stable, the latter mostly due to the fact that few short attractors imply large basins of attraction. In addition, the small cycle lengths imply the existence of large *frozen* components, which are sets of genes that do not change value as the network progresses through time. There are only isolated islands of genes that change values, but these cannot "communicate" or transfer information to each other because of the large frozen components. Thus, the network is highly resistant to perturbations (changes in the value of a gene) as well as wiring mutations (changes to the Boolean functions). It is unlikely that living systems operate in the ordered regime because evolution demands that there be some sensitivity to perturbations and mutations.

On the other hand, in the *chaotic* regime, the cycle length of attractors grows exponentially as a function of the number of genes and a perturbation of a gene propagates to many other genes, in an avalanche-like manner. Unlike in the ordered regime, there are few small islands of frozen genes with a large proportion of genes exhibiting variation. Thus, networks in the chaotic regime are very sensitive to initial conditions and perturbations, implying that organisms cannot be in the chaotic regime either.

The boundary between order and chaos is called the *complex* regime or the *critical phase*, as the transition from order to chaos is a phase transition. In this regime, the number of attractors and the cycle lengths are proportional to powers of n, where n is the total number of genes. As Stuart Kauffman puts it [56], "a living system must first strike an internal compromise between malleability and stability. To survive in a variable environment, it must be stable to be sure, but not so stable that it remains forever static." The complex regime can be elicited by "tuning" the parameters of a Boolean network, such as the connectivity K, the proportion of functions belonging to certain classes, such as canalizing functions, and the "bias" of the Boolean functions, which is the probability that the function outputs a 1.

Computer simulations have shown that for networks with low connectivity ($K = 2$) in which every function is *unbiased* or *balanced*, that is, when the probability p that it takes the value 1 is 0.5, the number of attractors is approximately $\sqrt{n}$. As the current estimate for the number of genes in the human genome is almost 40 000, this would imply that there are roughly 200 different cellular types. It is known that adult humans have about 254 cell types [55]. It was also shown (see [20]) that the expected cycle length for a $K = 2$ Boolean network is also on the order of $\sqrt{n}$. The cyclical nature of the attractors can be equated to the mitotic cycle in cells. It is also known that the cycle period or the cell doubling time—the time necessary for a cell to reproduce—is proportional to the cell's DNA content [20]. Although higher values of K result in more chaotic networks, the other parameters, such as p and the proportion of canalizing functions, can be tuned such that the network remains in the critical phase.

In the critical phase, the unfrozen components (genes that are changing over time) break up into isolated islands, separated by frozen components. Thus, there are many genes that are unfrozen and many that are frozen. As the network goes around the attractor cycle, representing the cell cycle, those genes that are unfrozen are presumably the "cell cycle genes" that are responsible for cell cycle regulation. For example, recent studies with microarray technology have revealed a comprehensive catalog of yeast genes whose transcript levels vary periodically within the cell cycle [57].

Another interpretation of the attractors in Boolean networks is that they represent cellular states, such as *proliferation* (cell cycle), *apoptosis* (programmed cell death), and *differentiation* (execution of tissue-specific tasks). This highly appealing view was expounded in [21] and [58] with ample biological justification. Such an interpretation can provide new insights into cellular homeostasis and cancer progression, the latter being characterized by a disbalance between these cellular states. For instance, if a (structural) mutation occurs, resulting in a very small probability of the network entering the apoptosis attractor(s), then the cells will be unable to undergo apoptosis and will exhibit uncontrolled growth. Similarly, large basins of attraction for the proliferation attractor would result in hyperproliferation, typical of tumorigenesis.

Such an interpretation need not be at odds with the interpretation that attractors represent cellular types. To the contrary, these views are complementary to each other, since, for a given cell type, different cellular functional states must exist and be determined by collective behavior of gene activity. Thus, one cell type can comprise several "neighboring" attractors each of which corresponds to different cellular functional states [59].

C. Inference: Lessons From Computational Learning Theory and Nonlinear Signal Processing

Although studying generic properties of large Boolean networks is quite important for gaining insight into the dynamical behavior and organization of real genetic networks, in order to make progress in understanding the genetic regulation in specific organisms and develop tools for rational therapeutic intervention in diseases such as cancer, it is necessary to be able to identify the networks from real experimental data. Much recent work on Boolean networks has focused on identifying the network structure from gene expression data [60]–[68]. At the same time, a large body of related work in computational learning theory [10], [69] has addressed very similar problems, namely, learning or inferring Boolean functions from examples of their input–output behavior.[5] A major focus in this field has been on the construction of algorithms for the efficient determination of Boolean formulas from examples. This type of induction of Boolean logic or the design of Boolean classifiers forms the core of many data-mining and knowledge-discovery algorithms [69], [70].

For example, the well-known *consistency problem* represents a search for a rule from examples [71]–[73]. That

[5]In a Boolean network, the input and output corresponds to time t and $t + 1$.

is, given some sets T and F of "true" and "false" vectors, respectively, the aim is to discover a Boolean function f that takes on the value 1 for all vectors in T and the value 0 for all vectors in F. It is also commonly assumed that the target function f is chosen from some class of possible target functions. In the context of Boolean networks, such a class could be the class of canalizing functions or functions with a limited number of essential variables. Formally, let $T(f) = \{v \in \{0, 1\}^n : f(v) = 1\}$ be called the *on-set* of function f and let $F(f) = \{v \in \{0, 1\}^n : f(v) = 0\}$ be the *off-set* of f. The sets $T, F \subseteq \{0, 1\}^n, T \cap F = \emptyset$, define a *partially defined* Boolean function $g_{T,F}$ as

$$g_{T,F}(v) = \begin{cases} 1, & v \in T \\ 0, & v \in F \\ *, & \text{otherwise.} \end{cases}$$

A function f is called an *extension* of $g_{T,F}$ if $T \subseteq T(f)$ and $F \subseteq F(f)$. The consistency problem (also called the extension problem) can be posed as: given a class C of functions and two sets T and F, is there an extension $f \in C$ of $g_{T,F}$?

In reality, a consistent extension may not exist either due to errors, or more likely, due to a number of underlying latent factors. This is no doubt the case for gene expression profiles as measured from microarrays. In this case, we may have to give up our goal of establishing a consistent extension and settle for a Boolean formula that minimizes the number of misclassifications. This problem is known as the *best-fit extension problem* [72] and is formulated as follows. Suppose we are given positive weights $w(x)$ for all vectors $x \in T \cup F$ and define $w(S) = \sum_{x \in S} w(x)$ for a subset $S \subseteq T \cup F$. Then, the *error size* of function f is defined as

$$\varepsilon(f) = w(T \cap F(f)) + w(F \cap T(f)). \tag{3}$$

If $w(x) = 1$ for all $x \in T \cup F$, then the error size is just the number of misclassifications. The goal is then to output subsets T^* and F^* such that $T^* \cap F^* = \emptyset$ and $T^* \cup F^* = T \cup F$ for which the partially defined Boolean function g_{T^*,F^*} has an extension in some class of functions C and so that $w(T^* \cap F) + w(F^* \cap T)$ is minimum. Consequently, any extension $f \in C$ of g_{T^*,F^*} has minimum error size. It is clear that the best-fit extension problem is computationally more difficult than the consistency problem, since the latter is a special case of the former, that is, when $\varepsilon(f) = 0$. In [68], it was shown that for many function classes, including the class of all Boolean functions, the best-fit extension problem for Boolean networks is polynomial-time solvable.

