Intervention in Probabilistic Gene Regulatory Networks

Aniruddha Datta[*,1], Ranadip Pal[1] and Edward R. Dougherty[1,2]

[1]*Department of Electrical Engineering and Genomic Signal Processing Laboratory, Texas A&M University, College Station, TX 77843-3128, USA*

[2]*Computational Biology Division, Translational Genomics Research Institute, Phoenix, AZ 85004, USA*

Abstract: In recent years, there has been a considerable amount of interest in the area of Genomic Signal Processing, which is the engineering discipline that studies the processing of genomic signals. Since regulatory decisions within the cell utilize numerous inputs, analytical tools are necessary to model the multivariate influences on decision-making produced by complex genetic networks. Signal processing approaches such as detection, prediction and classification have been used in the recent past to construct genetic regulatory networks capable of modeling genetic behavior. To accommodate the large amount of uncertainty associated with this kind of modeling, many of the networks proposed are probabilistic. One of the objectives of network modeling is to use the network to design different intervention approaches for affecting the time evolution of the gene activity profile of the network. More specifically, one is interested in intervening to help the network avoid undesirable states such as those associated with a disease. This paper provides a tutorial survey of the intervention approaches developed so far in the literature for probabilistic gene networks (probabilistic Boolean networks) and outlines some of the open challenges that remain.

Keywords: Gene regulatory network, markov chain, steady-state distribution, optimal control, dynamic programming, context sensitive networks.

1. INTRODUCTION

From a translational perspective, the ultimate objective of genetic regulatory network modeling is to use the network to design different approaches for affecting network dynamics in such a way as to avoid undesirable phenotypes, for instance, cancer. In this paper we present a tutorial survey of the results obtained to date on intervention in the context of probabilistic gene regulatory networks, which, owing to their original binary formulation and their usual application using binary and ternary gene-expression quantization, are generically called *probabilistic Boolean networks* (*PBNs*) [1]. These are essentially probabilistic generalizations of the standard Boolean networks introduced by Kauffman [2-4] that allow the incorporation of uncertainty into the inter-gene relationships. Given a PBN, the transition from one state to the next takes place in accordance with certain transition probabilities and their dynamics, and hence intervention, can be studied in the context of homogeneous Markov chains with finite state spaces.

A major goal of functional genomics is to screen for genes that determine specific cellular phenotypes (disease) and model their activity in such a way that normal and abnormal behavior can be differentiated. The pragmatic manifestation of this goal is the development of therapies based on the disruption or mitigation of aberrant gene function contributing to the pathology of a disease. Mitigation would be accomplished by the use of drugs to act on the gene products. Engineering therapeutic tools involves synthesizing nonlinear dynamical networks, analyzing these networks to characterize gene regulation, and developing intervention strategies to modify dynamical behavior. For instance, changes in network connectivity or functional relationships among the genes in a network, *via* mutations or re-arrangements, can lead to steady-state behavior associated with tumorigenesis, and this is likely to lead to a cancerous phenotype unless corrective therapeutic intervention is applied.

To date, intervention studies have used three different approaches: (i) resetting the state of the PBN, as necessary, to a more desirable initial state and letting the network evolve from there [5]; (ii) changing the steady-state (long-run) behavior of the network by minimally altering its rule-based structure [6]; and (iii) manipulating external (control) variables that alter the transition probabilities of the network and can, therefore, be used to desirably affect its dynamic evolution [7]. The control-theoretic approach has subsequently been extended. First, the optimal intervention algorithm has been modified to accommodate the case where the entire state vector, or *gene activity profile* (*GAP*) as it is known, is not available for measurement [8]. Second, whereas the original control-theoretic approach has been developed in the framework of *instantaneously random* PBNs, the intervention results have been extended to *context-sensitive* PBNs (terminology to be defined shortly) [9].

The paper is organized in the following manner: Section 2 reviews the necessary essentials of PBNs; Section 3 discusses intervention limited to a one-time flipping of the expression status of a single gene; Section 4 considers intervention to alter the steady-state behavior of the network; Section 5 formulates the intervention problem in probabilistic gene regulatory networks as an optimal control problem that is then solved using the standard approach of *dynamic programming*; Section 6 extends the results of Section 5 to the imperfect information case; Section 7

*Address correspondence to this author at the Room No. 216N, Zachry Engineering Center, Tamu 3128, College Station, TX, 77843, USA; Tel: 979-845-5917; Fax: 979-845-6259; E-mail: datta@ee.tamu.edu

1574-8936/06 $50.00+.00 © 2006 Bentham Science Publishers Ltd.

extends the results to context-sensitive PBNs; and Section 8 contains some concluding remarks.

2. REVIEW OF PROBABILISTIC BOOLEAN NETWORKS

Our review focuses on aspects critical to intervention and we refer to the literature for more detailed accounts [1, 10, 11]. In the original binary formulation of probabilistic Boolean networks, each gene can take on one of two binary values, 0 or 1 [1]. A 0 for a gene corresponds to the gene not being expressed (OFF) and a 1 corresponds to the gene being expressed (ON). The functional dependency of a given gene value on all the genes in the network is given in terms of a single Boolean function or a family of Boolean functions. The case of a single Boolean function for each gene arises when the functional relationships between the different genes in the network are known and are static. Although such a situation is not likely to occur in practice, networks of this type, referred to as *Boolean networks*, have been extensively studied in the literature [4]. To account for uncertainty in our knowledge of the functional dependencies between the different genes, one can postulate that the expression level of a particular gene in the network is described by a finite family of Boolean functions such that each member of the family is assumed to describe the functional relationship with a certain probability, thereby leading to probabilistic Boolean networks, as introduced in [1]. In either case, dynamics are introduced to the network by assuming that at each time step the value of each gene is updated using the Boolean functions evaluated at the gene values from the previous time step. For PBNs, the expression level of each gene will be updated in accordance with the probabilities corresponding to the different Boolean functions associated with a particular gene.

To concretize matters, let us begin with the formal definition of a Boolean network. A *Boolean network* is defined by a set of nodes, $V = \{x_1, x_2,\ldots, x_n\}$, and a collection of Boolean functions, $F = \{f_1, f_2,\ldots, f_n\}$. Each x_k represents the state (expression level) of a gene, g_k, where $x_k = 1$ or $x_k = 0$, depending on whether the gene is or is not expressed. The Boolean functions represent the rules of regulatory interaction between genes. Network dynamics result from a synchronous clock with times $t = 0, 1, 2,\ldots$. The value of gene g_k at time $t + 1$ is given by $x_k(t + 1) = f_k(x_{k1}, x_{k2},\ldots, x_{k,m(k)})$, where the nodes in the argument of f_k form the regulatory set for x_k (gene g_k). The numbers of genes in the regulatory sets define the connectivity of the network, with maximum connectivity often limited. At time point t, the state vector $x(t) = [x_1(t), x_2(t),\ldots, x_n(t)]$ is called the *gene activity profile* (*GAP*). The functions together with the regulatory sets determine the network wiring.

A *probabilistic Boolean network* (*PBN*) is defined by a set of nodes, $V = \{x_1, x_2,\ldots, x_n\}$ and a set of r network-wide functions, $\mathbf{f}_1, \mathbf{f}_2,\ldots, \mathbf{f}_r$, meaning that $\mathbf{f}_k$ consists of all predictors for all genes in the network. Specifically, $\mathbf{f}_k = [f_{k1}, f_{k2},\ldots, f_{kn}]$, where f_{kj} determines the value of gene g_j. Each $\mathbf{f}_k$ determines a constituent Boolean network of the PBN, and at each time point there is a positive probability q of switching network function (the governing constituent Boolean network). If a decision is made to switch networks then a new network function is chosen randomly, with the probability of selecting $\mathbf{f}_k$ being given by c_k. *Random perturbation* can be introduced by assuming that at each time point, each gene has a positive probability p of flipping values, and genes flip independently. If $q = 1$, then the network function switches at every time point and the network is said to be *instantaneously random*; if $q < 1$, the network is said to be *context sensitive*. The more general notion of a probabilistic gene regulatory network results from allowing an arbitrary finite quantization in place of the binary 0-1 quantization of the nodes; however, even for finer quantization the terminology "probabilistic Boolean network" is usually employed, owing to both the common application to binary and ternary networks and the logical character of the network functions for any finite quantization.

Context sensitivity models the practical situation in which there are latent variables outside the model network whose changes affect the rule structure of the network. For context-sensitive PBNs, one of the following events occurs at each time point: (1) the current network function is applied, the PBN transitions accordingly, and the network function remains the same for the next transition; (2) the current network function is applied, the PBN transitions accordingly, and a new network function is selected for the next transition; (3) there is a random perturbation and the network function remains the same for the next transition; or (4) there is a random perturbation and a new network function is selected for the next transition. The first and third cases correspond exactly to the situation with a Boolean network: so long as these cases apply, meaning that the network function remains fixed, the PBN behaves as a Boolean network.

Attractors play a key role in Boolean networks. Given a starting state, in a finite number of steps the network will move to a set of states through which it will endlessly cycle. Such a set is called an *attractor cycle* and states in attractor cycles are called attractors. Biologically, attractors have been conjectured to characterize phenotypes [4]. The full set of states is partitioned according to which attractor cycle an initial state will lead to. The class of states corresponding to an attractor cycle is called its *basin of attraction*. Non-attractor states are transient. They are visited no more than once on any network trajectory. Attractors characterize the long-run behavior of a Boolean network. By definition, the attractors of a PBN consist of the attractors of its constituent Boolean networks.

In a context-sensitive Markov chain, the collection of all state-function pairs, $(x, \mathbf{f}_k)$, forms a finite-state Markov chain and, under the assumption of random perturbation, $p > 0$, the Markov chain is ergodic, so that its stationary distribution is also a steady-state distribution. A basic reason that instantaneously random PBNs are easier to analyze is that, for them, the Markov chain consists of the states of the network. A key issue in PBN analysis concerns the steady-state probabilities of attractors [12]. If the switching and perturbation probabilities are small, the attractors possess a large majority of the steady-state mass.

To this point we have made no assumption on the formation of the network functions. Now, suppose that for each gene g_i, we are given $l(i)$ possible Boolean functions, $f_1^{(i)}, f_2^{(i)},\ldots.., f_{l(i)}^{(i)}$ that can be used to describe the

dependency of x_i on $x_1, x_2, \ldots, x_n$. Furthermore, suppose that $c_j^{(i)}$ is selected with a probability $c_j^{(i)}$ so that $c_1^{(i)} + c_2^{(i)} + \ldots + c_{l(i)}^{(i)} = 1$. Then the expression level of gene g_i transitions according to the equation:

$$x_i(t+1) = f_j^{(i)}(x(t)) \quad (1)$$

with probability $c_j^{(i)}$. Corresponding to a PBN with n genes, there are

$$N = \prod_{i=1}^{n} l(i)$$

distinct Boolean networks, each of which could capture the inter-gene functional relationships with a certain probability. In accordance with our previous notation, $c_1, c_2, \ldots, c_N$ are the probabilities associated with the selection of each of these networks. Suppose the k^{th} network is obtained by selecting the functional relationship $f_{i_k}^{(i)}$ for gene i, $i = 1, 2, \ldots, n$, $1 \le i_k \le l(i)$. If the choice of the functional relationship for each gene is assumed to be independent of that for other genes, then

$$c_k = \prod_{i=1}^{n} c_{i_k}^{(i)} \quad (2)$$

In this case, the PBN is said to be *independent*. As discussed in [1], even when there is dependence in the choice of the functional relationships for different genes, one can calculate the switching probabilities $c_1, c_2, \ldots, c_N$ by using conditional probabilities instead of the unconditional ones $c_j^{(i)}$.

To characterize the Markov chain associated with an instantaneously random PBN, we first focus on Boolean networks, for which the state vector $x(k)$ at any time step k is essentially an n-digit binary number whose decimal equivalent is given by

$$y(k) = \sum_{j=1}^{n} 2^{n-j} x_j(k) \quad (3)$$

As $x(k)$ ranges from 00...0 to 11…1, $y(k)$ takes on all values from 0 to $2^n - 1$. To be consistent with the development in [1], we define

$$z(k) = 1 + y(k) \quad (4)$$

As $x(k)$ ranges from 00...0 to 11…1, $z(k)$ take on all values from 1 to 2^n. The mapping from $x(k)$ to $z(k)$ is one-to-one and onto, and hence invertible. Thus, instead of the binary representation $x(k)$ for the state vector, we can equivalently work with the decimal representation $z(k)$. Furthermore, each $z(k)$ can be uniquely represented by a basis vector $w(k) \in R^{2^n}$, where $w(k) = e_{z(k)}$, e.g. if $z(k) = 1$, then $w(k) = [1, 0, 0, \ldots..]$. Then, as discussed in [1], the evolution of the vector $w(k)$ proceeds according to the difference equation

$$w(k+1) = w(k)A \quad (5)$$

where A is a $2^n \times 2^n$ matrix having only one non-zero entry (equal to one) in each row. Equation 5 is reminiscent of the state transition equation in Markov Chain theory. The only difference here is that, for a given initial state, the transition is completely deterministic. However, Eq. 5 can also be easily interpreted within a stochastic framework. For instance, the vector $w(k)$ represents the probability distribution over the entire state space at time step k; indeed, owing to the deterministic nature of the evolution, at each time step k, the entire probability mass is concentrated on only one out of the 2^n possible states, thereby accounting for the 2^n-dimensional vectors $w(k)$ with only one non-zero entry 1 corresponding to the location where the probability mass is concentrated. The matrix A is a stochastic matrix with the sole non-zero entry in each row being a 1. Thus, given an initial state, the transition to the next state is deterministic and takes place with probability 1.

