INTERVENTION IN GENE REGULATORY NETWORKS

A Dissertation

by

ASHISH CHOUDHARY

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

August 2006

Major Subject: Electrical Engineering

INTERVENTION IN GENE REGULATORY NETWORKS

A Dissertation

by

ASHISH CHOUDHARY

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Chair of Committee,	Aniruddha Datta
Committee Members,	Edward Dougherty
	Shankar Bhattacharyya
	Bani Mallick
Head of Department,	Costas Georghiades

August 2006

Major Subject: Electrical Engineering

ABSTRACT

Intervention in Gene Regulatory Networks. (August 2006)
Ashish Choudhary, B.Tech., Indian Institute of Technology Bombay;
M.S., Texas A&M University
Chair of Advisory Committee: Dr. Aniruddha Datta

In recent years Boolean Networks (BN) and Probabilistic Boolean Networks (PBN) have become popular paradigms for modeling gene regulation. A PBN is a collection of BNs in which the gene state vector transitions according to the rules of one of the constituent BNs, and the network choice is governed by a selection distribution.

Intervention in the context of PBNs was first proposed with an objective of avoiding undesirable states, such as those associated with a disease. The early methods of intervention were ad hoc, using concepts like mean first passage time and alteration of rule based structure. Since then, the problem has been recognized and posed as one of optimal control of a Markov Network, where the objective is to find optimal strategies for manipulating external control variables to guide the network away from the set of undesirable states towards the set of desirable states. This development made it possible to use the elegant theory of Markov decision processes (MDP) to solve an array of problems in the area of control in gene regulatory networks, the main theme of this work.

We first introduce the optimal control problem in the context of PBN models and review our solution using the dynamic programming approach. We next discuss a case in which the network state is not observable but for which measurements that are probabilistically related to the underlying state are available.

We then address the issue of terminal penalty assignment, considering long term

prospective behavior and the special attractor structure of these networks.

We finally discuss our recent work on optimal intervention for the case of a family of BNs. Here we consider simultaneously controlling a set of Boolean Models that satisfy the constraints imposed by the underlying biology and the data. This situation arises in a case where the data is assumed to arise by sampling the steady state of the real biological network.

To My Parents.

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Datta, and Dr. Dougherty from whom I learned a lot about engineering and a lot about life. I would also like to thank Yoga, JianPing, Marcel, Ivan, Ranadip, Sima, Yufei and all other past and present members of the GSP team for all the informative and thought provoking discussions I have had with them.

I would like to thank my parents for their love and support over this long period of study.

TABLE OF CONTENTS

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

FIGURE Page

CHAPTER I

INTRODUCTION

Numerous gene regulatory models have been proposed. For the most part these have been developed for descriptive purposes, by which we mean that their purpose is to characterize gene interaction. From a translational perspective, a salient objective is to base diagnosis and treatment for disease upon these models. For treatment, this constitutes the derivation of intervention strategies that affect the network in beneficial ways.

To date, the largest effort in deriving intervention methods for gene regulatory networks has been in the context of probabilistic Boolean networks (PBNs). PBNs are essentially probabilistic generalizations of the standard Boolean networks (introduced by Kauffman [1, 2, 3]), in which at any discrete time point the gene state vector transitions according to the rules of one of the constituent Boolean networks [4].

Early efforts in intervention included ad hoc methods like -resetting the state of the PBN, as necessary, to a more desirable initial state and letting the network evolve from there [5], and changing the steady-state (long-run) behavior of the network by minimally altering its rule-based structure [6].

In [7] it was explicitly recognized that since PBNs are essentially Markov chains, the well researched theory of Markov decision processes could be used to find optimal intervention strategies. In this seminal work, for the first time notions of (i) *control cost* i.e. the cost of using control, (ii) *terminal penalty* i.e. the cost of terminating control in a state (based on desirability and undesirable of the state profile), and (iii) minimization of the *composite cost function* over a *time horizon*; were formalized

---

The journal model is *IEEE Transactions on Automatic Control*.

in the context of biological networks. Since then the optimal intervention problem has been studied for (i) the imperfect information case [8], (ii) the context sensitive case [9], (iii)the infinite horizon case [10], (iv) and the family of networks case [11]. This thesis deals with the various aspects of this optimal intervention problem and is organized as follows.

We begin by reviewing the basic concepts and motivation of BNs and PBN models in chapter II. In chapter III we introduce the control problem in the context of the above models and also review our solution using the dynamic programming approach. In chapter IV we extend the results in chapter III to cover the case where the state is not observable. In chapter V we discuss the problem of assigning terminal penalties in PBNs. In chapter VI we pose and solve the problem of simultaneously controlling a family of Boolean Networks. Chapter VII has a discussion on issues not addressed in this thesis and possible future research directions. Certain technical details are relegated to the appendix.

Examples based on a hypothetical 3 gene network, and WNT5A networks(designed from data obtained from the study of metastatic melanoma) have been worked out in all chapters for illustration purposes.

CHAPTER II

DISCRETE MODELS OF GENE REGULATORY NETWORKS

In this chapter we review the philosophy and mechanics of BNs and PBNs.

A. Boolean Network

Boolean networks compose a class of discrete models where the expression levels of each gene are assumed to have two possible values: ON or OFF [1]. Such a model cannot capture the underlying continuous and stochastic biochemical nature of protein production and gene regulation; however, one often encounters genes that are essentially ON or OFF throughout a given biochemical pathway. The switch-like regulatory function of these genes determines their role in regulation, and this activity is well represented by a coarse-grain model like a BN. This, together with the relative simplicity of the dynamical system described by a BN, explains why such networks have attracted significant attention from the research community [3, 4, 12].

A *Boolean Network* (*BN*) consists of a set of genes (nodes) in which each gene can take on one of two binary values, 0 or 1 ([1, 3]). Given n genes, the activity level of gene i at time step k is denoted by $x_i(k)$, where $x_i(k) = 0$ indicates that gene i is not expressed and $x_i(k) = 1$ indicates that it is expressed. The overall expression levels of all the genes in the network at time step k is given by the state (row) vector $x(k) = [x_1(k), x_2(k), \cdots, x_n(k)]$, also called the *gene activity profile* (*GAP*) of the network at time k. Gene i evolves from time k to $k+1$ according to the Boolean function $f_i(x_1(k), x_2(k), \ldots, x_n(k))$. Usually the value of f_i does not depend on the entire set $\{x_1, x_2, \cdots, x_n\}$ of n gene values but only on a finite subset $\mathcal{P}_i$ of it. This set $\mathcal{P}_i$ is called the *predictor set* for the ith gene. Specifying the truth table for the functions $f_1, f_2, \cdots, f_n$ along with the associated predictor sets $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n$ supplies

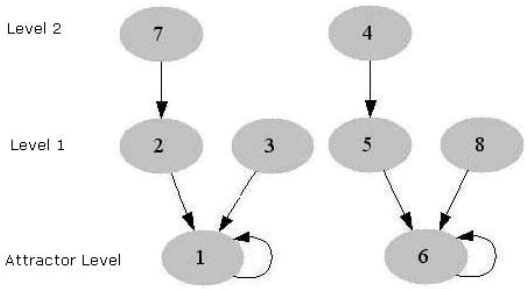

Fig. 1. 3 gene BN with $\mathcal{P}_1 = \{x_1, x_2, x_3\}$,$\mathcal{P}_2 = \{x_1, x_3\}$ and $\mathcal{P}_3 = \{x_1\}$

all the information necessary to determine the time evolution of the states of the BN.

Table I. Table of functions.

x_1	x_2	x_3	f_1	f_2	f_3
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	0	1
1	1	0	0	0	1
1	1	1	1	0	1

The binary n-digit state vector $x(k)$ can be mapped to positive integers $z(k)$ so that as $x(k)$ ranges from $00\cdots0$ to $11\cdots1$, $z(k)$ goes from 1 to 2^n. Here we employ the decimal representation $z(k)$ and the set $\mathcal{S} = \{1, 2, \cdots, 2^n\}$ constitutes the state space for the Boolean network.

The truth table and the corresponding boolean networks state transition diagram for a 3 gene network are shown in Table I and Figure 1 respectively.

Attractors play a key role in Boolean networks. Given a starting state, within a finite number of steps, the network will transition into a cycle of states, called an *attractor*, and will continue to cycle thereafter. Each attractor is a subset of a *basin* composed of those states that lead to the attractor if chosen as starting states. The basins form a partition of the state space for the network. Non-attractor states are transient. They are visited at most once on any network trajectory.

For the network in Figure 1 there are two singleton attractors, $1\{000\}$ and $6\{101\}$. There are two transient levels, where a state in level k transitions to an attractor in k time steps.

B. Probabilistic Boolean Network

A *Probabilistic Boolean Network* (*PBN*) consists of a finite collection of BNs over a fixed set of genes, where each BN is defined by a fixed network function. The network transitions according to one of the constituent BN at each time step. At each moment of time there is a probability q of switching to a different constituent BN, where, given a switch, each BN composing the network has a probability of being selected. If $q = 1$, then a new network function is randomly selected at each time point; if $q < 1$, then the PBN remains in a given constituent BN until the random binary variable governed by q calls for a network switch. If $q = 1$, the PBN is said to be *instantaneously random*, the idea being to model uncertainty in model selection; if $q < 1$, it is said to be *context-sensitive*, the idea being to model the situation where the model is affected by latent variables outside the model. Moreover, if at any given moment of discrete time there is a probability p of randomly switching the state of the PBN; such a PBN is said to be a PBN with *random perturbations*. A detailed exposition can be found in [4, 9].

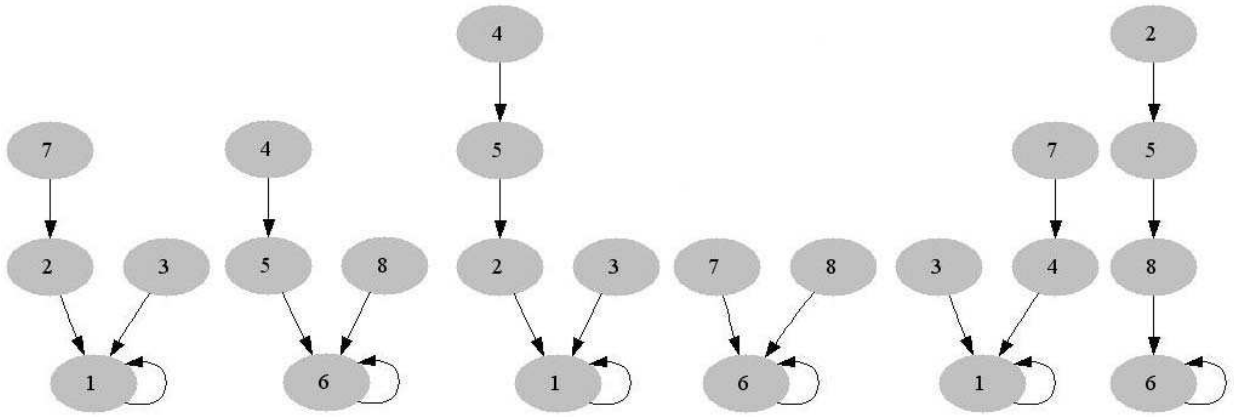

Fig. 2. 3 Boolean Networks (N_1,N_2 and N_3) over 3 genes

Figure 2 shows a set of 3 BNs over 3 genes. The corresponding instantaneously random PBN when each of the 3 BNs are equally likely to be selected is shown in Figure 3.

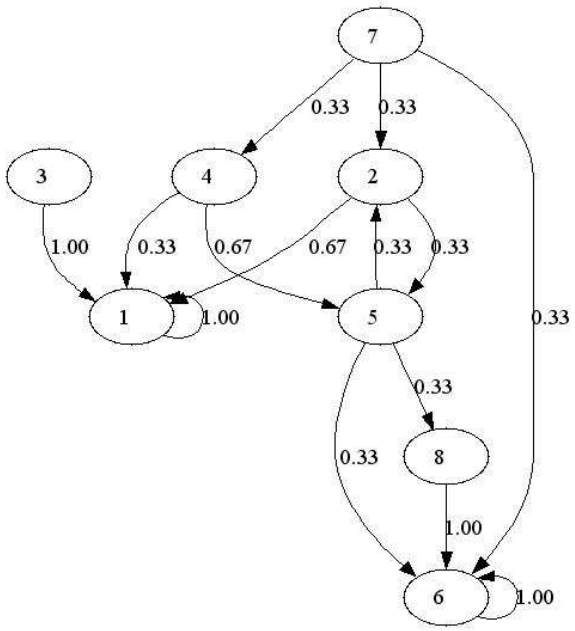

Fig. 3. The instantaneously random PBN obtained from the 3 BNs in Figure 2

CHAPTER III

INTERVENTION IN GENE REGULATORY NETWORKS

In this chapter we present our original analysis of the control problem for an instantaneously random PBN [7]. Subsequent advances in the control of PBNs like the imperfect information case; the context sensitive case and the infinite horizon case are discussed in [8], [9] and [10] respectively.