While the focus in computational learning theory has mostly been on the complexity of learning, very similar types of problems have been studied in nonlinear signal processing, specifically, in optimal filter design [74]–[80]. This typically involves designing an estimator from some predefined class of estimators that minimizes the error of estimation among all estimators in the class. An important role in filter design is played by these predefined classes or constraints. For example, stack filters are represented by the class of monotone Boolean functions. Although it would seem that imposing such constraints can only result in a degradation of the performance (larger error) relative to the optimal filter with no imposed constraints, constraining may have certain advantages. These include prior knowledge of the degradation process (or in the case of gene regulatory networks, knowledge of the likely class of functions, such as canalizing functions), tractability of the filter design, and precision of the estimation procedure by which the optimal filter is estimated from observations. For example, we often know that a certain class of filters will provide a very good sub-optimal filter, while lessening the data requirements for its estimation.

Even in the context of limited data, there are modest approaches that can be taken. One general statistical approach is to discover associations between variables via the coefficient of determination (COD). The COD was introduced in the context of optimal nonlinear filter design [81], but since then has been used for inferring multivariate relationships between genes [82], [83]. Such relationships, referred to as *predictors*, are the basic building blocks of a rule-based network. In the binary case, a predictor is just a Boolean function. The COD measures the degree to which the expression levels of an observed gene set can be used to improve the prediction of the expression of a target gene relative to the best possible prediction in the absence of observations. The method allows incorporation of knowledge of other conditions relevant to the prediction, such as the application of particular stimuli, or the presence of inactivating gene mutations, as predictive elements affecting the expression level of a given gene. Using the COD, one can find sets of genes related multivariately to a given target gene.

Let us briefly discuss the COD in the context of Boolean networks. Let x_i be a *target* gene that we wish to predict by observing some other genes $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$. Also, suppose $f(x_{i_1}, x_{i_2}, \ldots, x_{i_k})$ is an optimal predictor of x_i relative to some error measure ε. For example, in the case of mean-square error (MSE) estimation, it is well known that the optimal predictor is the conditional expectation of x_i given $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ [84]. Let ε_{opt} be the optimal error achieved by f. Then, the COD for x_i relative to $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ is defined as

$$\theta = \frac{\varepsilon_i - \varepsilon_{\text{opt}}}{\varepsilon_i} \tag{4}$$

where ε_i is the error of the best (constant) estimate of x_i in the absence of any conditional variables. It is easily seen that the COD must be between 0 and 1 and measures the relative decrease in error from estimating x_i via f rather than by just the best constant estimate. In practice, the COD must be estimated from training data with designed approximations being used in place of f. Those sets of (predictive) genes that yield the highest COD, compared to all other sets of genes, are the ones used to construct the optimal predictor of the target gene. Given limited amounts of training data, it is prudent to constrain the complexity of the predictor by limiting the number of possible predictive genes that can be used. This corresponds to limiting the connectivity K of the Boolean network. Although, as discussed above, high values of K are biologically implausible (with no additional assumptions about the class of Boolean functions), a natural way to incorporate more predictive genes, while maintaining simple predictor design in the face of limited training data, will be

discussed below. Finally, the above procedure is applied to all target genes, thus estimating all the functions in a Boolean network. The method is computationally intensive and massively parallel architectures have been employed to handle large gene sets [85].

IV. Why Probabilistic?

As discussed above, there are really two ways to gain insight about biological systems from Boolean network modeling. The first way is to construct random Boolean networks and study their general behavior, especially as it relates to "local" behavior, such as connectivity of genes and classes of Boolean functions. Such an approach can yield useful knowledge about the generic properties of gene regulatory networks and how these relate to cellular types, cell cycle regulation, and other functional cellular states. The second way is to explicitly infer the specific structure of a Boolean network from actual gene expression data. This approach has the potential to reveal useful information about the living system under study, how it fails in disease, and how to rationally design therapeutic intervention. Let us pursue this approach further.

In a Boolean network, each (target) gene is "predicted" by several other genes by means of a Boolean function (predictor). Thus, after having inferred such a function from gene expression data, it could be concluded that if we observe the values of the predictive genes, we know, with full certainty, the value of the target gene. Conceptually, such an inherent determinism seems problematic as it assumes an environment with no uncertainty. However, the data used for the inference exhibits uncertainty on several levels. First, there is biological uncertainty: gene expression is inherently stochastic, not in the sense that it is totally random, but that it has a stochastic nature on account of intrinsic biological variability. Second, there is experimental noise due to the complex measurement process, ranging from hybridization conditions to microarray image processing techniques [86]. Third, there may be interacting latent variables, such as proteins, various environmental conditions, or other genes that we simply do not measure, that further add to the variability of the measurements. Thus, we are in a position of having to infer a (deterministic) predictor under uncertainty.

Although reasoning under uncertainty is not a new problem and has been extensively studied in the artificial intelligence and pattern recognition communities [87], [88], it nonetheless presents a problem when the uncertainty cannot be reliably estimated. Without doing so, we cannot know how well the designed predictor generalizes over the population. In other words, we cannot know whether the predictor that is designed on the sample data will still be able to reliably make predictions when presented with future examples. One natural approach to remedy "overfitting" is to penalize the complexity of the predictor, as simpler explanations are expected to generalize better than complex explanations in an inductive reasoning framework. Such an approach was taken in [89], [90] for Boolean prediction of gene expression, using the well-known MDL principle as well as normalized maximum likelihood.

Another approach, proposed in [34], is to "absorb" the uncertainty into the predictor. The reasoning goes as follows. Although we cannot reliably estimate the uncertainty in the data, primarily because we typically have only a limited number of samples (examples) relative to the number of genes, we can try to infer a number of simple predictors, each of which performs relatively well in terms of predicting the target gene. By simple we mean functions that have only a few input variables. Having produced a number of simple but decent predictors, it is then necessary to *synthesize* them together so that each gets a chance to contribute its own modest prediction. Such an approach is similar, at least in spirit, to multiresolution modeling or splines, where rather than fitting one overly complex model to the data, one fits many simple models and uses them in a concerted manner.

As described in Section III-C, the COD can be used to produce a number of good predictors simply by choosing those sets of (predictive) genes along with the corresponding optimal predictors having the highest CODs. Since the COD itself is estimated from the data, we have little reason to put all our faith into just one possibly good predictor. Thus, the approach is to "probabilistically synthesize" the good predictors such that each predictor's contribution is proportional to its determinative potential, as measured by the COD. This idea leads to probabilistic Boolean networks (PBNs).

V. PBNs

A number of additional justifications for introducing PBNs are contained in [34]. Here, we briefly give their definition and state several results and possible applications. As stated above, the basic idea is to combine several promising predictors or Boolean functions together, so that each can make a contribution to the prediction of a target gene. A natural approach is to allow a random selection of the predictors for a given target gene, with the selection probability being proportional to the COD of each predictor. That is, given genes $V = \{x_1, \ldots, x_n\}$, we assign to each x_i a set $F_i = \{f_1^{(i)}, \ldots, f_{l(i)}^{(i)}\}$ of Boolean functions representing the "top" predictors for that target gene. Clearly, if $l(i) = 1$ for all $i = 1, \ldots, n$, then the PBN simply reduces to a standard Boolean network. The basic building block of a PBN is shown in Fig. 4.

Conceptually, the probabilistic predictor of each target gene can be thought of as a random switch, where at each point in time or step of the network, the function $f_j^{(i)}$ is chosen with probability $c_j^{(i)}$ to predict gene x_i. As discussed above, one way to assign these probabilities is to use the COD, normalized such that $\sum_{j=1}^{l(i)} c_j^{(i)} = 1$. That is,

$$c_j^{(i)} = \frac{\theta_j^i}{\sum_{k=1}^{l(i)} \theta_k^i}$$

where θ_j^i is the COD for gene x_i relative to the genes used as inputs to predictor $f_j^{(i)}$.