The stochastic interpretation of Eq. 5 allows us to readily extend Eq. 5 to accommodate state transitions in probabilistic Boolean networks. Towards this end, let a and b be any two basis vectors in R^{2^n}. Then, using the total probability theorem, it follows that the transition probability $\Pr\{w(k+1) = a \mid w(k) = b\}$ is given by

$$\Pr\{w(k+1) = a \mid w(k) = b\} = \sum_{i=1}^{N} \Pr\{w(k+1) = a \mid w(k) = b, \text{Network } i \text{ is selected}\} \times P_i$$

$$= \sum_{i \in I} P_i \quad (6)$$

where

$$I = \{i : \Pr(w(k+1) = a \mid w(k) = b, \text{Network } i \text{ is selected}) = 1\}.$$

By letting the vectors a and b range over all possible basis vectors in R^{2^n}, we can determine the $2^n \times 2^n$ entries of the transition probability matrix A.

Now let $w(k)$ denote the probability distribution vector at time k, i.e. $w_i(k) = Pr\{z(k) = i\}$. It is straightforward to show that $w(k)$ evolves according to the equation

$$w(k+1) = w(k)A \quad (7)$$

where the entries of the A matrix are determined using Eq. 6. This completes our discussion of PBNs. For a more rigorous derivation of Eq. 7, the reader is referred to [1].

As with the majority of the literature, we have focused on binary quantization; nevertheless, one should recognize that most of the theory and application carry over to any finite quantization in a fairly obvious fashion – that is, to probabilistic gene regulatory networks. A particularly important case of a PGRN arises in the case of ternary quantization, where the expression levels take on the values +1 (up-regulated), −1 (down-regulated), and 0 (invariant); indeed, it is in this ternary setting where we will consider the application of external control. As noted previously, the terminology "probabilistic Boolean network" is usually applied to PGRNs in general, under the supposition that it is the logical character of the networks that is at issue, as in the case of ternary networks.

3. INTERVENTION BY FLIPPING THE STATUS OF A SINGLE GENE

Recognizing that a key goal of PBN modeling is the discovery of possible intervention targets (genes) by which the network can be "persuaded" to transition into a desired

state or set of states, in this section, we consider the effects of intervention by deliberately affecting a particular gene in an instantaneously random PBN. Whereas in Boolean networks, attractors are hypothesized to correspond to functional cellular states [13], in PBNs this role is played by irreducible subchains. Absent the possibility of perturbation ($p = 0$), a PBN is unable to escape from an irreducible subchain, implying that the cellular state cannot be altered. If p is positive, then the Markov chain is ergodic and there is a chance that the current cellular state may switch to another cellular state by means of a random gene perturbation. Clearly, flipping the values of certain genes is more likely to achieve the desired result than flipping the values of some other genes. Our goal is to discover which genes are the best potential "lever points," to borrow the terminology from [13], in the sense of having the greatest possible impact on desired network behavior so that we can intervene with them by changing their value (1 or 0) as needed. In addition, we wish to be able to intervene with as few genes as possible in order to achieve our goals. To motivate the discussion, let us illustrate the idea with an example.

Example 1 [1]. Suppose we are given a PBN consisting of three genes x_1, x_2, x_3. There are two functions $f_1^{(1)}$, $f_2^{(1)}$ associated with x_1, one function associated with x_2 and two functions $f_1^{(3)}$, $f_2^{(3)}$ associated with x_3. These functions are given by the truth table in Table **1**. This truth table results in four possible Boolean networks $N_1 = (f_1^{(1)}, f_1^{(2)}, f_1^{(3)})$, $N_2 = (f_1^{(1)}, f_1^{(2)}, f_2^{(3)})$, $N_3 = (f_2^{(1)}, f_1^{(2)}, f_1^{(3)})$ and $N_4 = (f_2^{(1)}, f_1^{(2)}, f_2^{(3)})$ possessing the probabilities $c_1 = 0.3$, $c_2 = 0.3$, $c_3 = 0.2$ and $c_4 = 0.2$, respectively. The state diagram of the Markov Chain corresponding to this PBN is shown in Fig. **1**. Suppose that we are currently in state (111) and wish to eventually transition to state (000). The question is, with which of the three genes, x_1, x_2, or x_3, should we intervene such that the probability is greatest that we will end up in (000). By direct inspection of the diagram in Fig. **1**, we see that if we make $x_1 = 0$, then with probability 0.2 we will transition into (000), whereas if we make $x_2 = 0$ or $x_3 = 0$, then it will be impossible for us to end up in (000) and with probability 1 we will eventually return to (111). In other words, the network is resistant to perturbations of the second or third genes and will eventually maintain the same state. Thus, the answer to our question in this rather simple example is that only by intervening with gene x_1 do we have a chance of achieving our goal. To answer such questions in general, we need to develop several tools.

Table 1. Truth Table

$x_1x_2x_3$	$f_1^{(1)}$	$f_2^{(1)}$	$f_1^{(2)}$	$f_1^{(3)}$	$f_2^{(3)}$
000	0	0	0	0	0
001	1	1	1	0	0
010	1	1	1	0	0
011	1	0	0	1	0
100	0	0	1	0	0
101	1	1	1	1	0
110	1	1	0	1	0
111	1	1	1	1	1
$c_j^{(i)}$	0.6	0.4	1	0.5	0.5

Assume there is independent random perturbation with $p > 0$, so that the Markov chain is ergodic and every state will eventually be visited. The question of intervention can be posed in the sense of reaching a desired state as soon as possible. For instance, in the example considered above, if p is very small and we are in state (111), then it will be a long

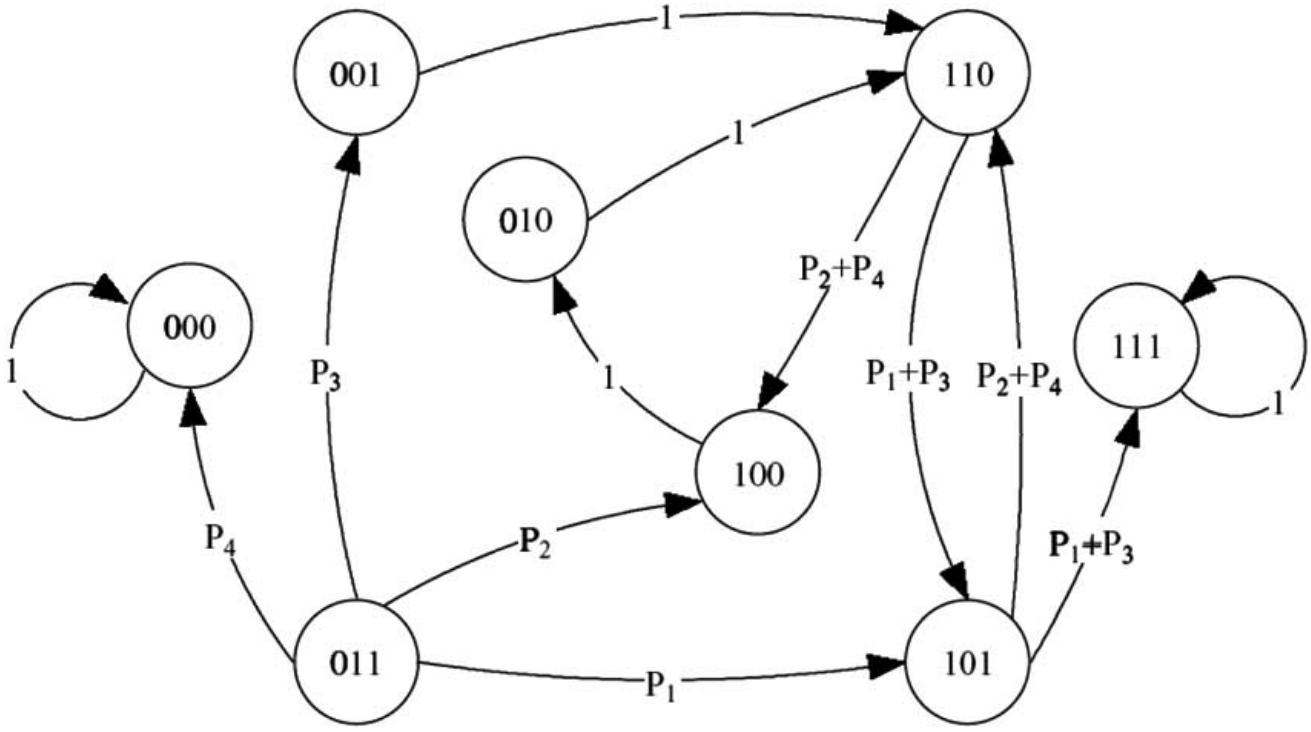

Fig. (1). State Transition Diagram [1].

time until we reach (000) and setting $x_1 = 0$ is much more likely to get us there faster. Hence, we are interested in the probability $F_k(x, y)$ that, starting in state x, the first time the PBN will reach some given state y will be at time k. This is known as the *first passage time* from state x to state y. For $k = 1$, $F_k(x, y) = A(x, y)$, which is just the transition probability from x to y. For $k \geq 2$, it can be shown [14] that

$$F_k(x,y) = \sum_{z \in \{0,1\}^n - \{y\}} A(x,z) F_{k-1}(z,y) \quad (8)$$

We can examine our results by considering

$$H_{K_0}(x,y) = \sum_{k=1}^{K_0} F_k(x,y) \quad (9)$$

which is the probability that the network, starting in state x, will visit state y before time K_0. (Note that the events {the first passage time from x to y will be at time k} are disjoint for different values of k.) As a special case, when $K_0 = \infty$, $H_{K_0}(x,y)$ is the probability that the chain ever visits state y, starting at state x, which is equal to 1 since the Markov chain is ergodic.

A related measure of interest is the *mean first passage time* from state x to state y, defined as

$$M(x,y) = \sum_k k F_k(x,y) \quad (10)$$

$M(x, y)$ is the average time it will take to get from state x to state y.

Example 2 [5]. The entries of the matrix A can be computed directly using the results of [5]. Supposing $p = 0.01$, the steady-state distribution is given by [0.0752, 0.0028, 0.0371, 0.0076, 0.0367, 0.0424, 0.0672, 0.7310], where the leftmost element corresponds to (000) and the rightmost to (111). The PBN spends much more time in state (111) than in any other state. Let our starting state x be (111) and the destination state y be (000), as before. Should we intervene with gene x_1, x_2, or x_3? Using first-passage time, we compute $F_k((011), (000))$, $F_k((101), (000))$, and $F_k((110), (000))$. Fig. **2** shows the plots of $H_{k_0}(x,y)$ for $K_0 = 1, 2, \ldots, 20$ and for the three states of interest, namely, (011), (101), and (110). The plots indicate that starting at state (011), the network is much more likely to enter state (000) sooner than by starting at states (110) or (101). For instance, during the first 20 steps, there is almost a 0.25 probability of entering (000) starting at (011), whereas starting at (110) or (101), there is only a 0.05 probability. Thus, we should intervene with gene x_1 rather than with x_2 or x_3. Were we to base intervention on mean first passage time (Equation 10), then the best gene for intervention would be the one possessing the smallest mean first passage time to the destination state. For this example, the mean first passage times corresponding to the perturbations of genes x_1, x_2, and x_3 are 337.51, 424.14, and 419.20, respectively. Since the first one is the smallest, this again supports the conclusion that gene x_1 is the best candidate for intervention.

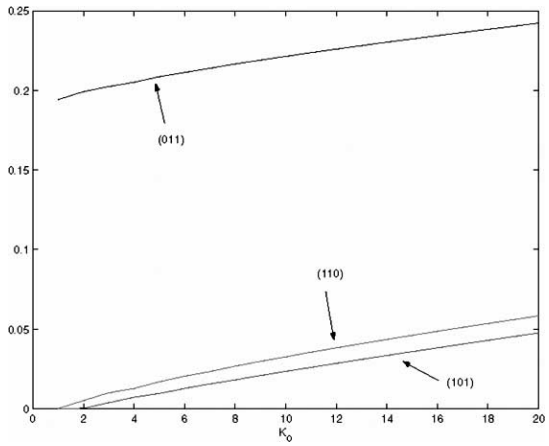

Fig. (2). $H_{K_0}(x^{(i)}, y)$ for $K_0 = 1, \ldots, 20$, for starting states (011), (101), and (110), corresponding to perturbations of first, second, and third genes, respectively [5].