Consider the problem of external control in an instantaneously random PBN with n genes and m control inputs, u_1, u_2, $\cdots$, u_m, each of which can take on only the binary values 0 or 1. At any time k, the row vector $u(k) = [u_1(k), u_2(k), \cdots, u_m(k)]$ describes the complete status of all the control inputs. $u(k)$ can take on all binary values from $00\cdots0$ to $11\cdots1$. One can equivalently represent the control input status using a decimal number $v(k)$ ranging from 0 to $2^m - 1$, so that $\mathcal{A} = \{0, 1, \cdots, 2^m - 1\}$ is the set of possible control actions. This set could be a function of the state, because not all control alternatives may be available from all states. As shown in [7], the one-step evolution of the probability distribution vector in the case of a PBN containing 2^n states with control inputs takes place according to the equation

$$w(k+1) = w(k)A(v(k)) \tag{3.1}$$

where $w(k)$ is the 2^n dimensional state probability distribution vector and $A(v(k))$ is the $2^n \times 2^n$ control-dependent transition probability matrix (TPM). Since the transition probability matrix is a function of the control input $v(k)$, 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 input at different time steps. Intuitively, it appears possible to make the states of the network evolve in a desirable fashion by appropriately choosing the control input at each time step.

A. Definitions and Problem Statement

To formalize the ideas from previous section we define the following quantities:

- $a_{ij}(v)$ is the ith row, jth column entry of the stochastic matrix $A(v)$, $v \in \mathcal{A}$.
- M represents the treatment/intervention window; control actions are taken at steps $0, 1, \cdots, M-1$.
- For any $i \in \mathcal{S}$, $C_k(i, v)$ is the cost of applying the control v in state i at the kth time step.
- For any $i \in \mathcal{S}$, $C_M(i)$ is the terminal cost associated with the state i, i.e. the cost of ending up in state i at the Mth time step, when no more control steps are remaining.

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,. . . , $M-1$. Suppose at time step k, the state of the PBN is given by $z(k)$ and the corresponding control input is $v(k)$, then by definition $C_k(z(k), v(k))$ is the associated control cost.

Thus expected cost of control over the entire treatment horizon becomes

$$E[\sum_{k=0}^{M-1} C_k(z(k), v(k))|z(0)] \tag{3.2}$$

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 (3.1). Consequently, the cost in (3.2) must be defined using expectation. (3.2) provides one component of the finite-horizon cost, namely the cost of control. We next discuss 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 (3.1) 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 in $\mathcal{S} = \{1, 2, \ldots, 2^n\}$.

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 we need to assign a penalty, or terminal cost, $C_M(z(M))$ to each of them. Thus we arrive 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 (3.2), the finite-horizon cost to be minimized is given by

$$E[\sum_{k=0}^{M-1} C_k(z(k), v(k)) + C_M(z(M))|z(0)] \tag{3.3}$$

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,

$$v(k) = \mu_k(z(k)) \tag{3.4}$$

where $\mu_k : \mathcal{S} \to \mathcal{A}$. 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

$$J_\pi(z(0)) = E[\sum_{k=0}^{M-1} C_k(z(k), \mu_k(z(k))) + C_M(z(M))] \tag{3.5}$$

subject to the constraint

$$\Pr\{z(k+1) = j | z(k) = i, v(k) = v\} = a_{ij}(v) \tag{3.6}$$

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

B. Solution Using Dynamic Programming

As explained in [7] the optimal control problem described by (3.5) and (3.6) can be solved using the technique of *Dynamic Programming*. For a given initial state $z(0)$, the *optimal cost* for the finite horizon optimal control problem is given by $J_0(z(0))$, where for $k = 0, 1, 2, \cdots, M-1$, $J_k(i)$ is known as the *cost to go* function at the kth time step from state i [7]. The J_k's can be found using the following recursive formula

$$J_k(i) = \min_{v \in \mathcal{A}} [C_k(i, v) + \sum_{j \in \mathcal{S}} a_{ij}(v).J_{k+1}(j)]., k = M-1, M-2, \cdots, 0. \tag{3.7}$$

$$J_M(i) = C_M(i). \tag{3.8}$$

Intuitively equation (3.7) states that the optimal cost to go from state i at the kth time step is the sum of the cost of the optimal control action at state i and the expected value of the cost to go at the $(k+1)$th time step. Since there is no control action in the terminal time step, (3.8) simply formalizes the fact that the cost to go at the terminal time step equals the penalty associated with the terminal state.

The optimal control obtained from (3.7), (3.8) can be represented as a table $\mathcal{S} \times \mathcal{T} \rightarrow \mathcal{A}$, where $\mathcal{T}$ is the discrete time variable. To set up such a table, we first tabulate $J_M(i)$ for any $i \in \mathcal{S}$ using (3.8). $J_{M-1}(i)$ and the corresponding minimizing control v can be calculated and stored for all $i \in \mathcal{S}$ using (3.7) and making use of the

$J_M(j)$ values tabulated earlier. By repeating these steps we can fill up the table for $k = M-2, \cdots, 0$.

C. 3 Gene PBN Example

To illustrate the algorithmic details, we consider the 3-gene network in Figure 3. Suppose x_3 is the penalty gene.[1] When $x_3 = 1$ in a state, it is undesirable, and a terminal penalty of +5 is assigned. States with $x_3 = 0$ are assigned a terminal penalty of 0. Let x_1 be the control gene and suppose that the control action is to forcibly flip this gene: for $v(k) = 1$, flip gene x_1 at the kth time step and for $v(k) = 0$ leave it as is. Let the cost of control $C_k(i, v) = C(v) = v$.

Transitions take place according to the network transition rule – for example, in the network in Figure 3, if $z(k) = 6$ (101) and $v(k) = 1$, then $z(k+1) = 1$, corresponding to a jump from state 6 (101) to state 2 (001) and then evolution to the state 1 (000) with a probability 0.67 or to state 5 (100) with probability 0.33.

Table II shows the optimal control action from each state at each time step while the optimal cost to go is shown in Table III. Figure 4 compares the expected cost of using control to that of not using control when for an $M = 2$ step policy.

Table II. Table of optimal control action. The entry in kth row and ith column is the optimal control v at time k in state i.

Time (k)	State							
	1	2	3	4	5	6	7	8
0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	1	1

[1] A penalty gene is a gene for which certain expression statuses are known to be undesirable.

Table III. Table of optimal cost to go for an M=2 step control. J_2 is the terminal penalty.

Time(k)	State i							
	1	2	3	4	5	6	7	8
$J_0(i)$	0	0.33	0	0.66	0.66	1.0	0.33	1.0
$J_1(i)$	0	0	0	0	1.0	1.0	1.0	1.0
$J_2(i)$	0	5	0	5	0	5	0	5

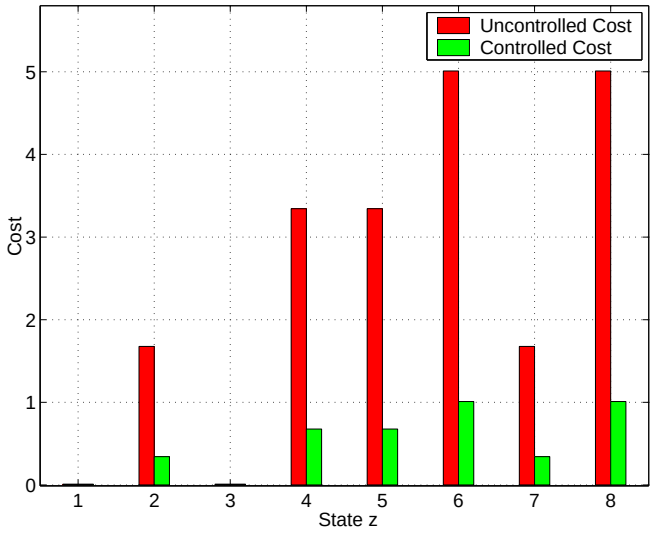

Fig. 4. Expected costs with the optimal control and no control

D. Metastatic Melanoma Application

In this section, we derive an optimal intervention strategy for a particular gene regulatory network, originally discussed in [7]. The network chosen is one developed from the data collected in a study of metastatic melanoma [13]. In this expression profiling study, the abundance of messenger RNA for the gene WNT5A was found to be a highly discriminating 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 [14]. 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, by the use of an antibody that binds Wnt5a protein, could substantially reduce Wnt5a's ability to induce a metastatic phenotype. This of course 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 suggests 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 [15]. As discussed in that paper, the WNT5A network was obtained by studying the predictive relationship between 587 genes. The expression status of each gene was quantized to one of three possible levels: -1 (down-regulated), 0 (unchanged) and 1(up-regulated). Thus in this case, the gene activity profile at any time step is not a binary number but a ternary one. However, the PBN formulation and the associated control strategy can be developed exactly as described in earlier sections, with the only difference that now for an n-gene network, we will have 3^n states instead of the 2^n states encountered earlier. A 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 was narrowed down to the 10 most significant ones. The dataset and the corrsponding 10 gene network are shown in Figure 5 and Figure 6 respectively.

We further narrowed down the number of genes in the network to 7 by using

Case No.	pirin	WNT5A	S100P	RET1	MMP3	PHOC	MART1	HADHB	synuclein	STC2
UACC457 UACC383 UACC1022 TC_1376_3 TD_1376_3 TD_1730 TD_1638 TD_1720	1	−1	1	−1	−1	−1	1	1	1	−1
UACC3093	0	−1	1	−1	−1	−1	1	1	1	−1
M92_001 UACC257	1	−1	1	0	0	−1	1	1	1	−1
WM1791C	0	1	0	−1	1	−1	0	0	1	1
UACC1097	0	0	0	0	0	0	0	0	1	0
UACC903	0	0	0	0	0	0	1	1	0	0
UACC2534	1	−1	1	−1	0	−1	1	0	1	−1
M93_007 UACC1273	1	−1	0	0	0	0	1	1	0	0
UACC1265	1	−1	1	0	0	0	1	1	1	0
UACC091 UACC502	1	−1	0	0	0	0	1	1	0	−1
TD1348	0	−1	1	−1	−1	−1	1	0	1	−1
UACC1012	0	1	0	−1	1	0	0	0	1	0
M91_054	1	−1	1	−1	0	0	1	1	0	0
M92_047	1	1	−1	0	0	0	0	0	0	−1
HA_A	0	−1	1	−1	0	0	0	0	0	−1
TC_F027	0	−1	1	0	−1	0	1	0	1	−1
UACC647	0	1	0	−1	1	0	0	0	0	0
UACC930	0	1	−1	−1	0	0	0	0	−1	1
UACC1529	−1	0	0	0	−1	0	0	0	1	−1
UACC827T	0	0	1	−1	0	−1	0	0	1	1
UACC2837	0	−1	−1	0	0	0	0	0	−1	0

Fig. 5. WNT5A dataset.

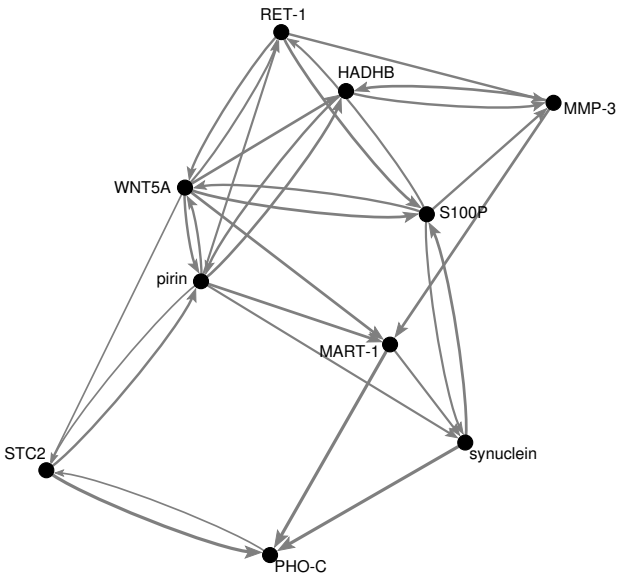

Fig. 6. 10 Gene network for metastatic melanoma.

COD analysis([16, 17, 18]) on the 31 samples. The resulting genes along with their multivariate relationship are shown in Figure 7.

For each gene in this network, we determined their two best two-gene predictors and their corresponding CODs. Using the procedure discussed in [4], the COD information for each of the predictors was then used to determine the $3^7 \times 3^7$ matrix of transition probabilities for the Markov Chain corresponding to the dynamic evolution of the gene-activity profile of the seven gene network.

In this context, it is appropriate to point out that to apply the control algorithm, 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.