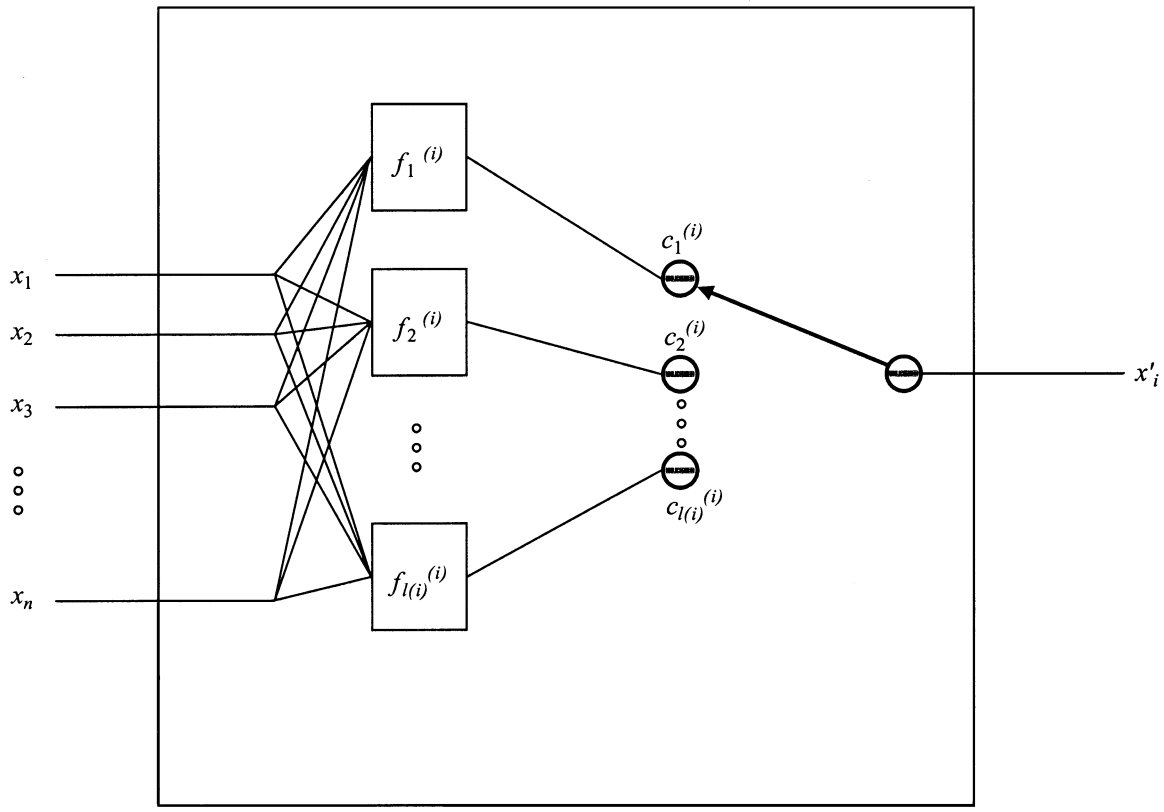


Fig. 4. A basic building block of a PBN. Although the "wiring" of the inputs to each function is shown to be quite general, in practice, each function (predictor) has only a few input variables.

Now consider the network as a whole. A *realization* of the PBN at a given instant of time is determined by a vector of Boolean functions, where the ith element of that vector contains the predictor selected at that instant for gene x_i. If there are N possible realizations, then there are N vector functions, $\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N$ of the form $\mathbf{f}_k = (f_{k_1}^{(1)}, f_{k_2}^{(2)}, \ldots, f_{k_n}^{(n)})$, for $k = 1, 2, \ldots, N$, $1 \leq k_i \leq l(i)$ and where $f_{k_i}^{(i)} \in F_i$ $(i = 1, \ldots, n)$. In other words, the vector function $\mathbf{f}_k$: $\{0, 1\}^n \rightarrow \{0, 1\}^n$ acts as a transition function (mapping) representing a possible realization of the entire PBN. Such functions are commonly referred to as multiple-output Boolean functions. If we assume that the predictor for each gene is chosen independently of other predictors, then $N = \prod_{i=1}^{n} l(i)$. More complicated dependent selections are also possible.

Each of the N possible realizations can be thought of as a standard Boolean network that operates for one time step. In other words, at every state $x(t) \in \{0, 1\}^n$, one of the N Boolean networks is chosen and used to make the transition to the next state $x(t+1) \in \{0, 1\}^n$. The probability P_i that the ith (Boolean) network or realization is selected can be easily expressed in terms of the individual selection probabilities $c_j^{(i)}$ (see [34]). It is also possible to generalize the model such that the decision to select a new network realization is made with probability λ at every time step. In other words, at every time step, a coin is tossed with probability λ of falling on heads and if it does, then a new network realization is selected as described above; otherwise, the current network realization is used for the next time step. The original definition of PBNs, as described in [34], corresponds to the case $\lambda = 1$.

Another interpretation of a PBN is that all N Boolean networks are operating in parallel, but the information about the state of the whole system is shared by all of them. Thus, after one of the Boolean networks makes a transition, all the other Boolean networks are also "synchronized" to that state such that each of them is "ready" to make the next transition should it be selected. Very similar ideas have been used in the analysis of parallel and distributed systems, using so-called *stochastic automata networks* [91].

It is easy to see that the state space of a PBN is the same as of a standard Boolean network. Namely, there are 2^n possible states, each represented by a binary vector of length n. The difference is that while in a standard Boolean network, the transitions are deterministic (i.e., with probability 1), in a PBN, a state may transition to a number of other states, depending on which realization $\mathbf{f}_k$ is selected at that moment. Thus, the dynamics of a PBN can be modeled by a Markov chain and we can talk of a Markov chain *corresponding* to a PBN. However, we should point out that the PBN contains much more information than its corresponding Markov chain. Another way to say it is that it is possible for two different PBNs, containing different Boolean functions, to produce the same corresponding Markov chain. Thus, in addition to Markovian analysis, discussed in Section V-B, there are other tools that can be developed for studying the interactions of genes in PBNs.