To summarize the results of this section, given an initial state x, we generate different states $x^{(i)} = x \oplus e_i$, $i = 1, 2, \ldots, n$, where e_i is the unit binary vector with a 1 in the i^{th} coordinate, by perturbing each of the n genes, and compute $H_{K_0}(x^{(i)}, y)$ for some desired destination state y and constant K_0. Then, the best gene for intervention is the one for which $H_{K_0}(x^{(i)}, y)$ is maximum; that is, given a fixed K_0, the optimal gene $x_{i_{opt}}$ satisfies

$$i_{opt} = \arg\max_i H_{k_0}(x^{(i)}, y)$$

Alternatively, by minimizing the mean first passage times, the optimal gene satisfies

$$i_{opt} = \arg\min_i M(x^{(i)}, y) .$$

4. INTERVENTION TO ALTER THE STEADY-STATE BEHAVIOR

The type of intervention described in the last section can be useful for modulating the dynamics of the network but it does not alter the underlying network structure. Accordingly, the stationary distribution remains unchanged. However, an imbalance between certain sets of states can be caused by mutations of the "wiring" of certain genes, thereby permanently altering the state-transition structure and, consequently, the long-run behavior of the network [13]. 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 way to accomplish this is by altering some Boolean functions (predictors) in the PBN. An additional goal is to alter as few functions as possible. In [6], formal methods and algorithms have been developed for addressing such a problem. Here we briefly discuss the results.

Consider a PBN with perturbation and two sets of states $A, B \subseteq \{0,1\}^n$. Since the Markov chain is ergodic, each state

$x \subseteq \{0,1\}^n$ has a positive stationary probability $\pi(x)$. Thus, we can define $\pi(A) = \sum_{x \in A} \pi(x)$, and $\pi(B)$ similarly. Suppose that we are interested in altering the stationary probabilities of these two sets of states in such a way that the stationary probability of A is decreased and the stationary probability of B is increased by λ, $0 < \lambda < 1$. As already mentioned above, these two states may represent two different cellular functional states or phenotypes. In order to achieve this, suppose we alter function $f_{j_0}^{(i_0)}$ by replacing it with a new function $g_{j_0}^{(i_0)}$. The probability $c_{j_0}^{(i_0)}$ corresponding to $g_{j_0}^{(i_0)}$ must remain the same as for $f_{j_0}^{(i_0)}$, since $c_1^{(i)} + c_2^{(i)} + \ldots + c_{l(i)}^{(i)} = 1$. Thus, we have a new PBN whose stationary distribution we can denote by μ. Letting $\mu(A)$ and $\mu(B)$ be the stationary probabilities of A and B under the altered PBN model, we pose the following optimization problem:

Given sets A and B, predictor functions $f_j^{(i)}$ together with their selection probabilities $c_j^{(i)}$, $i = 1, 2, \ldots, n$, $j = 1, 2, \ldots, l(i)$, and $\lambda \in (0,1)$, select i_0 and j_0, and a function $g_{j_0}^{(i_0)}$ to replace $f_{j_0}^{(i_0)}$, such that

$$\varepsilon(\pi(A) - \lambda, \mu(A)) \quad (11)$$

and

$$\varepsilon(\pi(B) + \lambda, \mu(B)) \quad (12)$$

are minimum among all i, j, $g_j^{(i)}$.

$\varepsilon(a,b)$ is some error function, such as the absolute error $\varepsilon(a,b) = |a-b|$. An additional constraint can be that $g_{j_0}^{(i_0)}$ has no more essential variables than $f_{j_0}^{(i_0)}$. In this scenario, we are only allowing the alteration of one predictor function. More generally, we can pre-select a number of predictor functions that we are willing to alter.

Example 3 [6]. For the PBN of Example 1, Fig. **1** shows the state transition diagram assuming no perturbation (p = 0). From the figure we see that thee are two absorbing states, (000) and (111). For the sake of this example, suppose (111) corresponds to cell invasion (and rapid proliferation) and (000) corresponds to quiescence. Now assume perturbation probability p = 0.01. A simple analysis based on the probability transition matrix shows that the stationary probabilities of states (000) and (111) are 0.0752 and 0.7310, respectively. Thus, in the long run, the network will be in quiescence only 7% of the time and will be in proliferation 73% of the time. Suppose we wish to alter this imbalance and require the stationary probabilities to be approximately 0.4 for both (000) and (111). The other six states will then be visited only 20% of the time. In the framework of the above optimization problem, $A = \{(111)\}$, $B = \{(000)\}$, $\pi(A) = 0.7310$, $\pi(B) = 0.0752$, $\mu(A) = \mu(B) = 0.4$, and $\lambda = 0.3279$. Finally, suppose we are allowed to change only one predictor function. In Table **1**, this corresponds to changing only one column, while keeping the selection probabilities $c_j^{(i)}$ unchanged. Thus, there are 5 possible columns (predictors) and 256 possibilities for each. The 5 × 256 = 1280 possible alterations have been generated and the stationary probabilities $\mu(000)$ and $\mu(111)$ have been computed for each (see Fig. **3**). The optimal values of $\mu(000)$ and $\mu(111)$ for the error function $\varepsilon(a,b) = |a-b|$ are indicated by an arrow. The objective function to be minimized is $|\mu(000) - 0.4| + |\mu(111) - 0.4|$, which corresponds to the sum of the two objective functions in Eqs. 11 and 12. The colors of the circles represent which predictor is altered. For example, red denotes that predictor $f_1^{(1)}$ is altered. The optimal predictor is the one that alters $f_2^{(1)}$ for gene 1 (column 2 in the truth tables) and the truth table of the optimal predictor is $(00010101)^T$. This predictor achieves the stationary probabilities $\mu(000)$ = 0.4068 and $\mu(111)$ = 0.4128. The structure of the plot in Fig. **3** reveals an interesting phenomenon: the two stationary probabilities exhibit regularities, forming clusters of points arranged in a linear fashion, with different directions. In fact, this phenomenon has been observed in numerous examples. It appears that the alterations of different predictors tend to occupy different parts of the space, implying that for a given predictor, there is a certain "range of action" that can be achieved by manipulating it. This suggests that a brute-force search for the optimal predictor alteration may possibly be avoided by following a number of search directions simultaneously, with the more promising ones being explored further. This, in turn suggests the use of genetic algorithms for optimization [15]. In fact, genetic algorithms have been used to solve the optimal structural intervention problem posed here and the resulting savings in computational effort have been remarkable [6]. Nonetheless, this remains an essentially a brute force procedure and better approaches need to be developed.

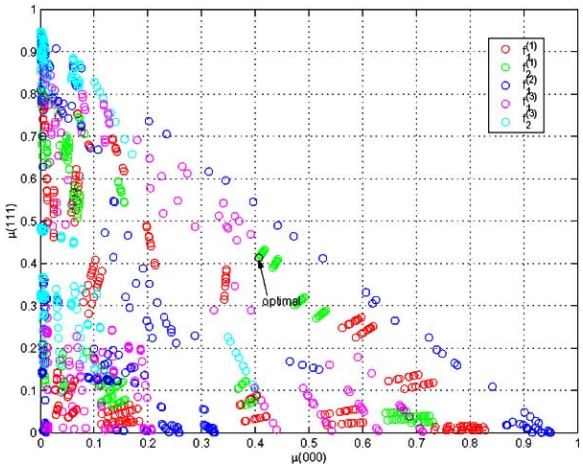

Fig. (3). Each circle represents one of the 1280 possible alterations to the predictors. The x-axis is $\mu(000)$ and the y-axis is $\mu(111)$. The optimal choice is shown with an arrow, as it comes closest to 0.4 for both stationary probabilities. The colors of the circles represent the predictor that is altered (see legend) [6].

5. EXTERNAL INTERVENTION BASED ON OPTIMAL CONTROL THEORY

As discussed in Section 2, probabilistic Boolean networks can be used for studying the dynamic behavior of gene regulatory networks. Once a probability distribution vector has been specified for the initial state, the probability distribution vector evolves according to Eq. 7. From this perspective PBNs are *descriptive* in nature. There is no mechanism for controlling the evolution of the probability distribution vector. For treatment or intervention purposes, we are interested in working with PBNs in a *prescriptive* fashion, where the transition probabilities of the associated Markov chain depend on certain auxiliary variables, whose values can be chosen to make the probability distribution vector evolve in some desirable manner.

The use of such auxiliary variables makes sense from a biological perspective. For instance, in the case of diseases like cancer, auxiliary treatment inputs such as radiation, chemo-therapy, etc. may be employed to move the state probability distribution vector away from one associated with uncontrolled cell proliferation or markedly reduced apoptosis. The auxiliary variables could also include genes that serve as external master-regulators for all the genes in the network. To be consistent with the binary nature of the expression status of individual genes in a PBN, we will assume that the auxiliary variables (*control inputs*) can take on only the binary values 0 or 1. The values of the individual control inputs can be changed from one time step to another in an effort to make the network behave in a desirable fashion.

Suppose that a PBN with n genes has m control inputs u_1, u_2,…, u_m. Then at any given time step k, the row vector

$$u(k) = [u_1(k), u_2(k), ..., u_m(k)] \tag{13}$$

describes the complete status of all the control inputs. $u(k)$ can take on all binary values from $00\cdots0$ to $11\cdots1$. Letting

$$v(k) = 1 + \sum_{i=1}^{m} 2^{m-i} u_i(k) \tag{14}$$

as $u(k)$ takes on binary values from $00\cdots0$ to $11\cdots1$, the variable $v(k)$ ranges from 1 to 2^m. We can equivalently use $v(k)$ as an indicator of the complete control input status of the PBN at time step k.

We now proceed to derive the counterpart of Eq. 7 for a PBN subject to auxiliary controls. Let v^* be any integer between 1 and 2^m and suppose that $v(k) = v^*$. The procedure outlined in Section 2 can be used to compute the corresponding A matrix, which will now depend on v^* and can be denoted by $A(v^*)$. Furthermore, the evolution of the probability distribution vector at time k will take place according to the equation

$$w(k+1) = w(k)A(v^*) \tag{15}$$

Since the choice of v^* is arbitrary, the one-step evolution of the probability distribution vector in the case of a PBN with control inputs takes place according to the equation

$$w(k+1) = w(k)A(v(k)) \tag{16}$$

The transition probability matrix here is a function of all the control inputs $u_1(k)$, $u_2(k)$,…, $u_m(k)$. Consequently, the evolution of the probability distribution vector of the PBN with control now depends not only on the initial distribution vector but also on the values of the control inputs at different time steps. Furthermore, intuitively it appears that it may be possible to make the states of the network evolve in a desirable fashion by appropriately choosing the control input at each time step. We next proceed to formalize these ideas.

Equation 16 is referred to in the control literature as a *controlled Markov chain* or a *Markov decision process* [16]. Markov chains of this type occur in many real life applications, the most notable example being the control of queues. Given a controlled Markov chain, the objective is to find a sequence of control inputs, usually referred to as a *control strategy*, so that an appropriate cost function is minimized over the entire class of allowable control strategies. To arrive at a meaningful solution, the cost function must capture the costs and the benefits of using any control. The design of a "good" cost function is application dependent and is likely to require considerable expert knowledge. We next outline a procedure that we believe would enable us to arrive at a reasonable cost function for determining the course of therapeutic intervention using PBNs.

In the case of diseases like cancer, treatment is typically applied over a finite time horizon. For instance, in the case of radiation treatment, the patient may be treated with radiation over a fixed interval of time following which the treatment is suspended for some time as the effects are evaluated. After that, the treatment may be applied again but the important point to note is that the treatment window at each stage is usually finite. We consider a finite-horizon problem, where the control is applied only over a finite number of steps.

Suppose that the number of steps over which the control input is to be applied has been *a priori* determined to be M and we are interested in controlling the behavior of the PBN over the interval $k = 0, 1, 2,\ldots, M-1$. Suppose at time step k, the state* of the PBN is given by $z(k)$ and the corresponding control input is $v(k)$. Then we can define a cost $C_k(z(k), v(k))$ as being the cost of applying the control input $v(k)$ when the state is $z(k)$. With this definition, the expected cost of control over the entire treatment horizon becomes

$$E\left[\sum_{k=0}^{M-1} C_k(z(k), v(k)) \mid z(0)\right] \tag{17}$$

Note that even if the network starts from a given (deterministic) initial state $z(0)$, the subsequent states will be random because of the stochastic nature of the evolution in Eq. 16. Consequently, the cost in Eq. 17 must be defined using expectation. Equation 17 provides one component of the finite-horizon cost, namely the cost of control. We next introduce the second component.

The net result of the control actions $v(0)$, $v(1)$,…, $v(M-1)$ is that the state of the PBN will transition according to Eq.