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 chose a window of length $M = 5$, i.e. control inputs would be applied only at time

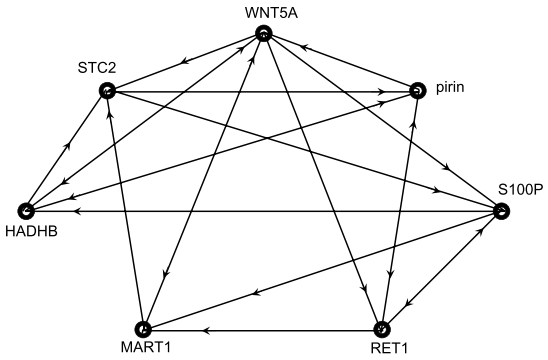

Fig. 7. 7 Gene network for metastatic melanoma

steps $0, 1, 2, 3$ and 4. The terminal penalty at time step 5 was chosen as follows. Since our objective is to ensure that WNT5A is down regulated, we assigned a penalty of zero 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 they do 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 were used and next we discuss the two cases separately.

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 assigned a cost of 1 to each such forcible change and solved for the optimal control using dynamic programming. The net result was a table of optimal control inputs for each of the $2187(= 3^7)$ states at each of the five time points. Using these control inputs, we studied the evolution of the state probability distribution vectors with and without control. For every possible initial state, our simulations indicated that at every time step from 1 to 5, the probability of WNT5A being equal to -1 was higher with control than that without control. Furthermore, with control, WNT5A always reached -1 at the final time point ($k = 5$). Thus, we conclude that the optimal control strategy, indeed, 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 has to be forcibly reset to -1 at that time step. Having chosen these design parameters, we implemented the dynamic programming algorithm with pirin as the control.

Using the resulting optimal controls, we studied the evolution of the state probability distribution vectors with and without control. For every possible initial state, our simulations indicated that, at the final state, the probability of WNT5A being equal to -1 was higher with control than that without control. In this case, there was, however, 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 was not, in general, equal to 1. This is not surprising given that, in this case, 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 0.6735. This is quite good in view of the fact that the same probability would have been equal to zero in the absence of any control action.

E. Conclusions

In this chapter we formally introduced the optimal control problem. We also demonstrated the efficacy of optimal control in reducing the expected cumulative cost function for the PBNs in Figure 3, and the Metastatic melanoma application. In this chapter we used full state feedback assuming that the entire state vector is observable, in the next chapter we relax this assumption.

CHAPTER IV

THE IMPERFECT INFORMATION CASE

The control table that emerges from (3.7) depends explicitly on knowledge of the current state z_k to compute the minimizing v_k at each time state.[1]

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. Like state and control, we can represent θ_k, the output status at time k of the PBN using a decimal number ranging from 1 to 2^q, so that $\mathcal{Q}$=$\{1, \cdots, 2^q\}$ is the set of possible outputs.

This output θ_k at time k is probabilistically related to the state z_k at time k and the input v_{k-1} through the known conditional probability measure $Pr_{\theta_k}(.|z_k, v_{k-1})$ defined by

$$Pr\{\theta_k = \theta | z_k = j, v_{k-1} = v\} = r^v_{j\theta}. \tag{4.1}$$

Let I_k denote the total information that is available for control at time k. Then clearly $I_k = [\theta_0, v_0, \theta_1, v_1, \cdots, v_{k-1}, \theta_k]^T$. Furthermore, I_k can be generated recursively using the equation

$$I_{k+1} = [I_k^T, v_k, \theta_{k+1}]^T,\ I_0 = \theta_0. \tag{4.2}$$

Since the state z_k is not available, we would to replace the state feedback by information feedback.

---

[1]In this chapter we use subscript for the time variable i.e z_k is equivalent to $z(k)$

A. Belief Vector

Along the lines of [19, 20], we now define the belief vector $P_k = [p_k^1, p_k^2, \cdots, p_k^{2^n}]$ at time k, where $p_k^j = Pr\{z_k = j|I_k\}$ is the probability of state being j given the current information vector I_k. In the appendix it is proved that the belief vector is a sufficient statistic for the control problem. The update rule T for $P_{k+1} = [p_{k+1}^1, p_{k+1}^2, \cdots, p_{k+1}^{2^n}]$

$$T(P_k|v_k, \theta_{k+1}) = \left[\cdots, \frac{\sum_{i\in\mathcal{S}} p_k^i a_{ij}(v_k).r_{j,\theta_{k+1}}^{v_k}}{\sum_{j\in\mathcal{S}}\sum_{i\in\mathcal{S}} p_k^i a_{ij}(v_k).r_{j,\theta_{k+1}}^{v_k}}, \cdots\right]. \; j = 1, 2, \cdots, 2^n. \tag{4.3}$$

is also proved. In other words, knowledge of the current value of the belief vector, the current control and the next output is sufficient to determine the value of the belief vector at the next time step.

B. Cost to Go Functions

Analogous to the perfect information case we get the solution in terms of minimizing cost to go functions. The cost to go functions now become a function of the current belief vector instead of the current state. Let $J_k(P_k)$ be the optimal cost to go at time step k with belief vector P_k. Then

$$J_k(P_k) = \min_{v_k\in\mathcal{A}} \sum_{i\in\mathcal{S}} p_k^i \left[C_k(i, v_k) + \sum_{j\in\mathcal{S}} a_{ij}(v_k)\{\sum_{\theta\in\mathcal{Q}} r_{j\theta}^{v_k} J_{k+1}(T(P_k|\theta, v_k))\}\right] \tag{4.4}$$

This is obtained by considering the expectation of the immediate control cost, and the expectation of the costs J_{k+1} over the all possible next states and observations. In the network we consider the control is external and thus independent of the state and the time i.e $C_k(i, v) = C(v)$. Also the type of observation we consider are independent of the control action at previous time step and just a function of the current state i.e. $r_{j\theta}^{v_k} = r_{j\theta}$. Thus equation (4.4) becomes

$$J_k(P_k) = \min_{v_k \in \mathcal{A}} \left[C(v_k) + \sum_{i \in \mathcal{S}} p_k^i \sum_{j \in \mathcal{S}} a_{ij}(v_k) \sum_{\theta \in \mathcal{Q}} r_{j\theta} J_{k+1}(T(P_k|\theta, v_k)) \right] \tag{4.5}$$

The terminal cost to go function is the expectation of terminal penalty.

$$J_M(P_M) = \sum_{i \in \mathcal{S}} p_M^i C_M(i). \tag{4.6}$$

C. 3 Gene PBN Example

To illustrate the algorithmic details, we consider the 3-gene network in Figure 3 discussed in chapter III. As before we consider x_3 to be the penalty gene with a terminal penalty of +5 being assigned when $x_3 = 1$ in a state. x_1 is the control gene and the control action is a forcible flipping of this gene.

We consider $M = 2$ step control, under 3 different observation schemes (i) gene x_1 is perfectly observable, (ii) a noisy version of x_1 is available (iii) No observation. The vector $r_{j\theta}$ for the different cases are plotted in figure 8. The expected costs are tabulated in Table IV starting from each initial state.

Table IV. Table of optimal costs for an M=2 step control.

Observation Model	Initial State i							
	1	2	3	4	5	6	7	8
Full State Feedback	0.0	0.33	0.0	0.66	0.66	1.0	0.33	1.0
(i)	0.0	0.33	0.0	0.66	0.66	1.0	0.33	1.00
(ii)	0.0	1.40	0.0	1.60	1.00	1.0	1.00	1.00
(iii)	0.0	1.66	0.0	2.0	1.00	1.0	1.00	1.00
No Control	0.0	1.66	0.0	3.33	3.33	5.0	1.66	5.00

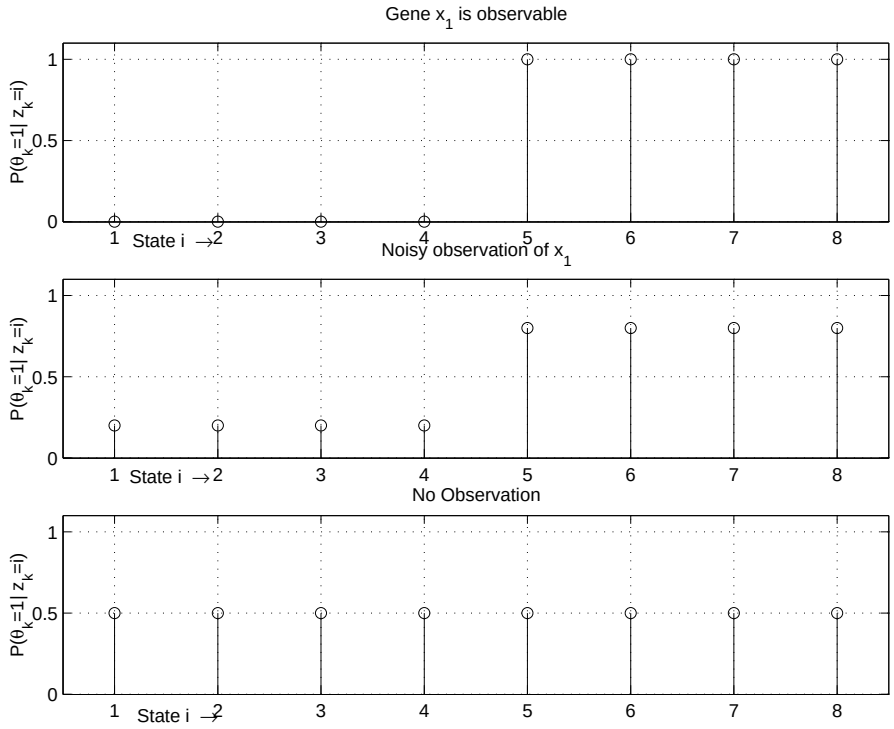

Fig. 8. $P(\theta_k = 0|z_k = i)$

D. Metastatic Melanoma Application

In this section, we apply the methodology of this chapter to derive an optimal intervention strategy for a network obtained from the study of metastatic melanoma discussed in chapter III.

For each gene in this network, we determined their two best two-gene predictors and their corresponding COD's. Using the procedure discussed in [4], the COD information for each of the predictors was then used to determine the $2^7 \times 2^7$ matrix of transition probabilities for the Markov Chain corresponding to the dynamic evolution of the gene-activity profile of the seven gene network. The transition probability matrix $A(v(k))$, the probability distribution of the observations given the current state and the immediately prior control ($r^v_{j\theta}$), and the initial state probability distribution vector (P_0) together constitute the data needed for setting up the optimal control problem in the presence of imperfect state information. In our construction, the vector $r^v_{j\theta}$ for θ, does not depend on the prior control input v and probabilistically relates only to the current state of the network. This relationship is shown in Figure 9 and it closely mimics the behavior of a gene MMP-3 which appears in the 10-gene network (Figure 6) but does not appear in the 7-gene network (Figure 7).

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 chose a window of length 5, i.e. the control inputs would be applied only at time steps 0, 1, 2, 3 and 4. The terminal penalty at time step 5 was chosen as follows. Since our objective is to ensure that WNT5A is not up-regulated, we assigned a penalty of zero to all states for which WNT5A equals 0 and a penalty of 3 to all states for which WNT5A equals 1. Here the choice of the number 3 is somewhat arbitrary but

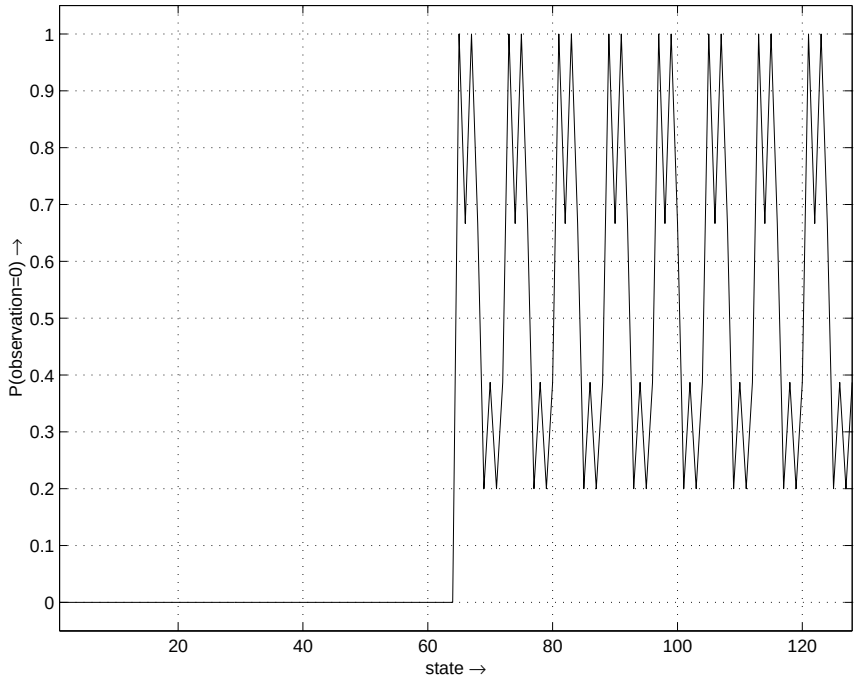

Fig. 9. Plot of probability{observed variable $\theta = 0$} versus the current state

it does reflect our attempt to numerically capture the biological notion that states where WNT5A equals 1 are less desirable than those where WNT5A equals zero. The cost of intervention is 1.