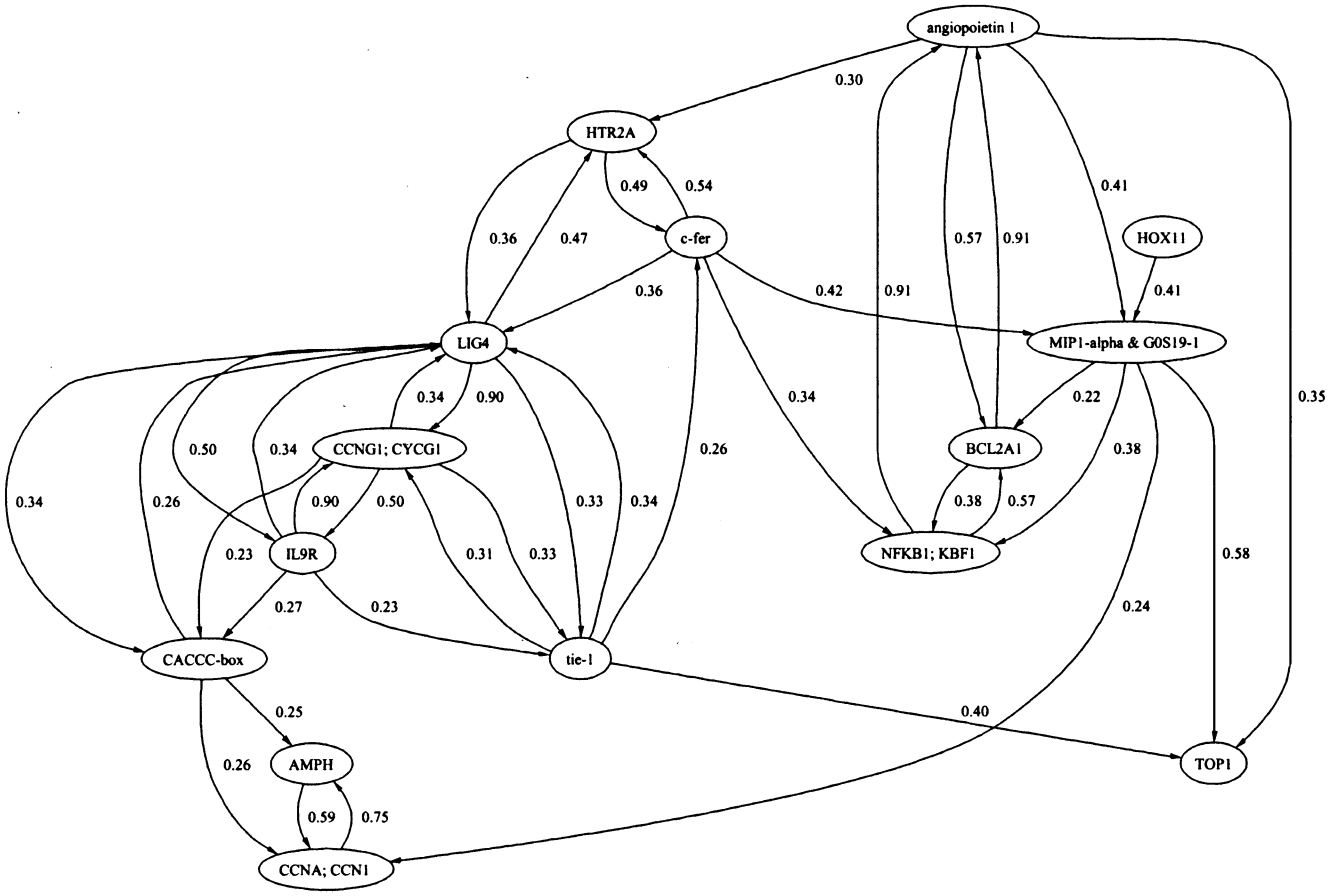


Fig. 5. An example of influences in a small network containing 15 genes. Each arrow represents the influence of a gene on another gene. The number next to the arrow is the magnitude of the influence. Only those influences that are above 0.2 are shown. The influence diagram is just a weighted directed graph. (The authors are grateful to Dr. R. Hashimoto for implementing the algorithms to produce this example.)

A. Influence and Sensitivity of Genes

One useful notion is the *influence* of a gene on another gene. Given a gene and a predictor for that gene, along with the genes used to make the prediction, it is important to distinguish those genes that have a major impact on the predictor from those that have only a minor impact. In other words, some genes are more "important" than others in determining the value of a target gene. Many examples of such biased regulation of gene expression are known to biologists. For example, the cell cycle regulator gene p21/WAF1/cip1 can be transcriptionally activated by a series of genes p53, smad4, AP2, BRCA1, and so on [35]. Among those genes, p53 has the most potent effect.

A good measure should reflect the extent to which a set of genes is capable of determining the value of the target gene. Although a number of approaches to measure the relative importance of variables in Boolean functions are possible [92], we have chosen the following.

The influence $I_j(f)$ of the variable x_j on the function f, with respect to the probability distribution $D(x)$, $x \in \{0, 1\}^n$, over the n-dimensional hypercube, is defined as

$$I_j(f) = \mathrm{E}_D\left[\frac{\partial f(x)}{\partial x_j}\right] \tag{5}$$

where $\mathrm{E}_D[\cdot]$ is the expectation operator with respect to distribution D, $(\partial f(x)/\partial x_j) = f(x^{(j,0)}) \oplus f(x^{(j,1)})$ is the partial derivative of the Boolean function f, the symbol $\oplus$ is addition modulo 2 (exclusive OR), and $x^{(j,k)} = (x_1, \ldots, x_{j-1}, k, x_{j+1}, \ldots, x_n)$, for $k = 0, 1$. In other words, (5) gives the influence as the probability [under the distribution $D(x)$] that a toggle of the jth variable (gene) changes the value of the function (predictor of the target gene). In the context of PBNs, the influence of gene x_k on gene x_i is given by [34]

$$I_k(x_i) = \sum_{j=1}^{l(i)} I_k\left(f_j^{(i)}\right) \cdot c_j^{(i)}. \tag{6}$$

Recall that $f_j^{(i)}$, $j = 1, \ldots, l(i)$, are the possible predictors for gene x_i. The *influence matrix* Γ contains the influences between every pair of genes: $\Gamma_{ij} = I_i(x_j)$. Fig. 5 shows an example of influences in a small network consisting of 15 genes, generated in an ongoing project with glioma data.

In an analogous manner, we can define the *sensitivity* of a gene as the sum of the influences acting upon it. Biologically, the sensitivity of a gene represents the stability or, in some sense, the autonomy of a gene. If the sensitivity of a gene is low, this implies that other genes have little affect on it. The

notions of influence and sensitivity can be easily generalized to sets of genes [34]. That is, we can define the influence of a set of genes on another set of genes. Typically, we are usually interested in *long-term influence*, which is the influence computed when the distribution $D(x)$ is the steady-state distribution of the PBN (see Section V-B).

The influence matrix Γ can be considered as an adjacency matrix of a weighted directed graph. Thus, the calculation of influences in a PBN can be thought of as a reduction of the rule-based dynamical model to a static directed graph structure, studied by some authors as models of gene regulatory networks [93]. In fact, many other models, such as Bayesian networks, are closely related to graphs. Indeed, Bayesian networks are just graphical models that explicitly represent probabilistic relationships between variables [87]. In [34], a relationship between PBNs and Bayesian networks was established. Specifically, the basic building blocks of Bayesian networks—conditional probabilities—can be explicitly determined in terms of the predictors, their selection probabilities, and the joint distributions of the predictors' input variables. Once again, the latter can be computed in the steady-state.

B. Markovian Analysis

A Markov chain is completely characterized by its state-transition matrix. For a PBN, this matrix is of size $2^n \times 2^n$ and the transition probabilities can be explicitly determined in terms of the selection probabilities $c_j^{(i)}$ and the Boolean functions $f_j^{(i)}$ [34]. For a given PBN, the corresponding Markov chain may consist of a number of *irreducible* subchains, which are sets of states from which the chain cannot "escape" once it enters them. Should this be the case, the long-term behavior of the network would depend on the initial distribution. This notion corresponds to the concept of attractors in Boolean networks. The attractor the system enters depends on the starting state. In a similar fashion, the *transient* states[6] in the Markov chain that lead to irreducible subchains correspond to the basins of attraction in Boolean networks. Thus, PBNs qualitatively exhibit the same dynamical properties as Boolean networks, but are inherently probabilistic. Similar interpretations of cell types and cellular functional states, as discussed in Section III-B, can be made for PBNs.

In dynamical systems analysis, the characterization of long-run behavior is often of prime importance. In the context of genetic networks, one may wish to know the long-term joint behavior of a certain group of genes or the long-term effect of one gene on a group of others. For example, the robustness or stability of genetic networks can be characterized by the sensitivity of the long-term behavior to single-gene perturbations. Let us discuss this question further, both in the context of gene perturbations and intervention.