* *In the rest of this paper, we will be referring to $z(k)$ as the state of the probabilistic Boolean network since, as discussed in section 2, $z(k)$ is equivalent to the actual state $x(k)$.*

16 and will end up in some state $z(M)$. Owing to the probabilistic nature of the evolution, the terminal state $z(M)$ is a random variable that can possibly take on any of the values 1, 2,…, 2^n. Depending on the particular PBN and the control inputs used at each step, it is possible that some of these states may never be reached because of non-communicating states in the resulting Markov chains; however, since the control strategy itself has not yet been determined, it would be difficult, if not impossible, to identify and exclude such states from further consideration. Instead, we assume that all 2^n terminal states are reachable and assign a penalty, or terminal cost, $C_M(z(M))$ to each of them. We next consider penalty assignment.

First, consider the PBN with all controls set to zero i.e. $v(k) \equiv 1$ for all k. Then divide the states into different categories depending on how desirable or undesirable they are and assign higher terminal costs to the undesirable states. For instance, a state associated with rapid cell proliferation leading to cancer should be associated with a high terminal penalty while a state associated with normal behavior should be assigned a low terminal penalty. For the purposes of this section, we will assume that the assignment of terminal penalties has been carried out and we have at our disposal a terminal penalty $C_M(z(M))$ that is a function of the terminal state. Thus we have arrived at the second component of our cost function. Once again, note that the quantity $C_M(z(M))$ is a random variable and so we must take its expectation while defining the cost function to be minimized. In view of Eq. 17, the finite-horizon cost to be minimized is given by

$$E[\sum_{k=0}^{M-1} C_k(z(k),v(k)) + C_M(z(M)) \mid z(0)] \tag{18}$$

To proceed further, let us assume that at time k the control input $v(k)$ is a function of the current state $z(k)$, namely,

$$v(k) = \mu_k(z(k)) \tag{19}$$

where $\mu_k : \{1,2,....,2^n\} \to \{1,2,...,2^m\}$. The *optimal control problem* can now be stated:

Given an initial state $z(0)$, find a control law $\pi = \{\mu_0, \mu_1, ..., \mu_{M-1}\}$ that minimizes the cost functional

$$J_\pi(z(0)) = E[\sum_{k=0}^{M-1} C_k(z(k),\mu_k(z(k))) + C_M(z(M))] \tag{20}$$

subject to the constraint

$$\Pr\{z(k+1) = j \mid z(k) = i\} = a_{ij}(v(k)) \tag{21}$$

where $a_{ij}(v(k))$ is the i^{th} row, j^{th} column entry of the matrix $A(v(k))$.

Solution Using Dynamic Programming

Optimal control problems of the type described by Eqs. 20 and 21 can be solved using the technique of *Dynamic Programming*. This technique, pioneered by Bellman, in the 1960's is based on the so-called *Principle of Optimality*. This principle is a simple but powerful concept and can be explained as follows. Consider an optimization problem where we are interested in optimizing a performance index over a finite number, M, of steps. At each step, a decision is made and the objective is to devise a strategy or *sequence* of M decisions that is optimal in the sense that the cumulative performance index over all the M steps is optimized. In general, such an optimal strategy may not exist. However, when such an optimal strategy does exist, the principle of optimality asserts: if one searches for an optimal strategy over a subset of the original number of steps, then this new optimal strategy will be given by the overall optimal strategy, restricted to the steps being considered. Although intuitively obvious, the principle of optimality can have far reaching consequences. For instance, it can be used to obtain the following proposition [16].

Proposition 1. Let $J^*(z(0))$ be the optimal value of the cost functional in Eq. 20. Then

$$J^*(z(0)) = J_0(z(0)),$$

where the function J_0 is given by the last step of the following dynamic programming algorithm which proceeds backward in time from time step $M-1$ to time step 0:

$$J_M(z(M)) = C_M(z(M)) \tag{22}$$

$$J_k(z(k)) = \min_{v(k)\in\{1,2,...2^m\}} E\{C_k(z(k),v(k)) + J_{k+1}[z(k+1)]\} \tag{23}$$

for $k = 0, 1,\ldots, M-1$. Furthermore, if $v^*(k) = \mu_k^*(z(k))$ minimizes the right hand side of (23) for each $z(k)$ and k, then the control law $\pi^* = \{\mu_0^*, \mu_1^*, \ldots, \mu_{N-1}^*\}$ is optimal.

Note that the expectation on the right hand side of Eq. 23 is conditioned on $z(k)$ and $v(k)$. Hence, in view of Eq. 21, it follows that

$$E[J_{k+1}(z(k+1)) \mid z(k),v(k)] = \sum_{j=1}^{2^n} a_{z(k),j}(v(k)) J_{k+1}(j).$$

Thus the dynamic programming solution to Eqs. 20 and 21 is given by

$$J_M(z(M)) = C_M(z(M)) \tag{24}$$

$$J_k(z(k)) = \min_{v(k)\in\{1,2,...2^m\}} \{C_k(z(k),v(k)) + \sum_{j=1}^{2^n} a_{z(k),j}(v(k)) J_{k+1}(j)\} \tag{25}$$

for $k = 0, 1,\ldots, M-1$.

We next present two extensive examples to show optimal control design using the dynamic programming approach. The first is contrived for illustrative purposes only while the second is a realistic and based on actual gene expression data.

A Simple Illustrative Example

We consider an example of a PBN with control and work through the details to show how Eqs. 24 and 25 can be used to arrive at an optimal control strategy. The example is adapted from the one used in the last two sections and involves the truth table in Table **1**, which corresponds to an uncontrolled PBN. To introduce control, let us assume that x_1 is now going to be a control input whose value can be externally switched between 0 and 1 and the states of the new PBN are x_2 and x_3. To be consistent with the notation introduced in this section, the variables x_1, x_2 and x_3 will be renamed; the variable x_1 now becomes u_1 while the variables

x_2 and x_3 become x_1 and x_2 respectively. With this change, we have the truth table shown in Table **2**, which also contains the values of the variables v and z corresponding to u_1 and x_1x_2, respectively. The values of $c_j^{(i)}$ in the table dictate that there are two possible networks, the first corresponding to the choice of functions ($f_1^{(1)}$, $f_1^{(2)}$) and the second corresponding to the choice of functions ($f_1^{(1)}$, $f_2^{(2)}$). The probabilities c_1 and c_2 associated with each of these networks is given by $c_1 = c_2 = 0.5$. We next proceed to compute the matrices $A(1)$ and $A(2)$ corresponding to the two possible values for v. From Table **2**, it is clear that when $v = 1$, the following transitions are associated with the network N_1 and occur with probability c_1:

$$z = 1 \rightarrow z = 1,\ z = 2 \rightarrow z = 3,\ z = 3 \rightarrow z = 3,\ z = 4 \rightarrow z = 2 \quad (26)$$

The corresponding transitions associated with network N_2 that occur with probability c_2 are given by:

$$z = 1 \rightarrow z = 1,\ z = 2 \rightarrow z = 3,\ z = 3 \rightarrow z = 3,\ z = 4 \rightarrow z = 1 \quad (27)$$

In view of Eqs. 26 and 27, the matrices $A(1)$ and $A(2)$ are given by

$$A(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ P_2 & P_1 & 0 & 0 \end{bmatrix} \quad (28)$$

$$A(2) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & P_2 & P_1 \\ P_2 & P_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (29)$$

Table 2. Truth Table for the Example of this Section

u_1	v	x_1	x_2	z	$f_1^{(1)}$	$f_1^{(2)}$	$f_2^{(2)}$
0	1	0	0	1	0	0	0
0	1	0	1	2	1	0	0
0	1	1	0	3	1	0	0
0	1	1	1	4	0	1	0
1	2	0	0	1	1	0	0
1	2	0	1	2	1	1	0
1	2	1	0	3	0	1	0
1	2	1	1	4	1	1	1
c_j^i					1	0.5	0.5

In this example, $n = 2$ so that the variable z can take on any one of the four values 1, 2, 3, or 4. Since $m = 1$, the control variable v can take on any one of the two values 1 or 2. Suppose that the control action is to be carried out over 5 steps, so that $M = 5$. Moreover, assume that the terminal penalties are given by

$$C_5(1) = 0,\ C_5(2) = 1,\ C_5(3) = 2,\ C_5(4) = 3 \quad (30)$$

Note that the choices of M and the values of the terminal penalties are completely arbitrary; in a real-world example, this information would be obtained from biologists. The current choice of terminal penalties indicates that the most desirable terminal state is 1 and the least desirable terminal state is 4. For the optimization problem of Eqs. 20 and 21, we need to define the function $C_k(z(k), v(k))$. For the sake of simplicity, let us define

$$C_k(z(k), v(k)) = \sum_{i=1}^{m} u_i(k) = u_1(k) \quad (31)$$

where $v(k)$ and $u_i(k)$, $i = 1, 2,\ldots, m$ are related by Eq. 14. The cost $C_k(z(k), v(k))$ captures the cost of applying the input $u_1(k)$ at the k^{th} step. The optimization problem of Eqs. 20 and 21 can now be posed using the quantities defined in Eqs. 28, 29, 30, 31. The dynamic programming algorithm resulting from Eqs. 24 and 25 becomes

$$J_5(z(5)) = C_5(z(5)) \quad (32)$$

$$J_k(z(k)) = \min_{v(k)\in\{1,2\}} [u_1(k) + \sum_{j=1}^{4} a_{z(k),j}(v(k)) J_{k+1}(j)]\ ,\ k = 0,\ 1,\ 2,\ 3,\ 4 \quad (33)$$

We proceed backwards step by step from $k = 4$ to obtain a solution to Eqs. 32 and 33. The resulting optimal control strategy for this finite horizon problem is:

$$\mu_0^*(z(0)) = \mu_1^*(z(1)) = \mu_2^*(z(2)) = \mu_3^*(z(3)) = 1 \text{ for all } z(0), z(1), z(2), z(3) \quad (34)$$

$$\mu_4^*(z(4)) = \begin{cases} 2 & \text{if } z(4) = 3 \\ 1 & \text{otherwise.} \end{cases} \quad (35)$$

Thus the control input is applied only in the last time step, provided the state z of the system at that time step is equal to 3; otherwise, the optimal control strategy is to not apply any control at all. Let us now consider a few different initial states $z(0)$ and see whether this optimal control strategy makes intuitive sense.

Case 1. $z(0) = 1$: According to Eqs. 34 and 35, the optimal control strategy in this case is no control. Note from Eq. 30 that the evolution of the PBN is starting from the most desirable terminal state. Furthermore, from Eq. 28, it is clear that in the absence of any control, the state of the network remains at this position. Hence, the control strategy arrived at is, indeed, optimal and the value of the optimal cost is 0.

Case 2. $z(0) = 4$: In this case, from Eq. 30, it is clear that the evolution of the PBN is starting from the most undesirable terminal state. Moreover, from Eq. 29, note that if the control input were kept turned ON over the entire control horizon, then the state would continue to remain in this most undesirable position during the entire control duration. Such a control strategy cannot be optimal since not only does the network end up in the most undesirable terminal state but also the maximum possible control cost is incurred over the entire time horizon.

To get a more concrete feel for the optimal control strategy, let us focus on the cases where the PBN degenerates into a standard (deterministic) Boolean network. There are two cases to consider:

(i) $c_2 = 1$, $c_1 = 0$: In this case, from Eq. 28 we have

$$A(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (36)$$

Clearly, if no control is employed then, starting from $z(0) = 4$, the network will reach the state $z(1) = 1$ in one step and stay there forever. Thus, this no-control strategy is optimal and the optimal cost is 0.

$c_2 = 0$, $c_1 = 1$: In this case, from Eqs. 28 and 29 we have

$$A(1) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, A(2) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (37)$$

From Eq. 34 the optimal control strategy is no control over the first four time steps. From Eq. 37 it follows that, with $z(0) = 4$, we will have $z(1) = 2$, $z(2) = 3$, $z(3) = 3$ and $z(4) = 3$. Then at the last time step, the control input is turned ON and from Eq. 37 the resulting state is $z(5) = 2$. The optimal cost is 2 (the sum of the terminal cost and the cost of control).

Melanoma Example

We now apply the methodology of this section to derive an optimal intervention strategy for a particular gene regulatory network. The network chosen as an example of how control might be applied is one developed from data collected in a study of metastatic melanoma [17]. In this expression profiling study, the abundance of messenger RNA for the gene WNT5A was found to be a highly discriminating difference between cells with properties typically associated with high metastatic competence versus those with low metastatic competence. These findings were validated and expanded in a second study [18]. In this study, experimentally increasing the levels of the Wnt5a protein secreted by a melanoma cell line *via* genetic engineering methods directly altered the metastatic competence of that cell as measured by the standard *in vitro* assays for metastasis. A further finding of interest in the current study was that an intervention that blocked the Wnt5a protein from activating its receptor, the use of an antibody that binds Wnt5a protein, could substantially reduce Wnt5a's ability to induce a metastatic phenotype. This suggests a study of control based on interventions that alter the contribution of the WNT5A gene's action to biological regulation, since the available data suggest that disruption of this influence could reduce the chance of a melanoma metastasizing, a desirable outcome.