We next 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. 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 no WNT5A inhibitory protein is used while 1 indicates that such an intervention has been applied. The one step cost of control is taken to be equal to the value of the control variable. 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 depending on the

Table V. Expected costs for different initial state distributions

$Pz0$	Control, Observation	Control, State feedback	No Control
P_{data}	0.4079	0.3226	0.9677
$\left[\frac{1}{128}, \frac{1}{128}, \ldots\right]$	0.7068	0.3395	0.9990
$\left[0, \frac{1}{64}, 0, \frac{1}{64}, \ldots\right]$	0.7296	0.3395	0.9990
$\left[\frac{1}{64}, 0, \frac{1}{64}, 0, \ldots\right]$	0.5692	0.3395	0.9990

initial distribution P_0 and the subsequent total information vector I_k. Note that unlike the perfect information scenario considered in [7], 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.

Using the algorithm in [20] we can find the complete solution to this optimal control problem. Unfortunately that would involve spanning a $2^7(= 128)$ dimensional probability distribution space. Instead we used (4.6), and (4.5) recursively to calculate the optimal controls for certain given initial state probability distributions. The net result, in each case, was a tree with optimal control action followed by branches corresponding to subsequent observation.

Starting with P_{data}, the distribution of states in the 31 point data set, we found the optimal expected cost based on imperfect information to be 0.4079. The corresponding optimal cost using full state observation as in [7] was found to be 0.3226. The expected cost incurred by not using any control was 0.9677. We computed these quantities for a few different cases of initial state distributions. The relevant quantities are tabulated in Table V.

We also calculated the optimal expected costs when the initial state is determin-

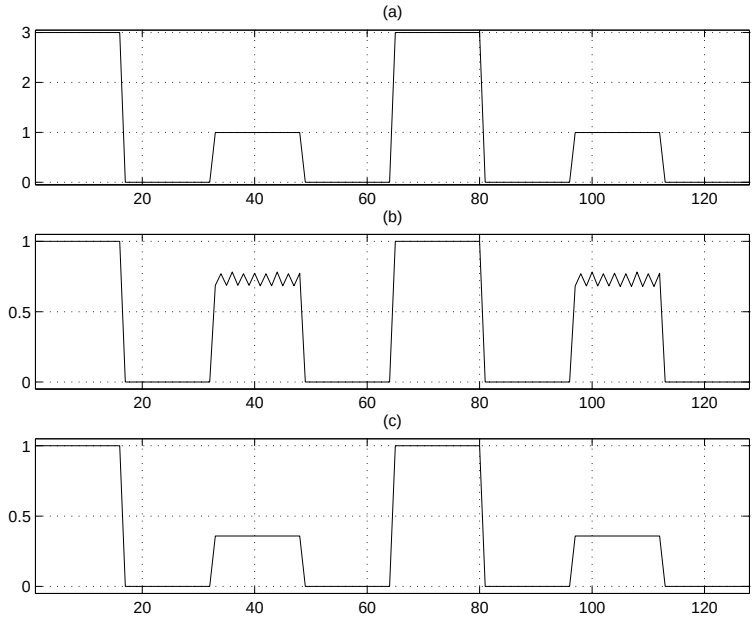

Fig. 10. Optimal expected cost versus initial states (a) uncontrolled (b) control using imperfect information (c) control using full state information

istic. These values for all the 128 possible initial states are shown in Figure 10.

Note that, 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 zero at the final time step, i.e. at $k = 5$. This quantity was computed for each (deterministic) initial state for both the uncontrolled and imperfect-information-based controlled cases. These plots are shown in Figure 11.

From this figure, it is clear that the control strategy for each initial state is increasing the probability for WNT5A equal to zero at the terminal time point relative to the corresponding probability in the uncontrolled case. This is, indeed, a desirable outcome achieved by using control.

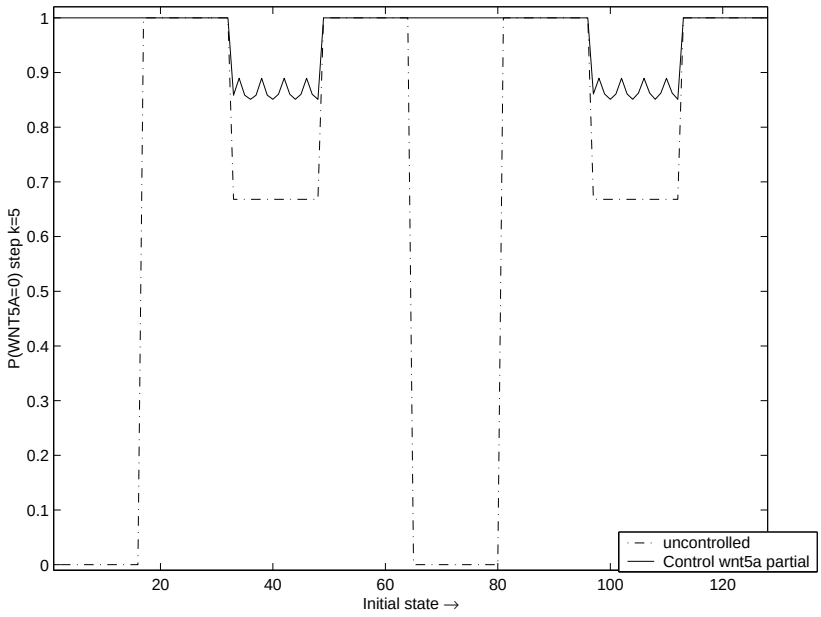

Fig. 11. Probability of WNT5A= 0 at the terminal time point versus the initial state for the uncontrolled and imperfect-information-based controlled cases

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. 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 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 implemented the algorithm with pirin as the control.

We found that using pirin as a control is totally ineffective. The expected cost, with pirin as the control, was found to be the same as the one obtained in Table V with no control. Even with full state feedback we still found that pirin was as ineffective

as before (data not shown). This is in stark contrast to our results in [7],Chapter III where we demonstrated the feasibility of doing full state feedback control of WNT5A through pirin. It is possible that going from a ternary setup in [7],Chapter III to the binary setup here may have drastically reduced our ability to control WNT5A through pirin. This suggests that sophisticated procedures need to be developed to reduce the number of states, that preserve properties like controllability and observability.

E. Conclusions

In this chapter, we have extended our earlier results on external control in Markovian genetic regulatory networks to the case where perfect information about the state of the network is not available. In such a situation, the optimal control must be designed based on the available measurements, which are assumed to be probabilistically related to the state of the genetic regulatory network. The conditional probability measure of the state, given the information, serves as a sufficient statistic for computing the optimal control.

CHAPTER V

ASSIGNMENT OF TERMINAL PENALTIES

In this chapter we provide an algorithm for assigning terminal penalties, (an issue not discussed in [7]) by taking the long term uncontrolled behavior into account. We also discuss the possibility of using gene influence for pre selection of genes to be used for intervention.

A. Attractors

Attractors play a key role in Boolean networks. Given a starting state, within a finite number of steps, the network will transition into a cycle of states, called an *attractor cycle*, and will continue to cycle thereafter. Non-attractor states are transient and are visited at most once on any network trajectory. The *level* of a state is the number of transitions required for the network to transition from the state into an attractor cycle. Attractors are often identified with phenotypes [3]. Real biological systems are typically assumed to have short attractor cycles. Singleton attractors are a key interest since these are associated with phenomena such as cell proliferation and apoptosis [12].

The key objective of intervention in BNs/PBNs is to steer the network from an undesirable attractor to a desirable attractor. I.e if a state is an undesirable attractor or in the basin of one, it should have a higher terminal penalty, since by stopping the control in such a state the network would transition to undesirable attractor and stay there ever after.

B. Terminal Penalty J

In [7] penalties were assigned to states based on the expression level of certain key genes which we call *penalty genes*. In particular we used WNT5A a gene known to be over expressed in metastatic melanoma.

We now present a more sophisticated procedure for terminal penalty assignment by looking at the long term prospective behavior of the system in the absence of control. Though this procedure was worked out in [21] for any Markov chain, it is particularly suited for applications on biological networks, that have few singleton attractors.

- Partition the states of the Markov chain into transient and persistent states.
- For singleton attractors the penalty J is set according to the status of the penalty gene or genes, e.g. for the Markov chain in Figure 12 the penalty gene is gene No.3 and if the gene is upregulated, the corresponding state penalty is +3.
- For a cycle the penalty is based on the fraction of time spent in states having penalty gene or genes in undesirable profile.
- For a transient state j, the penalty $J(j) = \sum_i P(S_\infty = i|S_t = j).J(i)$, where i is a cycle or a singleton attractor.

We illustrate this procedure using the following example

Consider the Markov chain in Figure 12, with upregulated penalty gene No.3 with a penalty 3. There are two persistent equivalence classes. Attractor $\{000\}$ with penalty 0 and cycle $\{100, 111\}$ with penalty $1/3 \times 0 + 2/3 \times 3 = 2$ corresponding to the stationary distribution $\pi = [1/3, 2/3]$ of states $\{100, 111\}$.The penalties are listed

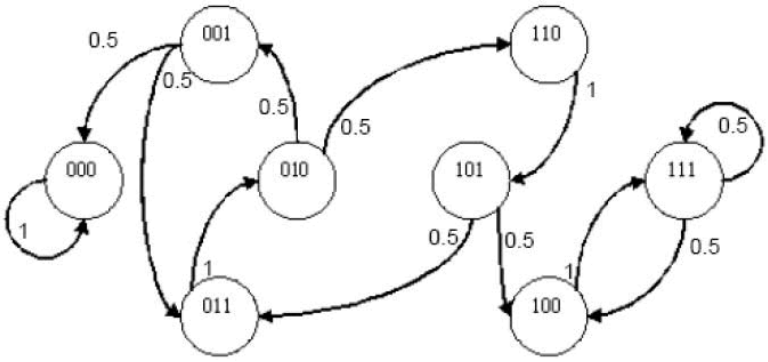

Fig. 12. Markov chain for a 3 gene PBN, $x_3 = 1$ is penalized with +3

in Table VI. The quantities P($\{000\}$)and P($\{100, 111\}$) are the probabilities of falling in the attractor $\{000\}$ and the cycle $\{100, 111\}$, respectively.

Table VI. Terminal penalty J_{eq} is based on the procedure in section B. J_s is based on the instantaneous state profile. Upregulated gene No.3 is used as the penalty gene with weight +3.

State	P($\{000\}$)	P($\{100, 111\}$)	J_{eq}	J_s
000	1	0	0	0
100	0	1	2	0
010	0.5	0.5	1	0
110	0.25	0.75	1.5	0
001	0.75	0.25	.5	3
101	0.25	0.75	1.5	3
011	0.5	0.5	1	3
111	0	1	2	3

A particular advantage of using the above procedure is that starting from any initial state, we can say that using more control steps is never disadvantageous. This is proved in section C.

C. Cost Function and Number of Control Steps

We now present a proof by induction of the fact that by doing the penalty assignment using the procedure in section B the cost function J is a non-increasing function of the number of control steps used, under some reasonable assumptions.

To do so, we first make the following observations/assumptions:

- From the definition of the terminal penalties, the following relationship holds

$$J_M(i) = \sum_{j \in \mathcal{S}} a_{ij}(0) J_M(j) \tag{5.1}$$

- In equation (5.1) the control input $v = 0$ corresponds to $u = [0, 0, \ldots, 0]$, the case with no control input i.e. autonomous evolution. Furthermore $C_k(i, 0) = 0$, since it is the cost of applying no control input.

- The cost of applying control is stationary and non-negative i.e. $C_k(i, v) = C(i, v)$ and $C(i, v) \geq 0$ for all $v \in \mathcal{A}$.

We now prove that the cost function for a 1 step procedure is less than that of a 0 step procedure. For any $i \in \mathcal{S}$ consider $J_{M-1}(i)$, the one step value function.