[6]A transient state is one for which there is a positive probability of never entering it. Also, the long-run probability of being in such a state is 0.

C. Stochastic Perturbation Analysis

Suppose that a gene can get perturbed with (a small) probability p, independently of other genes. In the Boolean setting, this is represented by a flip of value from 1 to 0 or vice versa. This idea was already considered by Kauffman and Levin [94], [20] and corresponds to the bit-flipping mutation operator in *NK* landscapes. This also corresponds to the mutation operator in genetic algorithms [95]. This type of "randomization," namely allowing genes to randomly flip value, is biologically meaningful. Since the genome is not a closed system, but rather has inputs from the outside, it is known that genes may become either activated or inhibited due to external stimuli, such as mutagens, heat stress, etc. Thus, a network model should be able to capture this phenomenon. As we shall shortly see, there is another, pragmatic, advantage.

Suppose that at every step of the network, we have a realization of a so-called random *perturbation vector* $\gamma \in \{0, 1\}^n$. If the ith component of γ is equal to 1, then the ith gene is flipped, otherwise it is not. In general, γ need not be independent and identically distributed (i.i.d.), but will be assumed so for simplicity. Thus, we will suppose that $\Pr\{\gamma_i = 1\} = \mathrm{E}[\gamma_i] = p$ for all $i = 1, \ldots, n$. Let $x(t) \in \{0, 1\}^n$ be the state of the network at time t. Then, the next state $x(t+1)$ is given by

$$x(t+1) = \begin{cases} x(t) \oplus \gamma, & \text{with probability } 1-(1-p)^n \\ \mathbf{f}_k(x_1(t), \ldots, x_n(t)), & \text{with probability } (1-p)^n \end{cases} \tag{7}$$

where $\oplus$ is component-wise addition modulo 2 and $\mathbf{f}_k$, $k = 1, 2, \ldots, N$, is the transition function representing a possible realization of the entire PBN, as discussed above. In [96], an explicit formulation of the state-transition probabilities in terms of the Boolean functions and the probability of perturbation p, was derived.

It is fairly easy to show [96] that, for $p > 0$, the Markov chain corresponding to the PBN is ergodic. This means that it is aperiodic and irreducible. The latter implies that all states can be reached from all other states and that theoretically, sooner or later, every state will be visited. Practically, however, for very small values of p, most states will have very small long-run (steady-state) probabilities. Thus, informally speaking, the irreducible subchains would become "almost irreducible" in the sense that the chain would be likely to stay in them for very long periods of time, but on rare occasions, would escape due to some perturbations. Similarly, the transient states would become "almost transient" in that they would be visited extremely rarely. This idea parallels the situation with standard Boolean networks: if we were to allow a random perturbation to occur while in an attractor, most of the time, it would send us to its own basin of attraction, while occasionally, it may send us to another basin of attraction that would eventually flow to another attractor. As discussed earlier, Boolean networks operating in the critical phase are quite resistant to perturbations, but not that resistant so as to preclude malleability.

The practical benefit of allowing small perturbations is that it becomes possible to compute the steady-state distribution of the Markov chain. In other words, the limiting behavior of the Markov chain is independent of the initial distribution. Thus, it also allows us to assess the extent to which such perturbations affect the long-term behavior of the entire network.

Using recent results from perturbation theory of stochastic matrices [97], an explicit bound on the steady-state probabilities was derived in terms of the perturbation probability [96]. In other words, the bound gives a measure of how much the steady-state probability of a given state can change, in terms of the perturbation probability p. The bound is also given in terms of the *mean first-passage times*[7] to that state. An interesting implication of the result given in [96] is that the steady-state probabilities of those states of the network to which it is easy to transition from other states, in terms of mean first-passage times, are more resilient to random gene perturbations. In other words, the states of the network that are more "easily reachable" from other states are more stable in the presence of gene perturbations. These "stable" sets of states are hypothesized to correspond to cellular functional states. The first-passage times provide a conceptual link with the question of finding the best candidate genes for intervention. We turn to this now.

D. Intervention

As we just discussed, most genetic networks are stable in the sense that they typically operate in sets of states that are stable to perturbations. In Boolean networks, this corresponds to a likely return to the attractor; in PBNs, it corresponds to a low sensitivity of the steady-state probabilities. The ideas are fundamentally the same. Now, let us turn the problem around. Suppose we wish to elicit certain long-run behavior from the network. What genes would make the best candidates for intervention so as to increase the likelihood of this behavior? That is, suppose that the network is operating in a certain "undesirable" set of states and we wish to "persuade" it to transition into a "desirable" set of states by perturbing some gene. For practical reasons, we may wish to be able to intervene with as few genes as possible in order to achieve our goals. Such an approach can expedite the systematic search and identification of potential drug targets in cancer therapy.

This question was taken up in [96], where several methods for finding the best candidate genes for intervention, based on first-passage times, were developed. The first-passage times provide a natural way to capture the goals of intervention in the sense that we wish to transition to certain states (or avoid certain states, if that is our goal) "as quickly as possible," or, alternatively, by maximizing the probability of reaching such states before a certain time.

Suppose, for example, that we wish to persuade the network to flow into a set of states (irreducible subchain—the counterpart of an attractor) representing apoptosis. This could be very useful, for example, in the case of cancer cells, which may keep proliferating. We may be able to achieve this action via the perturbation (intervention) of *several* different genes, but some of them may be better in the sense that the mean first-passage time to enter apoptosis is shorter. There are numerous examples in biology when the (in)activation of one gene can lead much quicker (or with a higher probability) to a certain cellular functional state or phenotype than the (in)activation of another gene. Such is the case with p53 and telomerase genes, for example. In a stable cancer cell line, when p53 is activated in the cells, for example, in response to radiation, the cells undergo rapid growth inhibition and apoptosis in as short as a few hours [98]. In contrast, inhibition of the telomerase gene also leads to cell growth inhibition, differentiation, and cell death, but only after cells go through a number of cell divisions [99], which takes a longer time to occur than via p53.

The type of intervention described above—one that allows us to intervene with a gene—can be useful for modulating the dynamics of the network, but it is not able to alter the underlying structure of the network. Accordingly, the steady-state distribution remains unchanged. However, a disbalance between certain sets of states, which is characteristic of neoplasia in view of gene regulatory networks, can be caused by mutations of the "wiring" of certain genes, thus permanently altering the state-transition structure and, consequently, the long-run behavior of the network [21].

Therefore, it is prudent to develop a methodology for altering the steady-state probabilities of certain states or sets of states with minimal modifications to the rule-based structure. The motivation is that these states may represent different phenotypes or cellular functional states, such as cell invasion and quiescence, and we would like to decrease the probability that the whole network will end up in an undesirable set of states and increase the probability that it will end up in a desirable set of states. One mechanism by which we can accomplish this consists of altering some Boolean functions (predictors) in the PBN. For practical reasons, as above, we may wish to alter as few functions as possible. Such alterations to the rules of regulation may be possible by the introduction of a factor or drug that alters the extant behavior.

In [100], a methodology for altering the steady-state probabilities of certain states or sets of states, with minimal modifications to the underlying rule-based structure, was developed. This approach was framed as an optimization problem that can be solved using genetic algorithms (GA), which are well suited for capturing the underlying structure of PBNs and are able to locate the optimal solution in a highly efficient manner. For example, in some computer simulations that were performed, the genetic algorithm was able to locate the optimal solution (structural alteration) in only 200 steps (evaluations of the fitness function), out of a total of 21 billion possibilities, which is the number of steps a brute-force approach would have to take. The reason for such high efficiency of the genetic algorithm is due to the embedded structure in the PBN that can be exploited.