The methods for choosing the genes involved in a small local network that includes the activity of the WNT5A gene and the rules of interaction have been described in [19]. As discussed in that paper, the WNT5A network is obtained by studying the predictive relationship between 587 genes. The expression status of each gene is quantized to one of three possible levels: –1 (down-regulated), 0 (unchanged) and 1 (upregulated). In this case, the gene activity profile at any time step is ternary, not binary; nonetheless, the PBN formulation and the associated control strategy can be developed exactly as described, the only difference being that now, for an n-gene network, we will have 3^n states instead of 2^n states. In this context, it is appropriate to point out that to apply the control algorithm of this paper, it is not necessary to actually construct a PBN; all that is required are the transition probabilities between the different states under the different controls.

A ternary network with 587 genes will have 3^{587} states, which is an intractably large number to use either for modeling or for control. Consequently, the number of genes has been reduced to the ten most significant ones and the resulting multivariate relationships, using the best three-gene predictor for each gene, are shown in Fig. **4**. These relationships were developed using the CoD (coefficient of determination) technique [20, 21, 22] applied to the gene-expression patterns across 31 different conditions and prior biological knowledge (a detailed description being given in [19]).

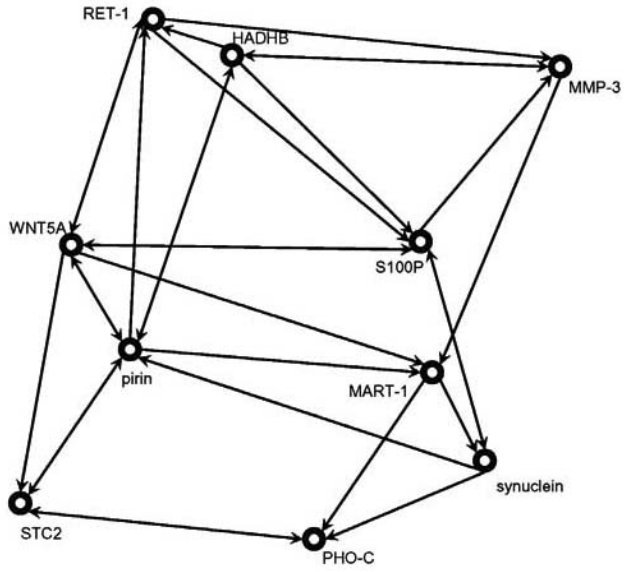

Fig. (4). Multivariate relationship between the genes of the 10-gene WNT5A network [26].

Because it is biologically known that WNT5A ceasing to be down-regulated is strongly predictive of the onset of metastasis, the control objective for this 10-gene network is to externally down-regulate theWNT5A gene. Controlling the 10-gene network using dynamic programming would require designing a control algorithm for a system with 3^{10} (59,049) states. Although there is nothing conceptually difficult about doing this, it is beyond the computational limits of our current software. Accordingly, we have further narrowed down the number of genes in the network to 7 by using CoD analysis on the 31 samples. The resulting genes, along with their multivariate relationships, are shown in Fig. **5**. For each gene in this network, we have determined their two best two-gene predictors and their corresponding CoD's. Using the procedure discussed in [1], the CoD information for each of the predictors has then been used to determine the $3^7 x 3^7$ matrix of transition probabilities for the Markov chain corresponding to the dynamic evolution of the gene-activity profile of the seven gene network.

The optimal control problem can now be completely specified by choosing (i) the treatment/ intervention window, (ii) the terminal penalty and (iii) the types of controls and the costs associated with them. For the treatment window, we arbitrarily choose a window of length 5, i.e. control inputs would be applied only at time steps 0, 1, 2, 3 and 4. The

terminal penalty at time step 5 is chosen as follows. Since our objective is to ensure that WNT5A is down regulated, we assigned a penalty of 0 to all states for which WNT5A equals −1, a penalty of 3 to all states for which WNT5A equals 0 and a penalty of 6 to all states for which WNT5A equals 1. Here the choice of the numbers 3 and 6 is arbitrary but it does reflect our attempt to capture the intuitive notion that states where WNT5A equals 1 are less desirable than those where WNT5A equals 0. Two types of possible controls are used and we discuss the two cases separately.

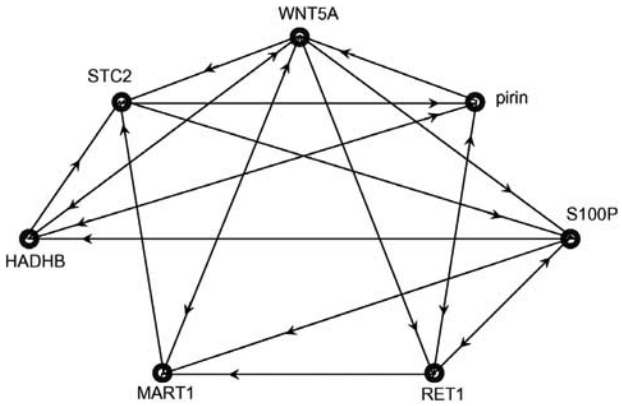

Fig. (5). Multivariate relationships between the genes of the 7-gene WNT5A network [7].

Case 1. WNT5A Controlled Directly: In this case, the control action at any given time step is to force WNT5A equal to −1, if necessary, and let the network evolve from there. Biologically such a control could be implemented by using a WNT5A inhibitory protein. In this case, the control variable is binary with 0 indicating that the expression status of WNT5A has not been *forcibly* altered while 1 indicates that such a forcible alteration has taken place. Of course, whether at a given time step, such intervention takes place or not is decided by the solution to the resulting dynamic programming algorithm and the actual state of the network immediately prior to the intervention. With this kind of intervention strategy, it seems reasonable to incur a control cost at a given time step if and only if the expression status of WNT5A has to be forcibly changed at that time step. Once again, we arbitrarily assign a cost of 1 to each such forcible change and solve for the optimal control using dynamic programming. The net result is a set of optimal control inputs for each of the 2187 (3^7) states at each of the five time points. Using these control inputs, we have studied the evolution of the state probability distribution vector with and without control. For every possible initial state, our simulations indicate that at every time step from 1 to 5, the probability of WNT5A being equal to −1 is higher with control than that without control. Furthermore, with control, WNT5A always reaches −1 at the final time point ($k = 5$). Thus, we conclude that the optimal control strategy of this section is successful in achieving the desired control objective. In this context, it is significant to point out that if the network starts from the initial state STC2 = −1, HADHB = 0, MART-1 = 0, RET-1 = 0, S100P = −1, pirin = 1, WNT5A = 1 and if no control is used, then it quickly transitions to a *bad absorbing* state (absorbing state with WNT5A = 1). With optimal control, however, this does not happen.

Case 2. WNT5A Controlled Through pirin: In this case, the control objective is the same as in Case 1, namely to keep WNT5A down-regulated. The only difference is that this time we use another gene, pirin, to achieve this control. The treatment window and the terminal penalties are kept exactly the same as before. The control action consists of either forcing pirin to −1 (corresponding to a control input of 1) or letting it remain wherever it is (corresponding to a control input of 0). As before, at any step, a control cost of 1 is incurred if and only if pirin is forcibly reset to −1 at that time step. Having chosen these design parameters, we implement the dynamic programming algorithm with pirin as the control. Using the resulting optimal controls, we have studied the evolution of the state probability distribution vector with and without control. For every possible initial state, our simulations indicate that, at the final state, the probability of WNT5A being equal to −1 is higher with control than that without control. In this case, however, there is no definite ordering of probabilities between the controlled and uncontrolled cases at the intermediate time points. Moreover, the probability of WNT5A being equal to −1 at the final time point is not, in general, equal to 1. This is not surprising given that now we are trying to control the expression status of WNT5A using another gene and the control horizon of length 5 simply may not be adequate for achieving the desired objective with such a high probability. Nevertheless, even in this case, if the network starts from the state corresponding to STC2 = −1, HADHB = 0, MART-1 = 0, RET-1 = 0, S100P = −1, pirin = 1, WNT5A = 1 and evolves under optimal control, then the probability of WNT5A = −1 at the final time point equals 0.673521. This is quite good in view of the fact that the same probability would have been equal to 0 in the absence of any control action.

6. EXTERNAL INTERVENTION IN THE IMPERFECT INFORMATION CASE

The control law that emerges from the solution of the dynamic programming problem of Eqs. 24 and 25 takes the form of a state feedback [†]

$$v_k = \mu_k(z_k),\ k = 0, 1, 2, \ldots, M-1 \tag{38}$$

When the state vector z_k of the PBN is not available for measurement, such a control law cannot be implemented. In that case, we will assume that when the PBN is in the state z_k, it emits q *measurable* outputs, each of which could take on the value 0 or 1. Thus, the output status of the PBN at any time k can be captured by a q-digit binary number or, alternatively, by its decimal equivalent plus one, which we shall call θ_k. As the outputs range over all possible binary values, θ_k takes on all values from 1 to 2^q.

The design of the optimal control in this case can make use of only the signals available to the controller. In other words, at time k, the controller tries to design the control input v_k using all the available signals, $\theta_0, \theta_1, \ldots, \theta_k, v_0, v_1, \ldots, v_{k-1}$. Although the state z_k evolves according to Eq. 21 and is not available for measurement, we assume that the output θ_k at time k is probabilistically related to the state z_k at time k

[†] In the rest of this paper, we will be denoting $w(k)$, $z(k)$, $v(k)$ by w_k, z_k, v_k respectively, mainly for the purpose of simplifying the notation.

and the input v_{k-1} through the known conditional probability measure $\Pr_{\theta_k}(.|z_k, v_{k-1})$ defined by

$$\Pr\{\theta_k = j \mid z_k = i, v_{k-1} = v\} = r_{ij}^{v} \quad (39)$$

The total information available for control at time k is given by $I_k = [\theta_0, v_0, \theta_1, v_1, \ldots, v_{k-1}, \theta_k]^T$. I_k can be generated recursively using the equation

$$I_{k+1} = [I_k^T, v_k, \theta_{k+1}]^T, \qquad I_0 = \theta_0 \quad (40)$$

Since the state z_k is not available, it seems reasonable to replace the state feedback control of Eq. 38 by the information feedback control

$$v_k = \mu_k(I_k),\ k = 0, 1, 2, \ldots, M-1 \quad (41)$$

and search for the optimal μ_k over the space of all functions μ_k mapping the space of information vectors I_k into the control space $\{1, 2, 3, \ldots, 2^m\}$. Thus the counterpart to the optimization problem of Eqs. 20 and 21 for this case becomes [7, 23]

$$\min_{\mu_0, \mu_1, \ldots, \mu_{M-1}} E_{z_0, d_0, d_1, \ldots, d_{M-1}, \theta_0, \theta_1, \ldots, \theta_{M-1}} \{\sum_{k=0}^{M-1} C_k(z_k, \mu_k(I_k), d_k) + C_M(z_M)\} \quad (42)$$

subject to

$$z_{k+1} = d_k, \quad (43)$$

$$\Pr\{d_k = j \mid z_k = i, v_k\} = a_{ij}(v_k), \quad (44)$$

$$I_{k+1} = [I_k^T, v_k, \theta_{k+1}]^T, \quad I_0 = \theta_0 \quad (45)$$

The dynamic programming algorithm for the above problem is given by [7, 23]

$$J_{M-1}(I_{M-1}) = \min_{v_{m-1} \in \{1,2,\ldots,2^m\}} \{E_{z_{M-1}, d_{M-1}}[C_M(d_{M-1}) + C_{M-1}(z_{M-1}, v_{M-1}, d_{M-1}) | I_{M-1}, v_{M-1}]\} \quad (46)$$

$$J_k(I_k) = \min_{v_k \in \{1,2,\ldots,2^m\}} \{E_{\theta_{k+1}, z_k, d_k}[\{C_k(z_k, v_k, d_k) + J_{k+1}([I_k^T, v_k, \theta_{k+1}]^T)\} | I_k, v_k]\} \quad (47)$$

for $k = 0, 1, 2, \ldots, M-2$, and the optimal control input is obtained from the values minimizing the right-hand side of Eqs. 46 and 47. Using this algorithm, we will ultimately arrive at $J_0(I_0) = J_0(\theta_0)$. The optimal cost J^* can be obtained by taking the expectation of this quantity with respect to θ_0, i.e.

$$J^* = E_{\theta_0}[J_0(\theta_0)] \quad (48)$$

Melanoma Example

Consider a 7-gene network which is a slight variation of the one considered in the last section. Since implementing the imperfect information based control is computationally more intensive compared to the perfect information case, we have developed a binary 7-gene network using CoD analysis on the same experimental data. The resulting genes along with their multivariate relationship are shown in Fig. **5**. For each gene in this network, we have determined their two best two-gene predictors and their corresponding CoDs. Using the procedure discussed in [1], the CoD information for each of the predictors is used to determine the 2^7 x 2^7 matrix of transition probabilities for the Markov chain corresponding to the dynamic evolution of the GAP of the 7-gene network. The transition probability matrix $A(v(k))$, the probability distribution of the observations given the current state and the immediately prior control r_{ij}^{v}, and the initial state probability distribution vector together constitute the data needed for the optimal control problem with imperfect state information. In our construction, the vector r_{ij}^{v} does not depend on the prior control input v and probabilistically relates the observation to the current state of the network. This relationship is shown in Fig. **6** and it closely mimics the behavior of a gene MMP-3 that appears in the 10-gene network of Fig. **4** but does not appear in the 7-gene network of Fig. **5**.