Then from (3.7),

$$J_{M-1}(i) = \min_{v \in \mathcal{A}} (C(i, v) + \sum_{j \in \mathcal{S}} a_{ij}(v).J_M(j)) \tag{5.2}$$

$$= \min(\min_{v \in \mathcal{A} - \{0\}} (C(i, v) + \sum_{j \in \mathcal{S}} a_{ij}(v).J_M(j)), C(i, 0) + \sum_{j \in \mathcal{S}} a_{ij}(0).J_M(j)) \tag{5.3}$$

In view of (5.1) and $C(i, 0) = 0$, we have

$$J_{M-1}(i) = \min(\min_{v \in \mathcal{A} - \{1\}} (C(i, v) + \sum_{j \in \mathcal{S}} a_{ij}(v).J_M(j)), J_M(i)) \tag{5.4}$$

i.e. we have

$$J_{M-1}(i) \leq J_M(i) \tag{5.5}$$

By process of induction, assume this to hold true for an $M-k-1$ step procedure, i.e.

$$J_{K+1}(i) \leq J_{K+2}(i) \tag{5.6}$$

Now from (3.7), we have

$$J_{K+1}(i) = \min_{v \in \mathcal{A}}(C(i,v) + \sum_{j \in \mathcal{S}} a_{ij}(v).J_{K+2}(j)) \tag{5.7}$$

Let v^* be an input that attains this minimum i.e.

$$J_{K+1}(i) = (C(i,v^*) + \sum_{j \in \mathcal{S}} a_{ij}(v^*).J_{K+2}(j)) \tag{5.8}$$

Now consider the step K:

$$J_K(i) = \min_{v \in \mathcal{A}}(C(i,v) + \sum_{j \in \mathcal{S}} a_{ij}(v).J_{K+1}(j)) \tag{5.9}$$

$$= \min(\min_{v \in \mathcal{A}-v^*}(C(i,v) + \sum_{j \in \mathcal{S}} a_{ij}(v).J_{K+1}(j)), C(i,v^*) + \sum_{j \in \mathcal{S}} a_{ij}(v^*).J_{K+1}(j)) \tag{5.10}$$

$$\Rightarrow J_K(i) \leq C(i,v^*) + \sum_{j \in \mathcal{S}} a_{ij}(v^*).J_{K+1}(j) \tag{5.11}$$

Now using (5.8) we get,

$$J_K(i) - J_{K+1}(i) \leq \sum_{j \in \mathcal{S}} a_{ij}(v^*)\{J_{K+1}(j) - J_{K+2}(j)\} \tag{5.12}$$

Now using (5.6) we have $J_{K+1}(j) \leq J_{K+2}(j)\ \forall j \in \mathcal{S}$,

$$\Rightarrow J_K(i) \leq J_{K+1}(i) \tag{5.13}$$

Hence for any initial state $i \in \mathcal{S}$, the value function $J_K(i)$ is a non increasing function of the number of control time steps used.

D. 3 Gene PBN Example

Let us consider the problem of assigning terminal penalty and optimal control for the problem in Figure 3. The long probabilities of falling into attractors 1 and 6 are shown in figure 13. For $J(6) = 5$ and $J(1) = 0$ the terminal penalties are shown in figure 14.

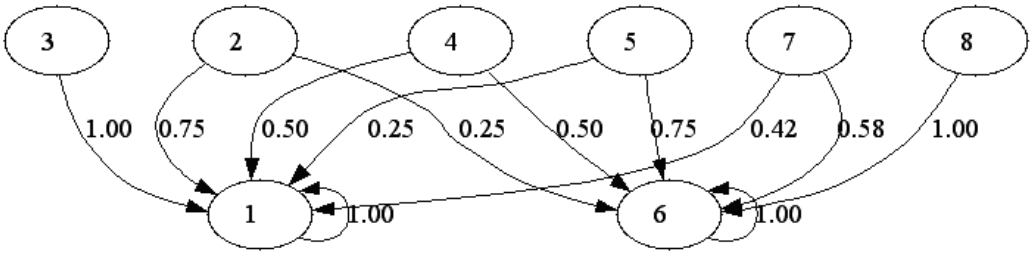

Fig. 13. $P(z(\infty) = j|z(0) = i)$

With cost of control $C(.,1) = 1$, $C(.,0) = 0$ as before the optimal expected cost as a function of time horizon used is shown in figure 15.

E. Selection of Genes for Intervention

For the purposes of intervention, in theory we could flip a number of genes. However from a biological perspective we would want the intervention to be minimal. Thus it makes sense to choose a particular gene, that is likely to be the most effective in bringing about the desired intervention.

In principle the optimal control problem could be solved for each gene and then the best gene chosen. However this would be a computationally demanding procedure.

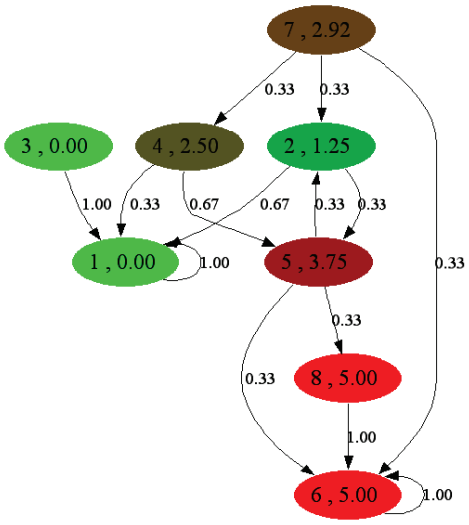

Fig. 14. State, terminal penalties are shown in the oval. Notice that (5.1) is satisfied

Here we suggest two different heuristic approaches for gene selection and compare their performance for the WNT5A example. These two approaches are based on (1) gene influence and (2) a one step control(with 0 control cost).

Gene influence is a property of the underlying PBN and depends only on the state distribution. It is independent of the cost of control, terminal penalties and time steps. This is unlike the optimal control problem which would have to be solved every time the cost functions are changed; gene influence has to be calculated only once. We could use gene influence to narrow down the pool of genes, that can then be studied using dynamic programming. We next present the formal definition of influence.

F. Influence

Gene influence as a possible way of quantifying the relative importance of different predictor genes on a target was introduced in [4]. The *influence* $I_j(f)$ of the gene x_j on the Boolean function f, with respect to a probability distribution of states $D(x)$

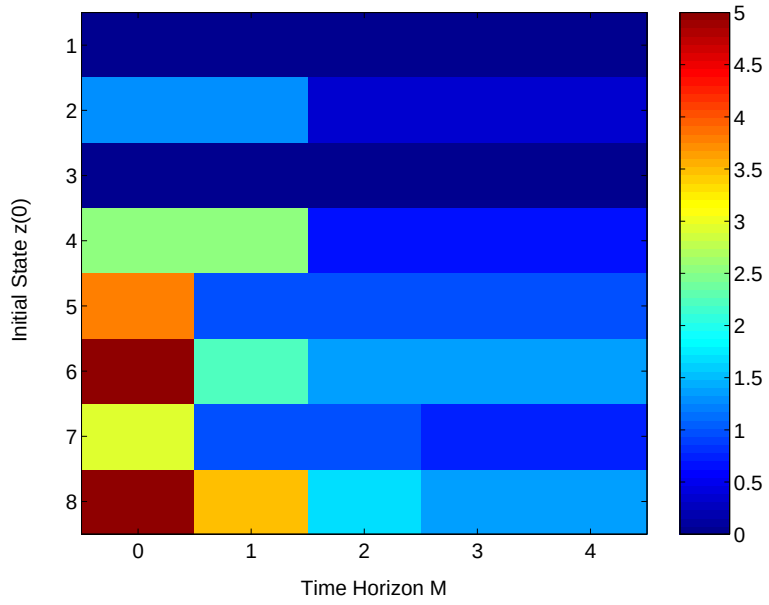

Fig. 15. Expected cost as a function of time horizon from each initial state

is defined as

$$I_j(f) = E_D\left[\frac{\partial f(x)}{\partial x_j}\right] \tag{5.14}$$

where E is the expectation operator, $\frac{\partial f(x)}{\partial x_j}$ is defined as $f(x) \oplus f(x^j)$ and x^j is defined as $(x_1, x_2, \ldots, x_{j-1}, x_j \oplus 1, x_{j+1}, \ldots, x_n)$. Essentially influence is the weighted average over states of the change in the value of function f in the event of the flipping of a variable. In the context of PBN's the influence of gene x_k on gene x_i becomes

$$I_k(x_i) = \sum_{j=1}^{l(i)} I_k(f_j^{(i)}).c_j^{(i)} \tag{5.15}$$

The influence matrix Γ has entries $\Gamma_{ij} = I_i(x_j)$. Also by taking the row sum we can find Γ_i which is the influence of the gene x_i on the network in general under the state distribution D. Under perfect observation D is degenerate, with Γ easy to calculate and interpret.

G. Metastatic Melanoma Application

In this section, we apply our methods to a network developed from data collected in a study of metastatic melanoma [13] discussed in chapter III.

The website [22] shows this 10 gene network and provides insights to the determination of the $2^{10} \times 2^{10}$ matrix of transition probabilities for the Markov Chain corresponding to the dynamic evolution of the gene-activity profile of the 10 gene network. The predictors and functions were determined from the data using COD analysis.

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.

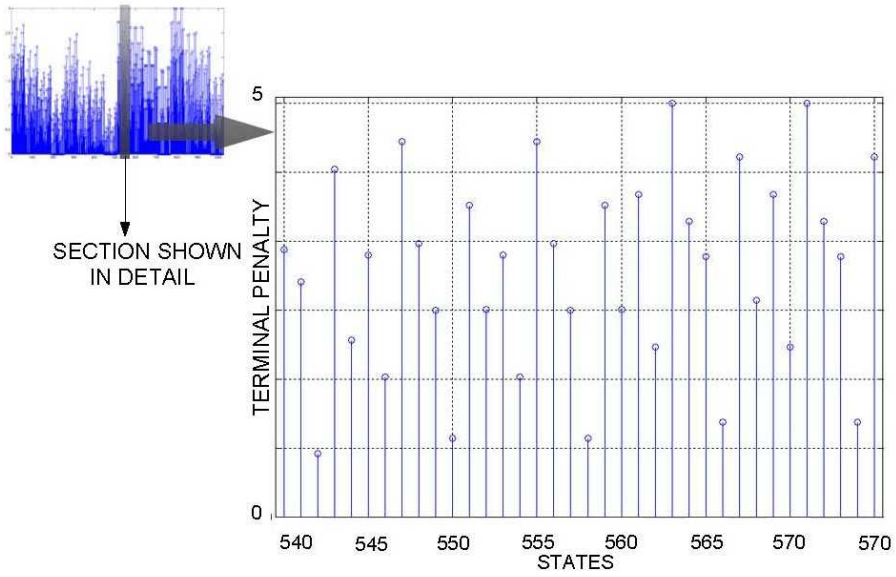

Fig. 16. Terminal penalty with WNT5A as the penalty gene

We next discuss two different aspects of the control scheme: We used the procedures in section B and [7] to assign the terminal penalties J_{eq} and J_s respectively, using WNT5A as a penalty gene with a penalty of +5, as shown in Figure 16. In

the optimal control problem, we used gene 1 (PIRIN) for intervention purposes. In particular consider the state 791 a data point corresponding to $[0, 1, 1, 0, 1, 0, 0, 0, 1, 1]$ (LSB$\rightarrow$ MSB) as the initial state. For the scheme based on states we observe that the value function J_s is not monotonic. Nevertheless we observe that after a certain number of steps the expected cost function decreases monotonically (12 steps in this case). This lack of monotonicity complicates the problem of selection of an appropriate control horizon particularly if the control horizon cannot be too large. We believe that the number of steps upto which the oscillations occur is related to the distance of the states in the network from the attractors. This is a topic still under investigation.

Using the terminal penalty based on equivalence classes mitigates this problem. It is guaranteed that starting from any initial state, using additional control steps, we cannot do any worse even in the short term (Figure 17).

One of the ten genes is to be preselected to be used as control. At each time step the control action is chosen according to equation (3.7) as either flipping that gene or leaving it as is. We found that genes 1(PIRIN), 2(WNT5A) itself and 8(HADHB) dominate other genes in reducing the expected cost after 5 steps of control from any of the 2^{10} initial states. However there is no one particular gene that performs better than other genes for all initial states. This is clear from Figure 18.

This motivated us to use the rank expectation to rank the genes. We used a uniform distribution over

- S: All $2^{10} = 1024$ states.
- S_{DATA}: States in the dataset.
- $S_{DATA\_WNT5A=1}$: States in dataset with WNT5A upregulated(9 in number).