[7]The mean first-passage time from state x to state y is the expected time it will take to reach y starting in x.

The use of genetic algorithms for modifying the structure of Boolean networks has recently been studied, in the context of evolution [101], [102].

VI. Concluding Remarks

The paper contains an overview of Boolean and probabilistic Boolean modeling of genetic networks. It is not meant to be exhaustive and only presents a selection of topics that may be of interest to the engineering, computer science, and mathematics community. Those readers that wish to dig deeper into the mathematics behind Boolean networks can consult a number of good papers in the physics literature. Kauffman's book [20] is also an excellent starting point. PBNs also present many interesting and challenging problems. A fascinating aspect of the research on PBNs is that it involves and spans so many fields and topics, such as random processes, estimation, optimization, control, parallel and distributed systems, computational learning theory, and signal processing, just to name a few.

Some of the current topics of research involve the design of fast, scalable inference algorithms and the associated study of robustness and complexity. Also under development are various approaches for constructing subnetworks of genes, using influences and sensitivities. For steady-state analysis, a number of methods for convergence analysis for Monte Carlo simulation of PBNs are under investigation.

Acknowledgment

The authors would like to acknowledge the following colleagues for many useful discussions and invaluable contributions without which this work would not be possible: S. Kim, M. Bittner, I. Gluhovsky, R. Hashimoto, J. Astola, I. Tabus, O. Yli-Harja, H. Lähdesmäki, A. Saarinen, H. Li, and E. Suh.

References

[1] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, vol. 270, pp. 467–470, 1995.

[2] J. E. Celis, M. Kruhøffer, I. Gromova, C. Frederiksen, M. Østergaard, T. Thykjaer, P. Gromov, J. Yu, H. Pálsdóttir, N. Magnusson, and T. F. Ørntoft, "Gene expression profiling: Monitoring transcription and translation products using DNA microarrays and proteomics," *FEBS Lett.*, vol. 480, no. 1, pp. 2–16, 2000.

[3] T. R. Hughes, M. Mao, A. R. Jones, J. Burchard, M. J. Marton, K. W. Shannon, S. M. Lefkowitz, M. Ziman, J. M. Schelter, M. R. Meyer, S. Kobayashi, C. Davis, H. Dai, Y. D. He, S. B. Stephaniants, G. Cavet, W. L. Walker, A. West, E. Coffey, D. D. Shoemaker, R. Stoughton, A. P. Blanchard, S. H. Friend, and P. S. Linsley, "Expression profiling using microarrays fabricated by an ink-jet oligonucleotide synthesizer," *Nature Biotechnol.*, vol. 19, pp. 342–347, 2001.

[4] R. J. Lipshutz, S. P. A. Fodor, T. R. Gingeras, and D. J. Lockhart, "High density synthetic oligonucleotide arrays," *Nature Genetics*, vol. 21, pp. 20–24, 1999.

[5] D. J. Lockhart and E. A. Winzeler, "Genomics, gene expression and DNA arrays," *Nature*, vol. 405, pp. 827–836, 2000.

[6] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531–537, 1999.

[7] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, "Tissue classification with gene expression profiles," *J. Comput. Biol.*, vol. 7, pp. 559–583, 2000.

[8] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, and P. S. Meltzer, "Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks," *Nature Medicine*, vol. 7, pp. 673–679, 2001.

[9] I. Hedenfalk, D. Duggan, Y. Chen, M. Radmacher, M. Bittner, R. Simon, P. Meltzer, B. Gusterson, M. Esteller, M. Raffeld, Z. Yakhini, A. Ben-Dor, E. Dougherty, J. Kononen, L. Bubendorf, W. Fehrle, S. Pittaluga, S. Gruvverger, N. Loman, O. Johannsson, H. Olsson, B. Wifond, G. Sauter, O. P. Kallioniemi, A. Borg, and J. Trent, "Gene expression profiles distinguish hereditary breast cancers," *New England J. Med.*, vol. 34, pp. 539–548, 2001.

[10] M. Anthony and N. Biggs, *Computational Learning Theory*. Cambridge, U.K.: Cambridge Univ. Press, 1992.

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

[12] P. Smolen, D. Baxter, and J. Byrne, "Mathematical modeling of gene networks," *Neuron*, vol. 26, pp. 567–580, 2000.

[13] J. Hasty, D. McMillen, F. Isaacs, and J. J. Collins, "Computational studies of gene regulatory networks: *In numero* molecular biology," *Nature Reviews Genetics*, vol. 2, pp. 268–279, 2001.

[14] H. de Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *J. Comput. Biol.*, vol. 9, no. 1, pp. 69–103, 2002.

[15] J. M. Bower and H. Bolouri, Eds., *Computational Modeling of Genetic and Biochemical Networks*. Cambridge, MA: MIT Press, 2001.

[16] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theoret. Biol.*, vol. 22, pp. 437–467, 1969.

[17] ——, "Homeostasis and differentiation in random genetic control networks," *Nature*, vol. 224, pp. 177–178, 1969.

[18] K. Glass and S. A. Kauffman, "The logical analysis of continuous, nonlinear biochemical control networks," *J. Theoret. Biol.*, vol. 39, pp. 103–129, 1973.

[19] S. A. Kauffman, "The large scale structure and dynamics of genetic control circuits: an ensemble approach," *J. Theoret. Biol.*, vol. 44, pp. 167–190, 1974.

[20] ——, *The Origins of Order: Self-Organization and Selection in Evolution*. New York: Oxford Univ. Press, 1993.

[21] S. Huang, "Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorigenesis and drug discovery," *J. Mol. Med.*, vol. 77, pp. 469–480, 1999.

[22] R. Somogyi and C. Sniegoski, "Modeling the complexity of gene networks: Understanding multigenic and pleiotropic regulation," *Complexity*, vol. 1, pp. 45–63, 1996.

[23] C.-H. Yuh, H. Bolouri, and E. H. Davidson, "Genomic cis-regulatory logic: Experimental and computational analysis of a sea urchin gene," *Science*, vol. 279, pp. 1896–1902, 1998.

[24] J. W. Bodnar, "Programming the *Drosophila* embryo," *J. Theoret. Biol.*, vol. 188, pp. 391–445, 1997.

[25] L. Mendoza, D. Thieffry, and E. R. Alvarez-Buylla, "Genetic control of flower morphogenesis in Arabidopsis thaliana: A logical analysis," *Bioinformat.*, vol. 15, pp. 593–606, 1999.

[26] S. Huang and D. E. Ingber, "Shape-dependent control of cell growth, differentiation, and apoptosis: Switching between attractors in cell regulatory networks," *Exp. Cell Res.*, vol. 261, no. 1, pp. 91–103, 2000.

[27] I. Shmulevich and W. Zhang, "Binary analysis and optimization-based normalization of gene expression data," *Bioinformatics*, vol. 18, no. 4, pp. 555–565, 2002.

[28] Y. Chen, E. R. Dougherty, and M. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Biomed. Opt.*, vol. 2, pp. 364–374, 1997.

[29] M. K. Kerr, E. H. Leiter, L. Picard, and G. A. Churchill, "Sources of variation in microarray experiments," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds. Boston, MA: Kluwer, 2002.

[30] H. H. McAdams and A. Arkin, "It's a noisy business! Genetic regulation at the nanomolar scale," *Trends in Genetics*, vol. 15, pp. 65–69, 1999.

[31] Z. Szallasi and S. Liang, "Modeling the normal and neoplastic cell cycle with 'realistic Boolean genetic networks': Their application for understanding carcinogenesis and assessing therapeutic strategies," in *Proc. Pacific Symp. Biocomputing 3*, 1998, pp. 66–76.