The optimal control problem is completely specified by choosing (i) the treatment/intervention window, (ii) the terminal penalty and (iii) the types of controls and the costs associated with them. For the treatment window, we arbitrarily choose a window of length 5, i.e. time steps 0, 1, 2, 3 and 4. Based upon the same reasoning as in the full-information case, the terminal penalty at time step 5 is chosen as 0 for all states for which WNT5A equals 0 and 3 for all states for which WNT5A equals 1. We now discuss two possible types of control actions for various initial state probability distributions.

Case 1. WNT5A Controlled Directly: In this case, the control action at any given time step is to force WNT5A equal to 0, if necessary, and let the network evolve from there. The control variable is binary with 1 and 0 indicating intervention and no intervention, respectively. The one-step cost of control is taken to be the value of the control variable. Whether at a given time step intervention takes place is decided by the solution to the resulting dynamic programming algorithm depending on the initial distribution and the subsequent total information vector I_k. Note that unlike the perfect information scenario considered in the last section, we are now not in a position to determine if forcible alteration of the state takes place or not. Consequently, it is reasonable to expect that WNT5A inhibition may be used, even when not absolutely necessary, thereby contributing to a possible increase in the total optimal expected cost, compared to the perfect information case. We recursively use Eqs. 46 and 47 to calculate the optimal controls for certain initial state probability distributions. The net result, in each case, is a tree of control actions corresponding to each control action and subsequent observation. Starting with P_{data}, the distribution of states in the 31 point data set, we find the optimal expected cost based on imperfect information to be 0.4079. The corresponding optimal cost using full state observation as in the last section is 0.3226. The expected cost incurred by not using any control is 0.9677. We have computed these quantities for a few different cases of initial state distributions. The relevant quantities are tabulated in Table **3**.

We have also calculated the optimal expected costs when the initial state is deterministic. These values for all the 128 possible initial states are shown in Fig. **7**. As expected, the optimal cost for control with imperfect information is higher than that for control with perfect state information. The cost function, however, is a somewhat subjective quantity chosen by us to mathematically capture the underlying biological objective. A more natural way to look at the performance of the control scheme would be to examine the probability of WNT5A being equal to 0 at the final time step, i.e. at $k = 5$.

This quantity has been computed for each (deterministic) initial state for both the uncontrolled and imperfect-information-based controlled cases. These plots are shown in Fig. **8**. From this figure, it is clear that the control strategy for each initial state is increasing the probability for WNT5A being equal to 0 at the terminal time point relative to the corresponding probability in the uncontrolled case. This is a desirable outcome achieved by using control.

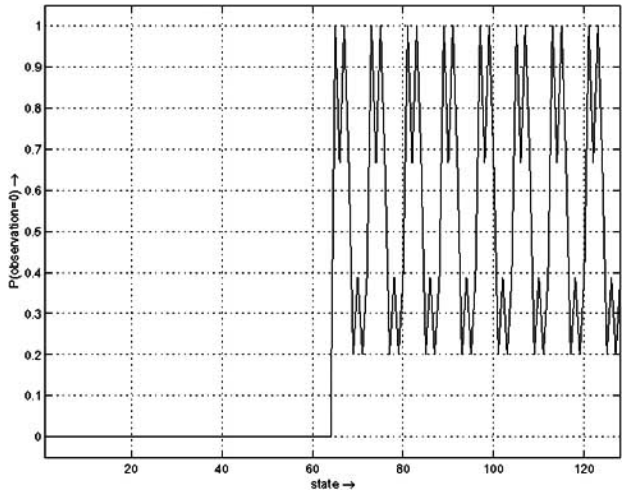

Fig. (6). Probability (Observed Variable =0) Versus Current State [8].

Table 3. Expected Costs for Various Initial State Distributions

Initial Distribution	Control using Observation	Full State	No Control
$P_{sample\text{-}data}$	0.4079	0.3226	0.9677
$[\frac{1}{128},\frac{1}{128},.....]$	0.7068	0.3395	0.9990
$[0,\frac{1}{64},0,\frac{1}{64},.....]$	0.7296	0.3395	0.9990
$[\frac{1}{64},0,\frac{1}{64},0,.....]$	0.5692	0.3395	0.9990

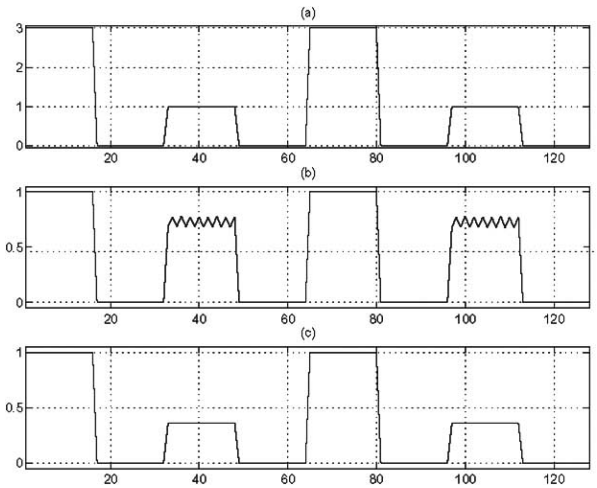

Fig. (7). Optimal expected cost versus initial states **(a)** uncontrolled, **(b)** control using imperfect information, **(c)** control using full state information [8].

Case 2. WNT5A Controlled Through pirin: in this case, the control objective is the same as in Case 1, namely to keep WNT5A at 0; however, now use pirin, to achieve control. The treatment window and the terminal penalties are kept exactly the same as before. The control action consists of either using a pirin inhibitor (corresponding to a control input of 1) or not employing such an inhibitor (corresponding to a control input of 0). The one-step cost of control is taken to be equal to the value of the control variable. As before, at any step, whether such intervention takes place or not is decided by the solution to the resulting dynamic programming algorithm. Having chosen these design parameters, we implement the algorithm with pirin as the control. We find that using pirin as a control is totally ineffective. The expected cost with pirin as the control is found to be the same as the one obtained in Table **3** with no control. Even with full state feedback we still find that pirin is similarly ineffective (data not shown). This is in stark contrast to our results in the last section where we have demonstrated the feasibility of doing full state feedback control of WNT5A through pirin. It is possible that going from a ternary set-up in the last section to the binary setup here may have drastically reduced our ability to control WNT5A through pirin. This suggests that the standard control theoretic notions of controllability and observability [25] may have to be revisited in the context of genetic regulatory networks to enable us to decide which genes can be used as effective controls and which ones can be used as meaningful observations.

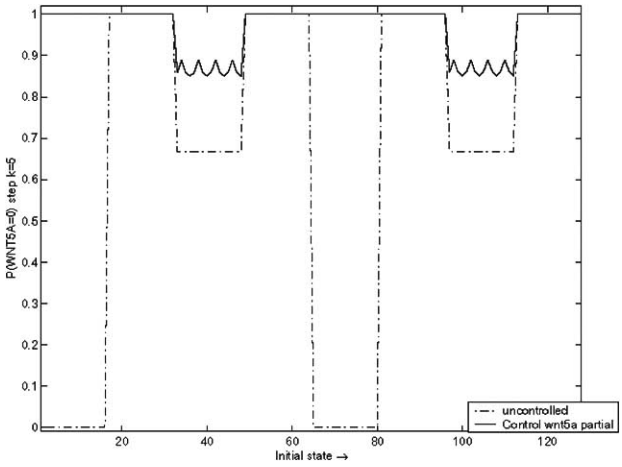

Fig. (8). Probability of WNT5A= 0 at the terminal time point versus the initial state for the uncontrolled and imperfect-information-based controlled cases [8].

7. EXTERNAL INTERVENTION IN THE CONTEXT SENSITIVE CASE

This section extends the results of Section 5 to *context-sensitive* PBNs with perturbation. The intervention results from Section 5 carry over to this case and the only difference is that the entries of the transition probability matrix have to be derived differently. Since there are n genes, the probability of there being a random perturbation at any time point is $1 - (1 - p)^n$. . For a context-sensitive PBN, the state $z(t)$ at time t could be originating from any one of the k possible networks. To keep track of the network emitting a

particular state, let us redefine the states by incorporating the network number inside the state label. Since we have k different Boolean networks forming the PBN, the total number of states becomes $2^n k$ and we label these states as $S_1, S_2, ..., S_{2^n k}$, where for each r = 1, 2, . . . , k, states $S_{2^n(r-1)+1}, S_{2^n(r-1)+2}, ..., S_{2^n r}$ belong to network r. Equivalently $S_{2^n(r-1)+i}$ corresponds to z_{r_i}, where z_{r_i} is the decimal representation of the i^{th} state in the network r. Denote the redefined state at time t by $w(t)$. We need to derive the transition probability expressions for the uncontrolled and controlled cases. First we treat the uncontrolled case.

Recall from Section 2 that for context-sensitive PBNs, one of the following events occurs at each time point t: (1) the current network function is applied, the PBN transitions accordingly, and the network function remains the same for the next transition; (2) the current network function is applied, the PBN transitions accordingly, and a new network function is selected for the next transition; (3) there is a random perturbation and the network function remains the same for the next transition; or (4) there is a random perturbation and a new network function is selected for the next transition. Assuming that the individual genes perturb independently, and letting $\mathrm{mod}(v, w)$ denote the remainder left over when v is divided by w, we consider two cases for determining the transition probability of going from state a to state b:

Case 1: $[(a-1)/2^n] = [(b-1)/2^n]$, meaning $2^n(r-1)+1 \le a, b \le 2^n r$ for the same r. This corresponds to the events (1) and (3) above and the transition probabilities are given by

$$\Pr(w(t+1)=b \mid w(t)=a) = (1-q)(1-p)^n f_{r,a,b} + (1-q)(1-p)^{n-h} p^h s(h) \quad (49)$$

where h is the Hamming Distance between $\mathrm{mod}(a-1, 2^n)$ and $\mathrm{mod}(b-1, 2^n)$, i.e. the number of genes which differ between the two states,

$$f_{r,a,b} = \begin{cases} 1, & \text{if } a \text{ transitions to } b \text{ in a single step in network } r, \\ 0, & \text{otherwise} \end{cases}$$

and

$$s(h) = \begin{cases} 0, & \text{if h} = 0, \\ 1, & \text{otherwise.} \end{cases}$$

The first term in Eq. 49 corresponds to event (1) above, where $1-q$ is the probability that the network selection does not change, $(1-p)^n$ is the probability that none of the n genes undergoes a perturbation, we assume that network selection and random gene perturbation are independent events, and $f_{r,a,b} = 1$ if that particular transition is possible in the r^{th} Boolean network. The second term corresponds to event (3), where h genes have to be perturbed to go from state a to state b.

Case 2: $2^n(r_1-1)+1 \le a \le 2^n r_1$ and $2^n(r_2-1)+1 \le b \le 2^n r_2$, where $r_1 \ne r_2$. This corresponds to the events (2) and (4) above and the transition probabilities are given by

$$\Pr(w(t+1)=b \mid w(t)=a) = qc_{r_2}\left(\sum_{i=1, i\ne r_1}^{k} c_i\right)^{-1}\left[(1-p)^n f_{r_1,a,b} + (1-p)^{n-h} p^h s(h)\right] \quad (50)$$

If we define

$$g(a,b) = \begin{cases} 1, & \text{if } [(a-1)/2^n] - [(b-1)/2^n] = 0 \\ 0, & \text{otherwise} \end{cases}$$

then a unified transition probability expression encompassing the two cases is given by

$$\Pr(w(t+1)=b \mid w(t)=a) =$$
$$[(1-q)(1-p)^n f_{r,a,b} + (1-q)(1-p)^{n-h} p^h s(h)]g(a,b) +$$
$$\left\lceil qc_{r_2}\left(\sum_{i=1, i\ne r_1}^{k} c_i\right)^{-1}\left\lceil (1-p)^n f_{r_1,a,b} + (1-p)^{n-h} p^h s(h)\right\rfloor\right\rfloor \times [1-g(a,b)] \quad (51)$$

By letting a and b range over all integers from 1 to $2^n k$ and using Eq. 51, we can determine all the entries of the $2^n k$ x $2^n k$ matrix of transition probabilities.