In general we observed that the influence heuristic performs better if the number of states over which the ranks are averaged are in particular, the states which need

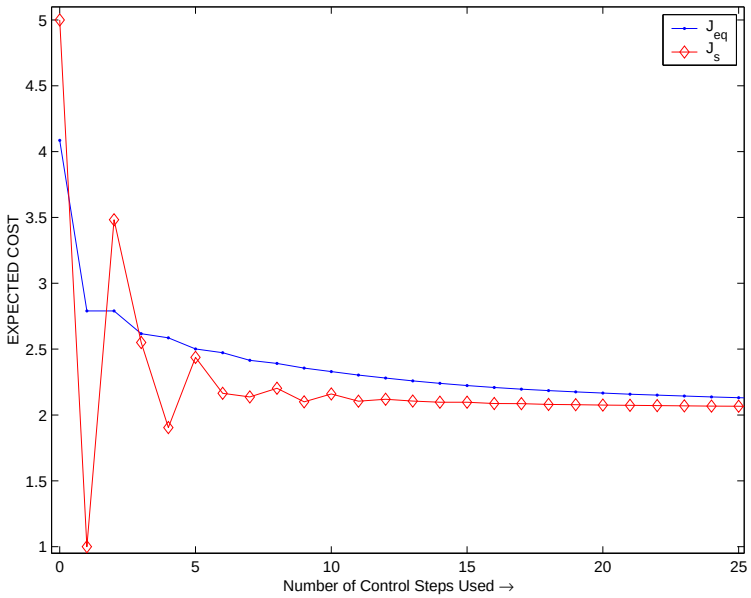

Fig. 17. With initial state 791 the expected cost is plotted for control with two different types of terminal penalty assignments, J_s based on the individual states and J_{eq} based on equivalence classes

more intervention. The heuristic does not perform well when we use averaging over all states since the majority of states need very little or no intervention.

We also found that gene influence was very effective in ruling out genes that should not be used for intervention. For the WNT5A network we discovered that the set of genes with least influence matched very closely the set of genes which were least effective when used for intervention. In particular the set of genes ranked in the bottom 20% by influence matched the set ranked by expected cost reduction in the 5 step optimal control with an accuracy ranging from $50 - 100\%$ for all states. We display the detailed results on the companion website [22].

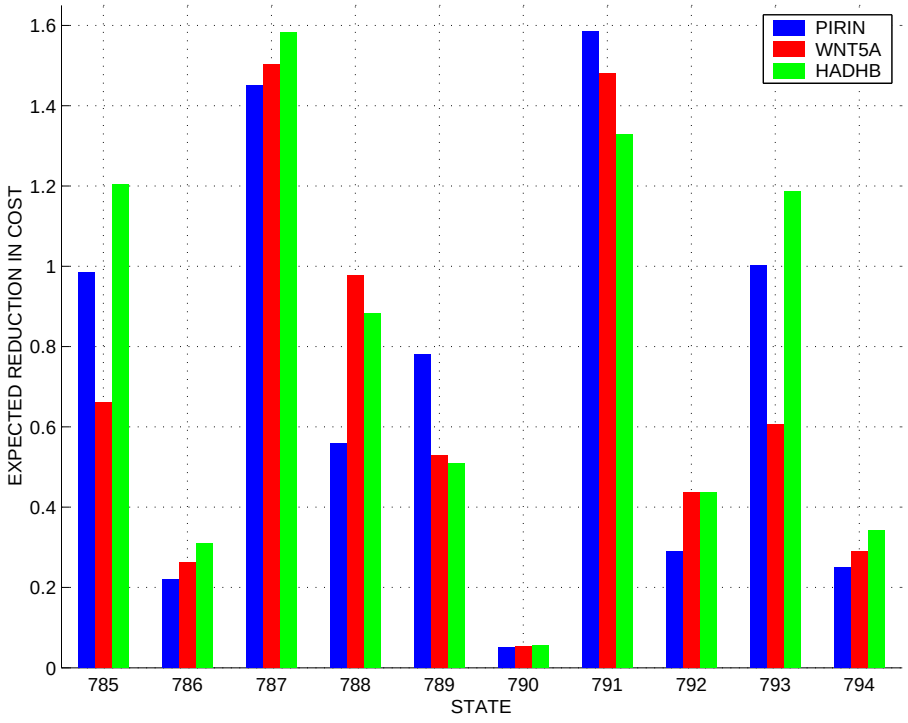

Fig. 18. Expected reduction in cost by using control for 5 time steps. Observe that different genes dominate when 787, 788 and 789 are the initial states

H. Conclusions

In this chapter we have refined our method of assignment of terminal penalties based on the individual state profile (Chapter III and [7]) by using equivalence classes of states. We also proved that such a terminal penalty assignment ensures that using more control steps produces better results, something that is not necessarily true for individual state based assignment in the short run.

We also introduced gene influence as a simple heuristic to narrow down the pool of candidate genes to be used for intervention purposes by selecting genes with high influence or more so by rejecting genes with low influence. This is important since the states in the network grow exponentially with the number of variables, and it may not be possible to check all candidate genes using the dynamic programming approach.

CHAPTER VI

FAMILY OF NETWORKS

Given a data set consisting of gene-expression measurements, PBN design constitutes an ill-posed inverse problem that is treated by using a design algorithm to generate a solution. Inference can be formalized by postulating criteria that constitute a solution space for the inverse problem. The criteria come in two forms: (1) the *constraint criteria* are composed of restrictions on the form of the network, and (2) the *operational criteria* are composed of relations that must be satisfied between the model and the data. The solution space consists of all PBNs that satisfy the two sets of criteria. Recognizing that PBNs are composed of Boolean networks, and since it is difficult to infer the probabilistic structure among the constituent Boolean networks from the steady-state data typically used for design, a more general view may be taken in which the inverse problem is restricted to determining a solution space of Boolean networks and then finding networks in that space [23]. Without a probabilistic structure between the Boolean networks, we have a family of Boolean networks satisfying both the constraint and operational criteria. If desired, one can then go further and construct a PBN by using networks from the family, or one can simply treat the family as a collection of solutions to the Boolean-network inverse problem.

In this chapter, we derive a control algorithm that can be applied to a family of Boolean networks. This is accomplished by minimizing a composite cost function that is a weighted average cost over the entire family. Ideally, the weighting for each member of the family at any time point would be proportional to the instantaneous probability of a particular network being the governing network. Although these instantaneous probabilities are not known, we adaptively estimate them from the

available data and the estimate is used to implement the control algorithm.

A. BN Design Algorithm

In most cases we lack time-course gene-expression measurements correspondng to the temporal evolution of the network, and our assumption is that the measurements (or almost all of them) are taken in the steady state [24, 9, 23] – see [23] for a discussion of the biological considerations concerning the steady-state assumption. Under this assumption data states are, with probability near one, attractor states. Thus, we would like them to be attractors in the model and their inclusion or lack of inclusion in a designed network can be used to support or not support network validity. If we take the view that there is no reason to believe that data states are not attractor states, then we may wish to require that the attractor states of a designed network exactly match the data states. To achieve this end, an algorithm has been developed to generate Boolean networks with a prescribed attractor structure [23]. To look for biologically meaningful networks in the space of desired networks, the number of predictors for a gene and the number of levels for a transient state are bounded. To avoid the inclusion of non-regulating genes, each gene must occur in the predictor set of at least one other gene. The algorithm can function in two modes. In one, it begins with genes, attractor states, predictor sets, and a maximum number of levels, and generates all possible networks having these; in a more general mode, it begins with genes, attractor states, a maximum predictor-set size, and a maximum level, and generates networks having these.

Table VII. Table of functions. Some parts have been filled using the information on attractors. Distinct networks obtained by assigning values to $\mathbf{a} = [a_1, \cdots, a_8]$ N_1 **a**=[0,0,1,1,0,1,0,0] N_2 **a**=[0,0,1,0,1,1,0,0] N_3 **a**=[1,0,0,1,0,1,0,1]. See Figure 2.

x_1	x_2	x_3	f_1	f_2	f_3
0	0	0	0	0	0
0	0	1	a_1	a_7	0
0	1	0	a_2	0	0
0	1	1	a_3	a_7	0
1	0	0	a_4	a_8	1
1	0	1	1	0	1
1	1	0	a_5	a_8	1
1	1	1	a_6	0	1

B. BN Design Example

Let us consider Boolean networks with three 3 genes x_1, x_2, x_3. Consider states {000},{101} as attractors and predictor sets $\mathcal{P}_1 = \{x_1, x_2, x_3\}$,$\mathcal{P}_2 = \{x_1, x_3\}$ and $\mathcal{P}_3 = \{x_1\}$. Now using the attractor structure we could fill out some parts of the truth Table VII leaving variable $a_1, \cdots, a_8$ to be randomly assigned. Three 3-gene networks with the given singleton attractors, predictor sets, and maximum level were shown in Figure 2 using decimal representation.

C. Dynamic Programming over a Family of Networks

If a family of BNs is designed whose attractors match the data, assuming the family is not too small we have the expectation that the underlying biological phenomena are closely modeled by at least some of the BNs in the family. In the absence of

perfect knowledge as to which BNs are capable of better representing the underlying phenomena, we develop a control policy that optimizes a composite cost function over the entire family of BNs.

Let $\mathcal{N}$ be a set of L Boolean networks $N_1, N_2, \ldots, N_L$ possessing identical sets of singleton attractors, all sharing the same state space $\mathcal{S}$ and the same control space $\mathcal{A}$. Associated with each network is an initial probability of it representing the underlying phenomenon. Since this information is not available, we will adaptively estimate these probabilities as more transitions are observed. For each network N_l, $l = 1, 2, \cdots, L$ define:

- $a_{ij}^l(v)$ to be the ith row, jth column entry of the matrix $A^l(v)$ of the network N_l;
- $C_k^l(i, v)$ to be the cost of applying the control v at the kth time step in state i in network N_l;
- $C_M^l(i)$ to be the terminal cost associated with state i in network N_l.

We define the *belief vector* $\pi_k = [\pi_k^1, \pi_k^2, \ldots, \pi_k^L]$, where π_k^l is the probability of network N_l being the underlying network at the kth time step. π_k is the probability distribution vector for the family of networks at the kth time step. Since π_k is unknown, we will make an initial guess for it and update it as more information becomes available. The use of this vector is inspired by the *information vector* in [20].

Suppose i is the current state at step k, π is the current estimate of the belief vector, and upon application of control v we observe state j at the next time step. Then the new belief vector is $\pi' = T(\pi, i|j, v)$, where the transformation T can be

obtained by use of Bayes' theorem and the theorem of total probability,

$$\pi' = [\cdots, \frac{a^l_{ij}(v).\pi^l_k}{\sum_{s\in\mathcal{N}} a^s_{ij}(v).\pi^s_k}, \cdots] \tag{6.1}$$

The optimal control problem over a family of networks problem is equivalent to solving(3.5),(3.6) under imperfect state information. This equivalence is proved in [25]. As before the solution can be presented as minimization of the of the cost to go function J.

Suppose we are given an initial belief vector π_0 and an initial state $z(0)$. The initial belief vector is based on our prior knowledge of the system. It could be a function of likelihood or Bayesian scores of networks, or it could be uniform to reflect no prior knowledge. Our objective is to find controls $v(0), v(1), \cdots, v(k), \cdots, v(M-1)$ to minimize the expectation of the cost-to-go function over all networks in $\mathcal{N}$. The cost to go function J the kth time step ($0 \leq k < M$) is a function of the current state $z(k)$ and the updated belief vector π_k and is given by

$$J_k(\pi_k, i) = \min_{v\in\mathcal{A}}[\sum_{l\in\mathcal{N}} \pi^l_k\{C^l_k(i,v) + \sum_{j\in\mathcal{S}} a^l_{ij}(v).J_{k+1}(T(\pi_k, i|j, v), j)\}] \tag{6.2}$$

Intuitively, the inner summation is the expectation over all $j \in \mathcal{S}$ of the cost to go at the $(k+1)$th step in the lth network on observing j. We then add to it the cost of control at the kth step and average over all the networks in the family. Finally we take the minimum over all control actions in $\mathcal{A}$ to obtain the optimal policy and the cost to go at the kth step.

The terminal cost for a state i is trivially defined to be the average terminal cost over the entire family:

$$J_M(\pi_M, i) = \sum_{l\in\mathcal{N}} \pi^l_M.C^l_M(i). \tag{6.3}$$

The terminal penalties are assinged using the procudure discussed in chapter V. Since the attractors are shared by each network in the family, the attractor states will have the same penalty across the different networks; however, penalties for non-attractor states will differ across networks, depending on the particular attractor in whose basin that non-attractor state may happen to lie in.

A version of this work that relates [7] and [21] is available in [11].

D. Implementation

The solution to the minimization problem (6.2), (6.3) will now be presented as a policy tree that is optimal specific to a particular initial state and an initial belief vector. An M -step policy tree has an optimal action as its root with branches for each possible observation (in our case states) followed by M-1 step policy trees. A detailed exposition on construction and pruning of such trees can be found in [26]. Here we state an algorithm that is close to exhaustive enumeration and subsequent pruning. We use a data structure *node* with five components $STATE$, $BELIEF\text{-}VECTOR$, $OPTIMAL\text{-}COST$, $OPTIMAL\text{-}CONTROL$ and $DEPTH$. The algorithm involves the following steps:

1. Compute the M step control and the corresponding $J_0(i), J_1(i), \cdots J_M(i)\ \forall i \in S$ for each of the networks $N_1, \cdots, N_L$ as a table. Table VIII is such a table of controls for the example to be presented later in the next section.