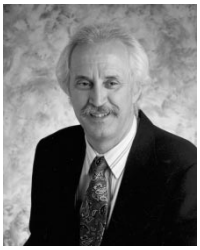
[32] A. Wuensche, "Genomic regulation modeled as a network with basins of attraction," in *Proc. Pacific Symp. Biocomputing*, vol. 3, 1998, pp. 89–102.

[33] R. Thomas, D. Thieffry, and M. Kaufman, "Dynamical behavior of biological regulatory networks—I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state," *Bull. Math. Biol.*, vol. 57, pp. 247–276, 1995.
[34] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: A rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.
[35] A. L. Gartel and A. L. Tyner, "Transcriptional regulation of the p21(WAF1/CIP1) gene," *Exp. Cell Res.*, vol. 246, no. 2, pp. 280–289, 1999.
[36] E. Schneider, M. Montenarh, and P. Wagner, "Regulation of CAK kinase activity by p53," *Oncogene*, vol. 17, pp. 2733–2741, 1998.
[37] J. P. Fitch, E. J. Coyle, and N. C. Gallagher, "Root properties and convergence rates of median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 1, pp. 230–240, 1985.
[38] P. Wendt, E. Coyle, and N. C. Gallagher, "Stack filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, pp. 898–911, 1986.
[39] N. C. Gallagher and G. L. Wise, "A theoretical analysis of the properties of median filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-29, no. 6, 1981.
[40] P.-T. Yu and E. J. Coyle, "Convergence behavior and N-roots of stack filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 38, no. 9, pp. 888–888, 1990.
[41] M. Gabbouj, M. P-T., M. Yu, and E. J. Coyle, "Convergence behavior and root signal sets of stack filters," *Circuits, Syst., Signal Process.*, vol. 11, no. 1, pp. 888–888, 1992.
[42] Q. Wang, M. Gabbouj, and Y. Neuvo, "Root properties of morphological filters," *Signal Processing*, vol. 34, pp. 131–148, 1993.
[43] J. Serra, *Image Analysis and Mathematical Morphology*. London, U.K.: Academic, 1982, vol. 1.
[44] E. R. Dougherty, *An Introduction to Morphological Image Processing*. Bellingham, WA: SPIE, 1992.
[45] G. Moran, "On the period-two-property of the majority operator in infinite graphs," *Trans. Amer. Math. Soc.*, vol. 347, no. 5, pp. 1649–1667, 1995.
[46] P.-T. Yu and E. J. Coyle, "The classification and associative memory capability of stack filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 40, no. 10, pp. 2483–2497, 1992.
[47] ——, "On the existence and design of the best stack filter based associative memory," *IEEE Trans. Circuits Syst.*, vol. 39, no. 3, pp. 171–184, 1992.
[48] E. F. Codd, *Cellular Automata*. New York: Academic, 1968.
[49] M. Mitchell, J. P. Crutchfield, and P. T. Hraber, "Evolving cellular automata to perform computations: Mechanisms and impediments," *Physica D*, vol. 75, pp. 361–391, 1994.
[50] F. Jiménez Morales, J. P. Crutchfield, and M. Mitchell, "Evolving two-dimensional cellular automata to perform density classification: A report on work in progress," *Parallel Comput.*, vol. 27, no. 5, pp. 539–553, 2001.
[51] A. W. Burks, "Von Neumann's self-reproducing automata," in *Essays on Cellular Automata*, A. W. Burks, Ed. Champaign, IL: Univ. Illinois Press, 1970, pp. 3–74.
[52] J. P. Fitch, E. J. Coyle, and N. C. Gallagher, "Median filtering by threshold decomposition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32, no. 6, pp. 1183–1188, 1984.
[53] J. H. Holl, *Hidden Order: How Adaptation Builds Complexity*. Reading, MA: Helix Books, 1995.
[54] D. M. Wolf and F. H. Eeckman, "On the relationship between genomic regulatory element organization and gene regulatory dynamics," *J. Theoret. Biol.*, vol. 195, pp. 167–186, 1998.
[55] A. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, and J. D. Watson, *Molecular Biology of the Cell*. New York: Garland, 1983.
[56] S. A. Kauffman, *At Home in the Universe*. New York: Oxford Univ. Press, 1995.
[57] P. T. Spellman, G. Sherlock, M. Q. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization," *Mol. Biol. Cell*, vol. 9, pp. 3273–3297, 1998.
[58] S. Huang, "Genomics, complexity and drug discovery: Insights from Boolean network models of cellular regulation," *Pharmacogenomics*, vol. 2, no. 3, pp. 203–222, 2001.
[59] S. Huang, "Cell state dynamics and tumorigenesis in Boolean regulatory networks," *InterJournal Genetics*. MS: 416, [Online]. Available: http://www.interjournal.org.
[60] S. Liang, S. Fuhrman, and R. Somogyi, "REVEAL, a general reverse engineering algorithm for inference of genetic network architectures," in *Proc. Pacific Symp. Biocomputing 3*, 1998, pp. 18–29.
[61] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano, "Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions," in *Proc. 9th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA'98)*, 1998, pp. 695–702.
[62] T. Akutsu, S. Miyano, and S. Kuhara, "Identification of genetic networks from a small number of gene expression patterns under the Boolean network model," in *Proc. Pacific Symp. Biocomputing 4*, 1999, pp. 17–28.
[63] ——, "Inferring qualitative relations in genetic networks and metabolic pathways," *Bioinformatics*, vol. 16, pp. 727–734, 2000.
[64] T. E. Ideker, V. Thorsson, and R. M. Karp, "Discovery of regulatory interactions through perturbation: Inference and experimental design," in *Proc. Pacific Symp. Biocomputing 5*, 2000, pp. 302–313.
[65] R. M. Karp, R. Stoughton, and K. Y. Yeung, "Algorithms for choosing differential gene expression experiments," in *Proc. RECOMB99 (ACM)*, 1999, pp. 208–217.
[66] Y. Maki, D. Tominaga, M. Okamoto, S. Watanabe, and Y. Eguchi, "Development of a system for the inference of large scale genetic networks," in *Proc. Pacific Symp. Biocomputing*, vol. 6, 2001, pp. 446–458.
[67] K. Noda, A. Shinohara, M. Takeda, S. Matsumoto, S. Miyano, and S. Kuhara, "Finding genetic network from experiments by weighted network model," *Genome Informatics*, vol. 9, pp. 141–150, 1998.
[68] I. Shmulevich, A. Saarinen, O. Yli-Harja, and J. Astola, "Inference of genetic regulatory networks under the best-fit extension paradigm," in *Computational and Statistical Approaches To Genomics*, W. Zhang and I. Shmulevich, Eds. Boston, MA: Kluwer, 2002.
[69] L. G. Valiant, "A theory of the learnable," *Commun. Assoc. Comput. Mach.*, vol. 27, pp. 1134–1142, 1984.
[70] Y. Crama, P. Hammer, and T. Ibaraki, "Cause–effect relationships and partially defined Boolean functions," *Ann. Oper. Res.*, vol. 16, pp. 299–326, 1988.
[71] L. Pitt and L. G. Valiant, "Computational limitations on learning from examples," *J. ACM*, vol. 35, pp. 965–984, 1988.
[72] E. Boros, T. Ibaraki, and K. Makino, "Error-free and best-fit extensions of partially defined Boolean functions," *Inform. Computat.*, vol. 140, pp. 254–283, 1998.
[73] I. Shmulevich, M. Gabbouj, and J. Astola, "Complexity of the consistency problem for certain post classes," *IEEE Trans. Syst., Man, Cybernet. B*, vol. 31, no. 2, pp. 251–253, Apr. 2001.
[74] E. J. Coyle and J. H. Lin, "Stack filters and the mean absolute error criterion," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 8, pp. 1244–1254, Aug. 1988.
[75] E. J. Coyle, J. H. Lin, and M. Gabbouj, "Optimal stack filtering and the estimation and structural approaches to image processing," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 37, no. 12, pp. 2037–2066, 1989.
[76] R. Yang, L. Yin, M. Gabbouj, J. Astola, and Y. Neuvo, "Optimal weighted median filtering under structural constraints," *IEEE Trans. Signal Processing*, vol. 43, pp. 591–604, Mar. 1995.
[77] E. R. Dougherty and R. P. Loce, "Precision of morphological-representation estimators for translation-invariant binary filters: Increasing and nonincreasing," *Signal Processing*, vol. 40, pp. 129–154, 1994.
[78] R. P. Loce and E. R. Dougherty, "Optimal morphological restoration: The morphological filter mean-absolute-error theorem," *Vis. Commun. Image Representation*, vol. 3, no. 4, 1992.
[79] E. R. Dougherty and J. T. Astola, *Nonlinear Filters for Image Processing*. Piscataway, NJ: SPIE/IEEE Press, 1999.
[80] E. R. Dougherty and Y. Chen, "Optimal and adaptive design of logical granulometric filters," *Adv. Imaging Electron Phys.*, vol. 117, pp. 1–71, 2001.
[81] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Processing*, vol. 80, no. 10, pp. 2219–2235, 2000.
[82] S. Kim, E. R. Dougherty, Y. Chen, K. Sivakumar, P. Meltzer, J. M. Trent, and M. Bittner, "Multivariate measurement of gene expression relationships," *Genomics*, vol. 67, pp. 201–209, 2000.
[83] S. Kim, E. R. Dougherty, M. L. Bittner, Y. Chen, K. Sivakumar, P. Meltzer, and J. M. Trent, "General nonlinear framework for the analysis of gene interaction via multivariate expression arrays," *J. Biomed. Opt.*, vol. 5, no. 4, pp. 411–424, 2000.
[84] L. L. Scharf, *Statistical Signal Processing*. Reading, MA: Addison-Wesley, 1991.