In practice, it will likely be impossible to detect the Boolean network from which the current gene activity profile is being emitted. In most cases, we will only have knowledge of the states of the genes. To handle such situations, we can derive an expression for the transition probability from state s_2 to state s_1, where these states run from 1 to 2^n and reflect only the expression status of the n-gene state vector:

$$\Pr[z(t+1)=s_1 \mid z(t)=s_2] = \sum_{i=1}^{k} \Pr[z(t+1)=s_1, s_2 \text{ belongs to network } i \mid z(t)=s_2]$$
$$= \sum_{i=1}^{k} \Pr[z(t+1)=s_1 \mid z(t)=s_2, s_2 \text{ belongs to network } i] \times \Pr[s_2 \text{ belongs to network } i]$$
$$= \sum_{i=1}^{k} \Pr[z(t+1)=s_1 \mid w(t)=s_2 + 2^n(i-1)].c_i$$
$$= \sum_{i=1}^{k}\sum_{j=1}^{k} c_i . \Pr[w(t+1)=s_1 + 2^n(j-1) \mid w(t)=s_2 + 2^n(i-1)] \quad (52)$$

Note that here state s_1 is equivalent to the distinct states s_1, $s_1 + 2^n, \ldots, s_1 + (k-1)2^n$ in the previous $2^n k$ formulation. Similarly s_2 here is equivalent to $s_2, s_2 + 2^n, \ldots, s_2 + (k-1)2^n$ in the earlier formulation. By letting s_1 and s_2 range from 1 to 2^n and using Eq. 52, we can derive the 2^n x 2^n transition probability matrix A corresponding to the context-sensitive PBN.

If a control action is applied, then the transition probability expressions will change. Suppose our control action consists of forcibly altering the value of a single gene, g, from 0 to 1 or from 1 to 0. Thus, m = 1. Then the new transition probabilities with control, denoted by $Prc1$, are given by

$$Prc1(w(t+1) = b \mid w(t) = a) = Pr(w(t+1) = b \mid w(t) = a + 2^{n-g})\, func(a)$$
$$+ Pr(w(t+1) = b \mid w(t) = a - 2^{n-g})(1 - func(a)) \quad (53)$$

where

$$func(a) = \begin{cases} 1, & \text{if state of gene } g \text{ is 0 for } a, \\ 0, & \text{if state of gene } g \text{ is 1 for } a, \end{cases}$$

and the transition probabilities, Pr, without control are given by Eq. 51. Here, a and b range over 1 through $2^n k$. As before

we can reduce the dimension of the state space by replacing the ws in Eq. 53 by zs and using Eq. 52 to determine the transition probabilities without the control action:

$$Prc1(z(t+1)=b \mid z(t)=a) = Pr(z(t+1)=b \mid z(t)=a+2^{n-g})func(a) + Pr(z(t+1)=b \mid z(t)=a-2^{n-g})(1-func(a)) \quad (54)$$

By letting a and b vary over 1 to 2^n and making use of Eq. 54, we can determine the 2^n x 2^n matrix $A(v(t))$ of control-dependent transition probabilities.

From this point onwards the formulation and solution of the control problem is exactly the same as in Section 5. To avoid unnecessary repetition we proceed directly to the melanoma example considered in the two previous sections.

Melanoma Example

We consider a 7-gene network with genesWNT5A, pirin, S100P, RET1, MART1, HADHB and STC2. Although derived from the same data, this network is designed based on *steady-state considerations* and, therefore, differs from the PBNs considered in Sections 5 and 6. Carrying out the new design can be justified by the fact that most microarray-based gene-expression studies do not involve controlled time series experimental data; rather, it is assumed that data result from sampling from the steady state. Consequently, to obtain the PBN here, we have used a Bayesian connectivity-based approach of [23] to construct four highly probable Boolean networks that are used as the constituent Boolean networks in the PBN, with their selection probabilities based on their Bayesian scores. The four generated Boolean networks are shown in Fig. **9** through 12, where the states are labeled from 1 to 128 = 2^7. Each constituent network is assumed to be derived from steady-state gene-expression data, and the attractor states and the level sets are shown in the figures. Observe that in each of these networks the state enters an attractor cycle in a small number of steps (at most nine), which is consistent with what is expected in real networks [23]. The control strategy of this section has been applied to the designed PBN with pirin chosen as the control gene and $p = q = 0.01$. Fig. **13** shows the expected cost for a finite horizon problem of length 5 originating from each of the 128 states. In these simulations, the problem formulation for 2^n states has been used. The cost of control is assumed to be 0.5 and the states are assigned a terminal penalty of 5 if WNT5A is 1 and 0 if WNT5A is 0. The control objective is to down-regulate the WNT5A gene. From Fig. **13**, it is clear that the expected cost with control is much lower than that without control, which agrees with our objective.

8. CONCLUDING REMARKS

We have discussed several approaches that have been recently developed for addressing the issue of intervention in probabilistic gene regulatory networks. The results reported indicate that significant progress has been made in this area; however, numerous open issues remain and these will have to be successfully tackled before the methods suggested in this paper find application in actual clinical practice. We next discuss some of the issues that we are aware of at the current time:

Methodical Assignment of Terminal Penalties

The formulation of the optimal control problem in Section 5 assumes that there is a terminal penalty associated with each state of the PBN; however, assignment of these terminal penalties for cancer therapy is by no means a straightforward task. The reason is that while the intervention will be carried out only over a finite horizon, one would like to continue to enjoy the benefits in the steady state. For such purposes, the kind of terminal penalty used for the melanoma cell line study of Section 5 is inadequate since it fails to capture the steady state behavior once the intervention has ceased. To remedy the situation, we propose to assign terminal penalties based on equivalence classes. The results of preliminary simulation studies in this regard [24] appear to be encouraging.

Choice of Control Input

In the case of the melanoma cell line study presented in Section 5, one of the genes in the PBN, namely pirin, has been used as a control input. The question is how to decide which gene to use. Of course, one consideration is to use genes for which inhibitors or enhancers are readily available. However, even if such a gene is chosen, how can we be certain that it is capable of controlling some other gene(s)? Although the answer is not clear at this stage, we do believe that the traditional control theoretic concepts such as *controllability* and *observability* [25] may yield some useful insights. Another possibility is to use the concept of gene *influence* introduced in [1], an approach that we have preliminarily explored in [9].

Intervening to Alter the Steady-State Behavior

Given a Boolean network, one can partition the state-space into a number of attractors along with their basins of attraction. The attractors characterize the long-run behavior of the Boolean network and have been conjectured by Kauffman to be indicative of the cell type and phenotypic behavior of the cell. Consequently, a reasonable objective of therapeutic intervention could be to alter the attractor landscape in the associated Boolean network. Such an idea can be generalized to PBN's and a brute force approach aimed at such intervention has been presented in Section 4. More systematic approaches for affecting the steady-state behavior will need to be developed possibly by exploiting and building upon existing control-theoretic results.

PBN Design from Steady State Data

Yet another aspect that merits further investigation is motivated by the fact that most currently available gene-expression data comes from steady-state phenotypic behavior and really does not capture any temporal history. Consequently, the process of inferring PBNs from the data will have to be modified, in the sense that it will have to be guided more by steady-state and limited connectivity considerations. Major research efforts in these directions are currently under way [23, 26]. This last aspect further underscores the fact that the category of intervention cannot be researched in isolation. Issues that arise upstream will definitely impact intervention and vice versa.

The optimal control results presented in this paper assume known transition probabilities and pertain to a finite-horizon problem of known length. Their extension to the situation where the transition probabilities and the horizon length are unknown is a topic for further investigation. Finally, the results presented in this paper correspond to the

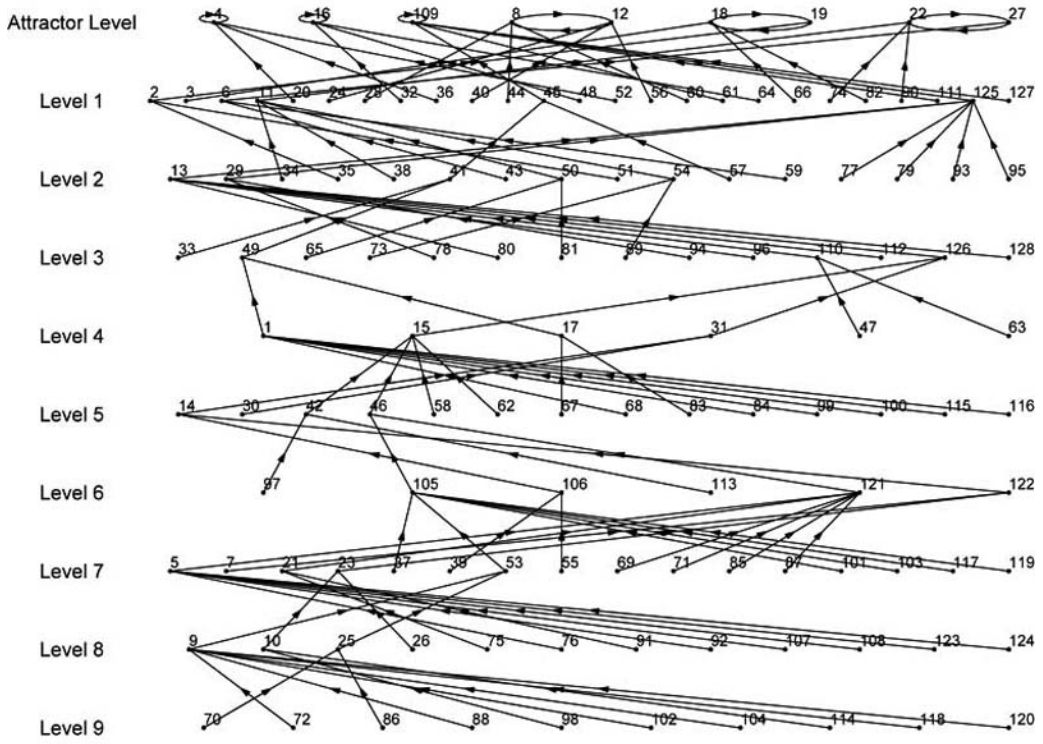

Fig. (9). Network 1 [9].

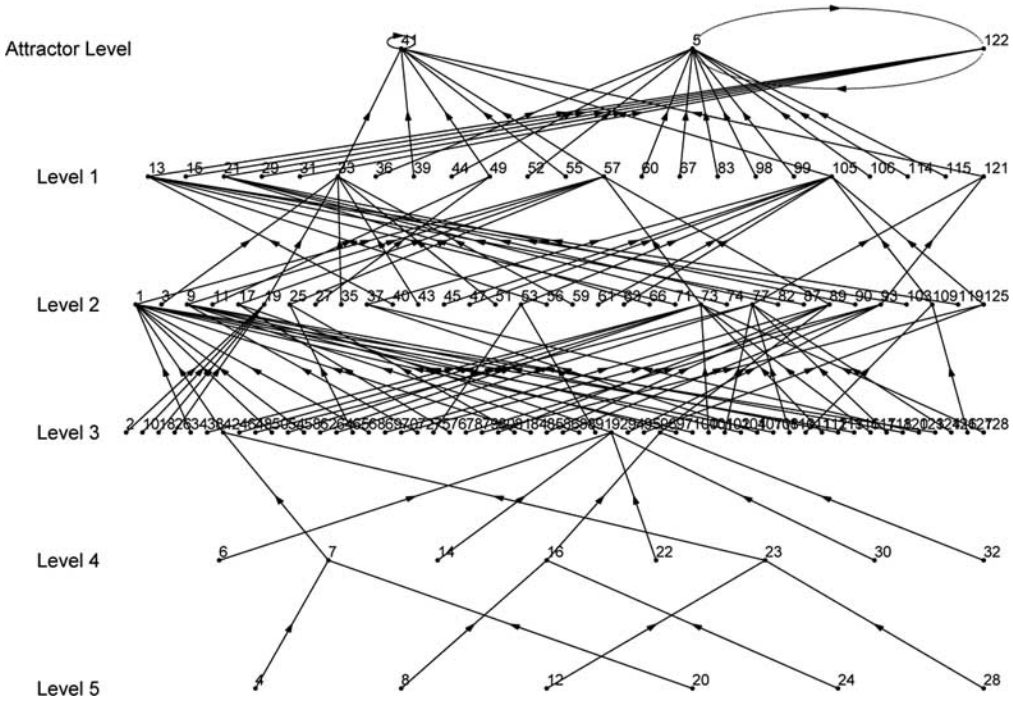

Fig. (10). Network 2 [9].

following stages in standard control design: modeling, controller design and verification of the performance of the designed controller *via* computer simulations. The designed controllers will have to be successfully implemented in practical studies, at least with cancer cell lines, to validate the use of engineering approaches in translational medicine. A considerable amount of effort needs to be focused on this endeavor.

ACKNOWLEDGEMENTS

This work was supported in part by the National Science Foundation under Grants ECS-0355227 and CCF-0514644, the National Cancer Institute under Grant CA 90301, the National Human Genome Research Institute, the University of M. D. Anderson Cancer Center, and the Translational Genomics Research Institute.