2. Initialize the tree's root node with the first observed state $STATE = z(0)$, $BELIEF\text{-}VECTOR = \pi_0$ and $DEPTH = 0$.

3. Expand the root node and all the subsequently generated nodes; while $BELIEF\text{-}VECTOR \neq e_l$ (i.e. the network N_l is not uniquely identified to be the underlying network) and $DEPTH \leq M$.

To expand a particular node in the tree with $STATE = i$, $BELIEF\text{-}VECTOR = \pi_k$ and $DEPTH = k$, we consider all possible states that could be observed next. In other words, a child node is created for any j, such that $a^l_{ij}(v) > 0$ with $\pi^l_k > 0$. Such a node has $STATE = j$, $BELIEF\text{-}VECTOR = T(\pi_k, i|j, v)$ and $DEPTH = k + 1$.

4. Now consider all the leaf nodes. For the nodes with $DEPTH = M$ we use (6.3) to obtain $OPTIMAL\text{-}COST$. For leaf nodes with $DEPTH = k \neq M$ and $BELIEF\text{-}VECTOR = e_l$ and some $STATE = i$, $OPTIMAL\text{-}COST$ is set to $J_k(i)$ from the table for network N_l.

5. Now use (6.2) for all nodes with $DEPTH = M-1, \cdots 0$ (in that order) to obtain $OPTIMAL\text{-}COST$ and the minimizing v as the $OPTIMAL\text{-}CONTROL$.

6. Prune the subtrees generated with non optimal actions to obtain Pol^{TR}, the policy tree. The optimal policy follows the table for network N_l onwards from a node which has $BELIEF\text{-}VECTOR = e_l$.

At first glance, generating the tree may seem to be a formidable task due to the potentially large branching factor which can be as high as $|\mathcal{S}| \times |\mathcal{A}|$. However, in the case of a family of BNs this is a much more manageable task due to the following mitigating factors: (1) the branching factor is small since not all states would be observed following an action due to similarities in the different BN transitions – for instance, in Figure 2 state 3 goes to state 1 in all the three networks; (2) from a given node, if more than one node is generated for some v, then the $BELIEF\text{-}VECTOR$s for the children would be more sparse than the parent, and in some cases it would become a leaf node with $BELIEF\text{-}VECTOR = e_l$; and (3) the set of possible control actions $\mathcal{A}$ is not large owing to the limited number of genes for intervention.

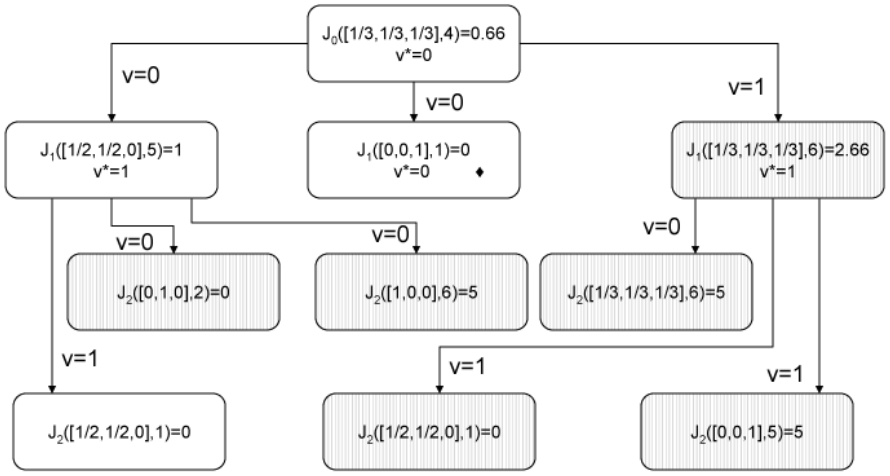

Fig. 19. Tree calculation for initial belief vector π_0=[1/3,1/3,1/3], initial state $z(0) = 4$. The shaded region is pruned. $\diamond$ is a leaf node at $DEPTH < M$

E. 3 BN Example

To illustrate the algorithmic details, we consider the 3-gene network introduced previously. Suppose state 1 (000) is a desirable attractor state with terminal penalty 0 and state 6 (101) is an undesirable state with terminal penalty +5. The terminal cost of any other non-attractor state is the cost of the attractor whose basin it is in. For instance for the network N_2, the nonattractor states 2, 3, 4 , and 5 have terminal penalty 0, while states 7 and 8 have terminal penalty +5. Let x_1 be the control gene and suppose that the control action is to forcibly flip this gene: for $v(k) = 1$, flip gene x_1 at the kth time step and for $v(k) = 0$ leave it as is. Let the cost of control $C_k^l(i, v) = C(v) = v$. Transitions take place according to the network transition rule – for example, in network N_2, if $z(k) = 6$ (101) and $v(k) = 1$, then $z(k + 1) = 1$, corresponding to a jump from state 6 (101) to state 2 (001) and then evolution to the state 1 (000). We show the evaluation of the policy tree Pol^{TR} starting from an initial state $z(0) = 4$ and π_0 =[1/3,1/3,1/3] in Figure 19. Figure 20 shows the

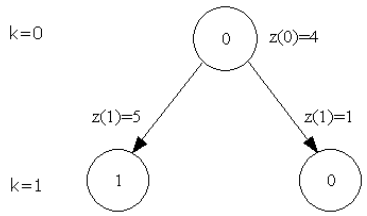

Fig. 20. Pruned policy tree. The number inside the circle is the optimal control action. The arc corresponds to the next observation, which leads to the next optimal control action.

corresponding policy tree obtained after pruning.

We now proceed to compare the performance of a $M = 2$ step policy Pol^{TR} and policies obtained using two other methods to control this family of networks.

Single network.-We calculate the optimal policy Pol^l for each network N_l in the family. We obtain the control policy as a table with M rows and $|\mathcal{S}|$ columns. Each element $v(m, i)$ is the control alternative to be used when the state is i at the mth time step. Since a single-network policy does not apply to the entire family, to implement a policy tree we follow one of the possible state observations after each action. It may happen that some of the possible states observed may not be listed as an option in a single-network policy tree. For a single BN the policy tree is a tree with a branching factor of 1, i.e. a path. Single network optimal policies for each network are listed in Table VIII.

Context Switching.- The context-sensitive PBN design of [9] is more general than the method proposed here because there is no requirement that the constituent BNs possess identical attractor structures; however, it is more constrained in the sense that it assumes knowledge of the PBN switching structure. If, as is assumed here, we lack knowledge of the probabilistic structure governing BN selection so that we do not view the family of BNs as composing a single PBN and if the attractor structures are identical, as with a design strategy in which the attractors of each BN match the

Table VIII. Optimal control policies obtained from different networks. N_{SW} is obtained for π=[1/3,1/3,1/3].

Network	Time Step(k)	State							
		1	2	3	4	5	6	7	8
N_1	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	1	1	0	0
N_2	0	0	0	0	0	0	0	0	0
	1	0	0	0	0	0	1	1	1
N_3	0	0	0	0	0	0	1	0	1
	1	0	0	0	0	1	0	0	1
N_{SW}	0	0	0	0	0	1	1	1	1
	1	0	0	0	0	1	1	1	1

data states, then it may well be that the policy proposed here could outperform the context-sensitive-PBN method. If, for the present 3-gene example, we assume that the BNs compose a PBN in which each has equal probability of being selected, then the method of [9] yields the optimal control policy, Pol^{SW}, presented in the last row of Table VIII.

To assess the performance of a particular policy Pol, we apply it to all the networks in the family starting from each initial state i and obtain $J_0^{l,Pol}(i)$. We then compute

$$J_0^{Pol}(i) = \sum_{l \in \mathcal{N}} J_0^{l,Pol}(i).\pi^l \tag{6.4}$$

by averaging over all the networks. Assuming that π_0=[1/3,1/3,1/3], Table IX shows the results of applying various policies for all possible initial states. As measured by the value of the optimal cost, the policy Pol^{TR} of this work is superior. More examples appear on the companion website [27].

Table IX. Performance of control with π_0=[1/3,1/3,1/3], $M = 2$.

Policy	$J_0^{Pol}(1)$	$J_0^{Pol}(2)$	$J_0^{Pol}(3)$	$J_0^{Pol}(4)$	$J_0^{Pol}(5)$	$J_0^{Pol}(6)$	$J_0^{Pol}(7)$	$J_0^{Pol}(8)$
Pol^1	0.00	0.33	0.00	0.66	2.00	2.66	0.33	2.66
Pol^2	0.00	1.66	0.00	1.66	0.66	2.66	0.33	2.66
Pol^3	0.00	0.33	0.00	0.66	2.00	1.33	1.66	1.66
Pol^{SW}	0.00	0.33	0.00	0.66	1.00	1.33	1.00	1.66
Pol^{TR}	0.00	0.33	0.00	0.66	0.66	1.33	0.33	1.66

F. Metastatic Melanoma Application

We now apply the methodology of this chapter to derive an optimal intervention strategy for a family of gene regulatory networks obtained from the study of metastatic melanoma discussed in Chapter III. We began with the binary 7 gene data.

Since all 31 data points correspond to steady-state behavior, they should be considered as attractors in the networks. However, out of the 31 samples only 18 were distinct. To reduce the number of attractors, we formed seven clusters from the data points and treated the cluster centers as attractors. These attractors are shown in Table X. The first column is used to classify them into two categories, GOOD and BAD, depending on the status of the WNT5A gene.

Using the procedure of [23], we obtained 4 distinct BN's (N_1, N_2, N_3, N_4) with the same set of 7 attractors.

Table X. Cluster centers as attractors for the WNT5A network. The good attractors are the ones with the profile of WNT5A gene downregulated. PIRIN is the most significant bit(MSB) and WNT5A is the least significant bit(LSB)

	z	Gene Activity Profile x						
		PIRIN	S100P	RET1	MART1	HADHB	STC2	WNT5A
B	4	0	0	0	0	0	1	1
A	32	0	0	1	1	1	1	1
D	82	1	0	1	0	0	0	1
G	33	0	1	0	0	0	0	0
O	57	0	1	1	1	0	0	0
O	95	1	0	1	1	1	1	0
D	109	1	1	0	1	1	0	0