[85] E. B. Suh, E. R. Dougherty, S. Kim, M. L. Bittner, Y. Chen, D. E. Russ, and R. Martino, "Parallel computation and visualization tools for codetermination analysis of multivariate gene-expression relations," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds. Boston, MA: Kluwer, 2002.
[86] S. E. Wildsmith and F. J. Elcock, "Microarrays under the microscope," *J. Clin. Pathol.: Mol. Pathol.*, vol. 54, pp. 8–16, 2001.
[87] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Mateo, CA: Morgan Kaufmann, 1997.
[88] V. Vapnik, *The Nature of Statistical Learning Theory*, 2nd ed. New York: Springer-Verlag, 2000.
[89] I. Tabus and J. Astola, "On the use of the MDL principle in gene expression prediction," *J. Appl. Signal Process.*, vol. 4, pp. 297–303, 2001.
[90] I. Tabus, J. Rissanen, and J. Astola, "Normalized maximum likelihood models for Boolean regression with application to prediction and classification in genomics," in *Computational and Statistical Approaches to Genomics*, W. Zhang and I. Shmulevich, Eds. Boston, MA: Kluwer, 2002.
[91] B. Plateau and J-M. Fourneau, "A methodology for solving Markov models of parallel systems," *J. Parallel Distrib. Comput.*, vol. 12, pp. 370–387, 1991.
[92] P. L. Hammer, A. Kogan, and U. G. Rothblum, "Evaluation, strength, and relevance of variables of Boolean functions," *SIAM J. Discrete Math.*, vol. 13, no. 3, pp. 302–312, 2000.
[93] A. Wagner, "How to reconstruct a large genetic network from n gene perturbations in fewer than n^2 easy steps," *Bioinformatics*, vol. 17, no. 12, pp. 1183–1197, 2001.
[94] S. A. Kauffman and S. Levin, "Toward a general theory of adaptive walks on rugged landscapes," *J. Theoret. Biol.*, vol. 128, pp. 11–45, 1987.
[95] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
[96] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.
[97] G. E. Cho and C. D. Meyer, "Markov chain sensitivity measured by mean first passage times," *Linear Algebra and Its Applications*, vol. 316, pp. 21–28, 2000.
[98] T. Kobayashi, S.-B. Ruan, J. R. Jabbur, U. Consoli, K. Clodi, H. Shiku, L. Owen-Schaub, M. Andreeff, J. Reed, and W. Zhang, "Differential p53 phosphorylation and activation of apoptosis-promoting genes Bax and Fas/APO-1 by radiation and ara-C treatment," *Cell Death and Differentiation*, vol. 5, pp. 584–591, 1998.
[99] C. B. Harley, "Telomere loss: Mitotic clock or genetic time bomb?," *Mutation Res.*, vol. 256, pp. 271–282, 1991.
[100] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Control of stationary behavior in probabilistic Boolean networks by means of structural intervention," J. Biologic. Syst., to be published.
[101] M. D. Stern, "Emergence of homeostasis and noise imprinting in an evolution model," *Proc. Nat. Acad. Sciences USA*, vol. 96, pp. 10 746–10 751, 1999.
[102] S. Bornholdt and K. Sneppen, "Robustness as an evolutionary principle," *Proc. Royal Soc. London B*, vol. 266, pp. 2281–2286, 2000.

Ilya Shmulevich (Member, IEEE) received the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, in 1997.

In 1997–1998, he was a Post-Doctoral Researcher at the Nijmegen Institute for Cognition and Information at the University of Nijmegen and National Research Institute for Mathematics and Computer Science at the University of Amsterdam, The Netherlands, where he studied computational models of music perception and recognition. In 1998–2000, he worked as a Senior Researcher at the Tampere International Center for Signal Processing at the Signal Processing Laboratory in Tampere University of Technology, Tampere, Finland. Presently, he is an Assistant Professor at the Cancer Genomics Laboratory at The University of Texas M. D. Anderson Cancer Center, Houston. He is an Associate Editor (in Genomics) of Environmental Health Perspectives: Toxicogenomics. His research interests include computational genomics, nonlinear signal and image processing, computational learning theory, and music recognition and perception.



Edward R. Dougherty received the M.S. degree in computer science from Stevens Institute of Technology, Hoboken, NJ, and the Ph.D. degree in mathematics from Rutgers University, New Brunswick, NJ.

He is a Professor of Electrical Engineering at Texas A&M University and is Director of the Genomic Signal Processing Laboratory. He is also an Adjunct Professor in the Department of Pathology at The University of Texas M. D. Anderson Cancer Center and a Visiting Researcher at the National Human Genome Research Institute of the National Institutes of Health. His current research is focused on the use of gene-expression data for phenotype classification and the development of genetic regulatory networks.

Wei Zhang received the Ph.D. degree in molecular biology from The University of Texas Graduate School of Biomedical Sciences, Houston, TX, in 1992.

He joined the faculty of the University of Texas M. D. Anderson Cancer Center in 1994. Currently, he is an Associate Professor of Pathology and Cancer Biology in the Department of Pathology. He has been the Director of the Cancer Genomics Core Lab since its inception in 1999. He is an Associate Editor for Clinical Cancer Research and serves on the editorial board of three other cancer research journals. His research interests are genomic/systems biology, molecular biology, and cancer biology.