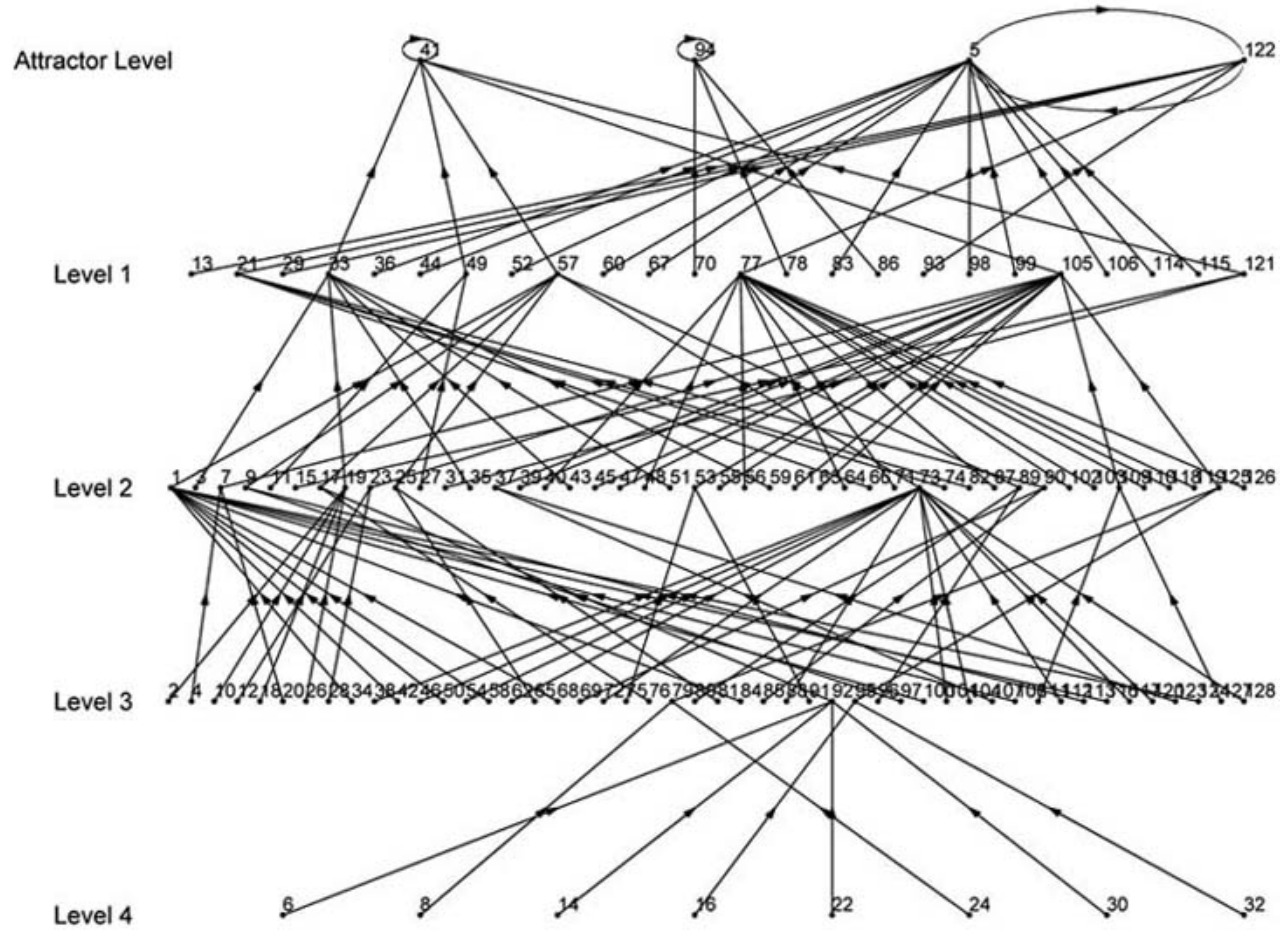

Fig. (11). Network 3 [9].

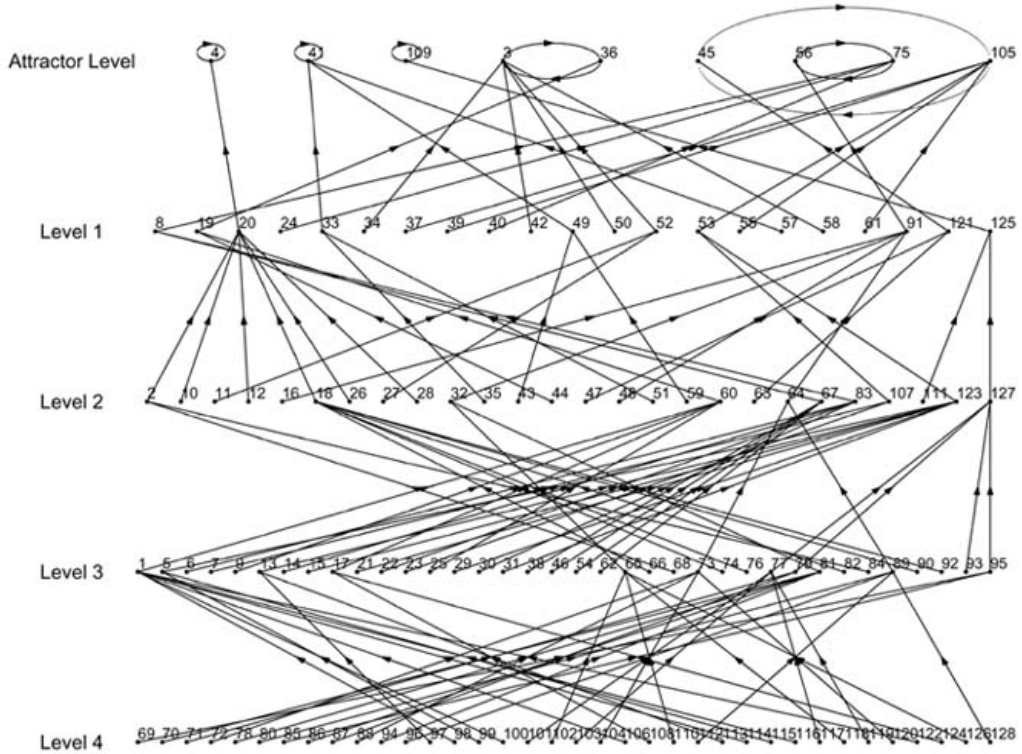

Fig. (12). Network 4 [9].

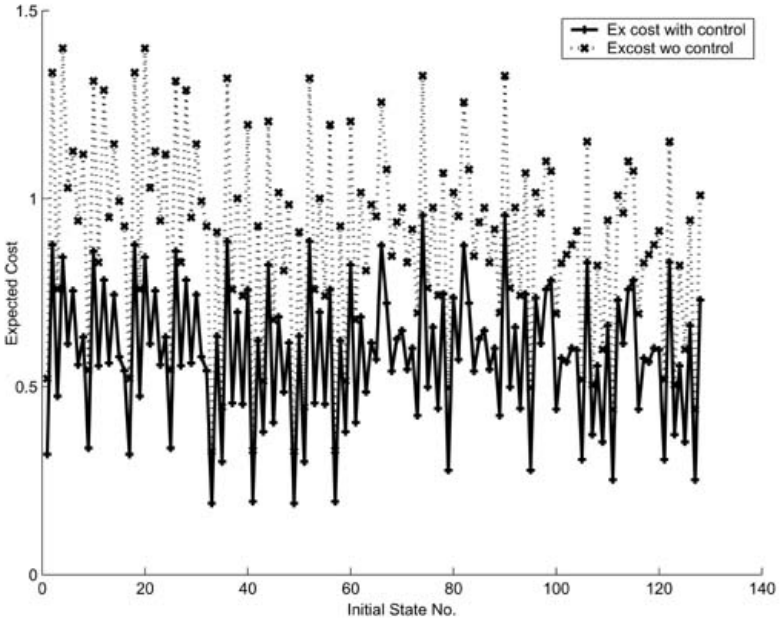

Fig. (13). Expected cost for a finite horizon problem of length 5 originating from the different initial states [9].

REFERENCES

[1] Shmulevich I, Dougherty ER, Kim S, Zhang W. Probabilistic Boolean Networks: A Rule-based Uncertainty Model for Gene Regulatory Networks. *Bioinformatics* **2002**; 18: 261-74.

[2] Kauffman SA. Metabolic Stability and Epigenesis in Randomly Constructed Genetic Nets. *Theo Biol* **1967**; 22: 437-67.

[3] Kauffman SA, Levin S. Towards a General Theory of Adaptive Walks on Rugged Landscapes. *Theo Biol* **1987**; 128: 11-45.

[4] Kauffman SA, The Origins of Order: Self-organization and Selection in Evolution, Oxford University Press, New York, 1993.

[5] Shmulevich I, Dougherty ER, Zhang W. Gene Perturbation and Intervention in Probabilistic Boolean Networks. *Bioinformatics* **2002**; 18:1319-1331.

[6] Shmulevich I, Dougherty ER, Zhang W. Control of Stationary Behavior in Probabilistic Boolean Networks by Means of Structural Intervention. *Biological Sys* **2002**; 10:431-446.

[7] Datta A, Choudhary A, Bittner ML, Dougherty ER. External Control in Markovian Genetic Regulatory Networks. *Mach Learning* **2003**; 52: 169-191.

[8] Datta A, Choudhary A, Bittner ML, Dougherty ER. External Control in Markovian Genetic Regulatory Networks: The Imperfect Information Case. *Bioinformatics* **2004**; 20: 924-930.

[9] Pal R, Datta A, Bittner ML, Dougherty ER, Intervention in Context-Sensitive Probabilistic Boolean Networks. *Bioinformatics* **2005**; 21: 1211-1218.

[10] Shmulevich I, Dougherty ER. Modeling Genetic Regulatory Networks with Probabilistic Boolean Networks, Genomic Signal Processing and Statistics, In: Dougherty ER, Shmulevich I, Chen J, Wang ZJ Eds, EURASIP Book Series on Signal Processing and Communication. Hindawi Publishing Corporation, New York, 2005.

[11] Shmulevich I, Dougherty ER, Zhang W. From Boolean to Probabilistic Boolean Networks as Models of Genetic Regulatory Networks, *Proc IEEE* **2002**; 90(11): 1778-1792.

[12] Brun M, Dougherty ER, Shmulevich I, "Steady-State Probabilities for Attractors in Probabilistic Boolean Networks," *Signal Processing* **2005**; 85(10): 1993-2013.

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

[14] Cinlar E. Introduction to Stochastic Processes. Prentice Hall, Eglewood Cliffs, NJ 1997.

[15] Goldberg D. Genetic Algorithms in Search, Optimization, and Machine Learning , Addison-Wesley, Reading, MA, 1989.

[16] Bertsekas DP, Dynamic Programming and Optimal Control. *Athena Scientific* **1995**; 1 & 2.

[17] Bittner M, Meltzer P, Chen Y, *et al.* Molecular Classification of Cutaneous Malignant Melanoma by Gene Expression Profiling. *Nature* **2000**; 406 (6795): 536-540.

[18] Weeraratna AT, Jiang Y, Hostetter G, *et al*. Wnt5a Signalling Directly Affects Cell Motility and Invasion of Metastatic Melanoma. *Cancer Cell* **2002**; 1: 279-288.

[19] Kim S, Li H, Dougherty ER, *et al*. Can Markov Chain Models Mimic Biological Regulation. *Biological Sys* **2002**; 10(4): 447-458.

[20] Dougherty ER, Kim S, Chen Y. Coefficient of determination in nonlinear signal processing. *Signal Processing* **2000**; 80(10): 2219–2235.

[21] Kim S, Dougherty ER, Chen Y, *et al*. Multivariate measurement of gene-expression relationships. *Genomics* **2000**; 67: 201–209.

[22] Kim S, Dougherty ER, Bittner ML, *et al.* A general framework for the analysis of multivariate gene interaction *via* expression arrays. *Biomedical Optics* **2000**; 4(4): 411–424.

[23] Zhou X, Wang X, Pal R, Ivanov I, Bittner ML, Dougherty ER. A Bayesian Connectivity - Based Approach to Constructing Probabilistic Gene Regulatory Networks**,** *Bioinformatics* **2004**; 20(17): 2918-2927.

[24] Choudhary A, Datta A, Bittner M, Dougherty ER. Assignment of Terminal Penalties in Controlling Genetic Regulatory Networks. *Am Control Conf* **2005**; 417-422.

[25] Kalman RE. Canonical structure of linear dynamical systems. *Proc Natl Acad Sci USA* **1962**; 596–600.

[26] Pal R, Ivanov I, Datta A, Bittner M, Dougherty ER. Generating Boolean Networks with a Prescribed Attractor Structure. *Bioinformatics* **2005**; 21: 4021-4025.

Received: August 23, 2005 Revised: September 17, 2005 Accepted: September 23, 2005