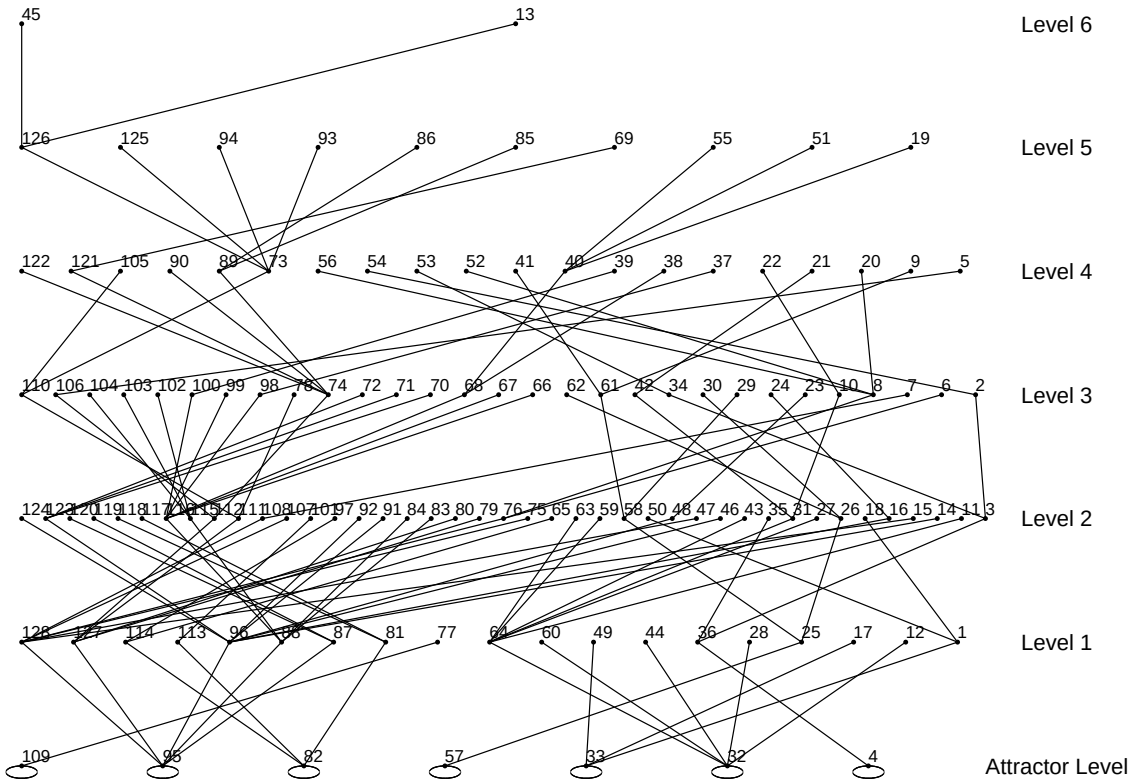

Fig. 21. WNT5A network N_1

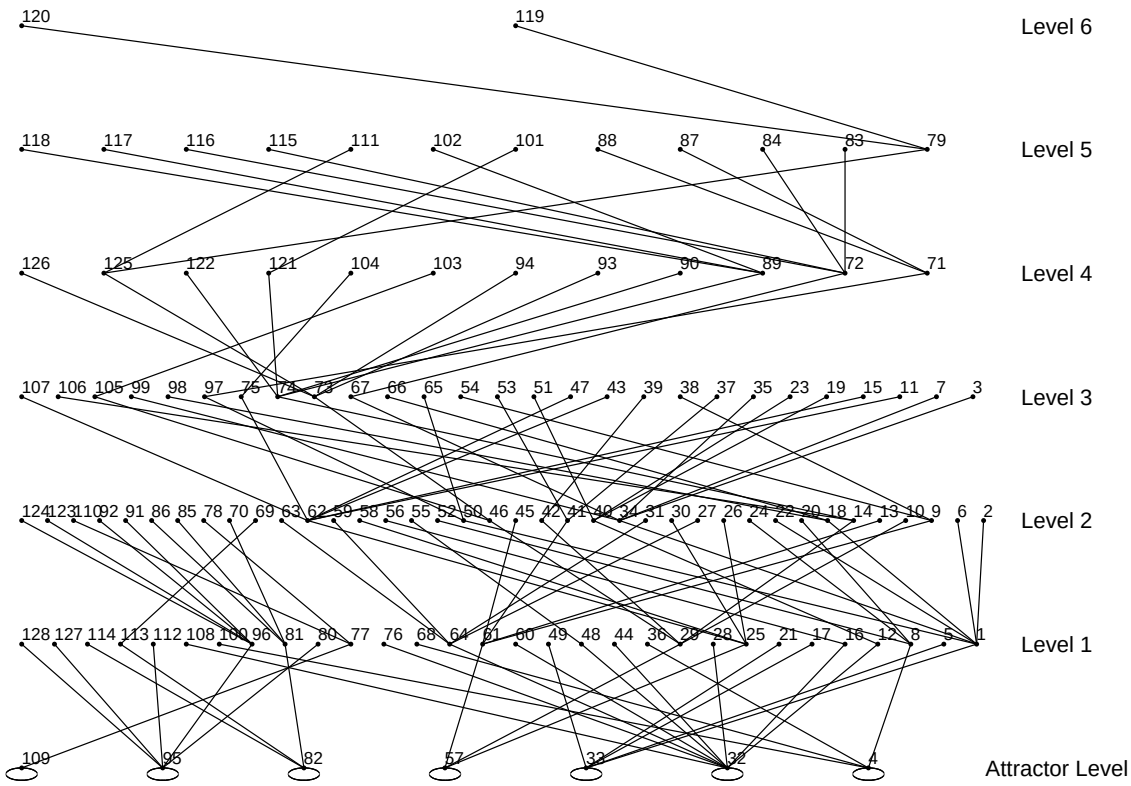

Fig. 22. WNT5A network N_2

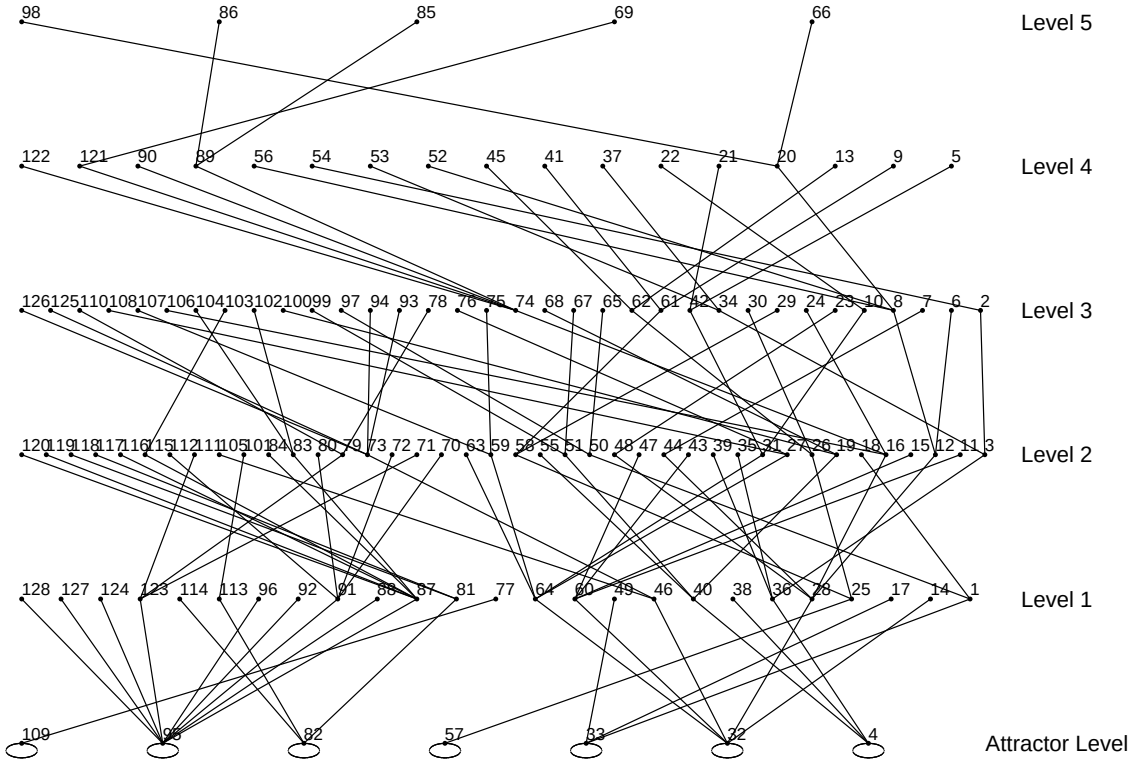

Fig. 23. WNT5A network N_3

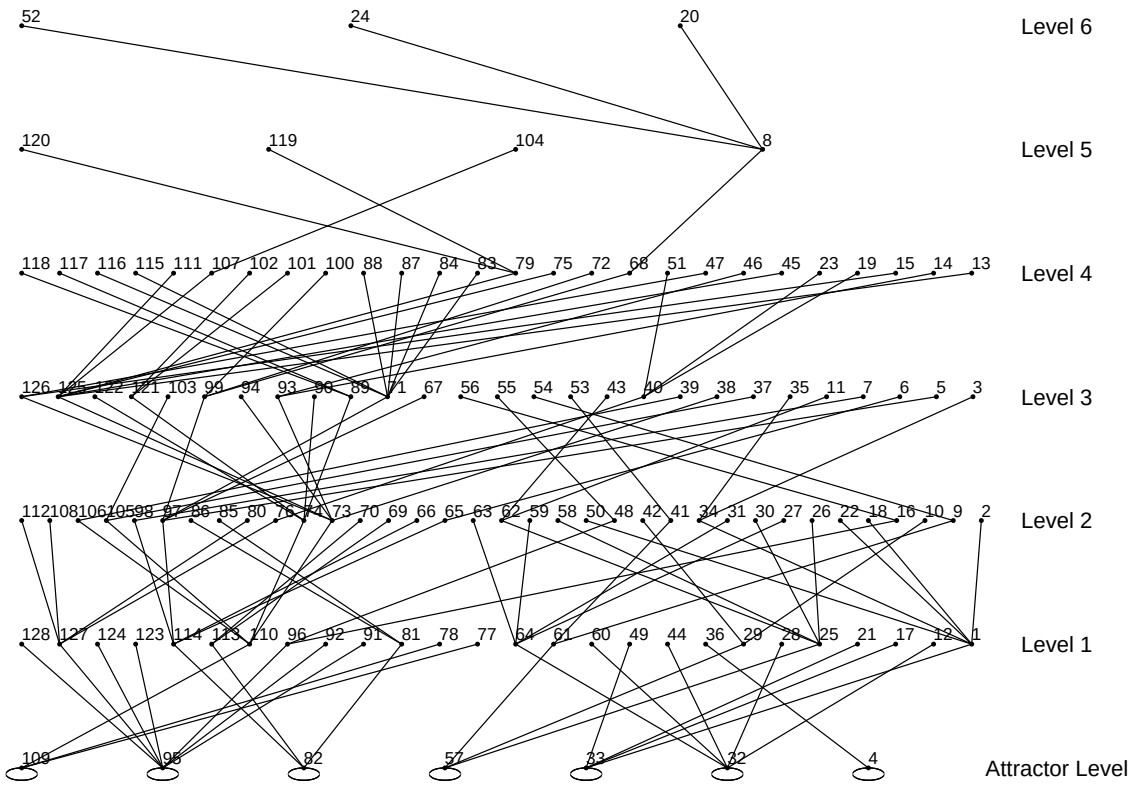

Fig. 24. WNT5A network N_4

We assigned a penalty of 5 to all states in the basin of the undesirable attractors (WNT5A = 1) and 0 to all the other states. We used PIRIN as the control gene. A forcible alteration in the expression level of PIRIN is associated with $v = 1$ while $v = 0$ represents no control. In a reasoning similar to our previous work [7, 8], a terminal penalty of 5 for bad states vs. 0 for good states and a control cost of 1 for intervention vs. 0 for no intervention is our attempt to capture the intuitive notions of the relative costs of ending up in desirable vs. undesirable state and the cost of intervention.

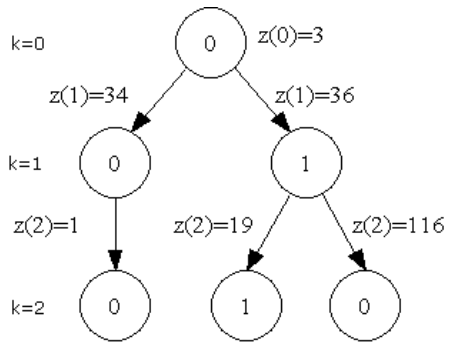

Fig. 25. Policy tree for $M = 3$, initial state $z(0) = 3$ and initial belief vector π_0=[1/4, 1/4, 1/4, 1/4]

A pruned policy tree for $M = 3$ with initial belief vector π_0=[1/4, 1/4, 1/4, 1/4] and initial state $z(0) = 3$ is shown in Figure 25. The expected cost is 0.75 when we control using Pol^{TR}, 1.5 when using Pol^{SW}, and 1.75, 2.5, 1.5 and 1.75 when using Pol^1, Pol^2, Pol^3 and Pol^4, respectively. The expected uncontrolled cost is 2.5. For all horizons M and all initial states $z(0) = i \in \mathcal{S}$ the method of this chapter is superior to those discussed in the earlier chapters. Out of the 128 states in the network, 89 states needed to be controlled in at least one of the 4 networks. In particular for $M = 5$, starting from such states Pol^{TR} was more effective than Pol^{SW} in reducing the cost by 0.1152 on average. In terms of absolute probabilities Pol^{TR} was able

to take the system to a desirable attractor starting from all initial states and all networks with a probability 1.0, except for states $4, 36, 68, 100$ in network N_2, which are uncontrollable from PIRIN. For Pol^{SW}, states $4, 8, 24, 36, 68, 100$ are not taken to a desirable attractor in N_2. In the event of N_2 being the underlying network, starting from states $4, 36, 68, 100$, Pol^{TR} recognizes this and gives up promptly, while Pol^{SW} keeps on applying control, incurring extra costs, without any extra benefit.

Policy trees for initial state $z(0) = 93$, π_0=[1/4, 1/4, 1/4, 1/4], and $M = 2, 3$ and 4 are shown in Figure 26. The expected cost with $M = 2$ is 1.0 which can be further reduced to 0.25 if $M \geq 4$. This is reasonable because the algorithm has more time steps to identify and control the system. For this $M = 4$, the policy computation took 0.28 seconds on a 2.4 GHz, P4 processor system.

For this example no states needed more than $M = 4$ steps to reach the minimum possible value of the expected cost.

More examples appear on the website [27].

G. Conclusion

In conclusion, we have developed a method to optimally control a family of BNs that share a common attractor structure. Such a family arises naturally from the steady state data obtained from gene expression microarrays. The control algorithm is presented as a policy tree depending on an initial belief vector that is updated in an adaptive fashion. At every stage of the evolution, the estimated belief vector is used to appropriately weight the individual networks in the construction of the composite cost function to be minimized.

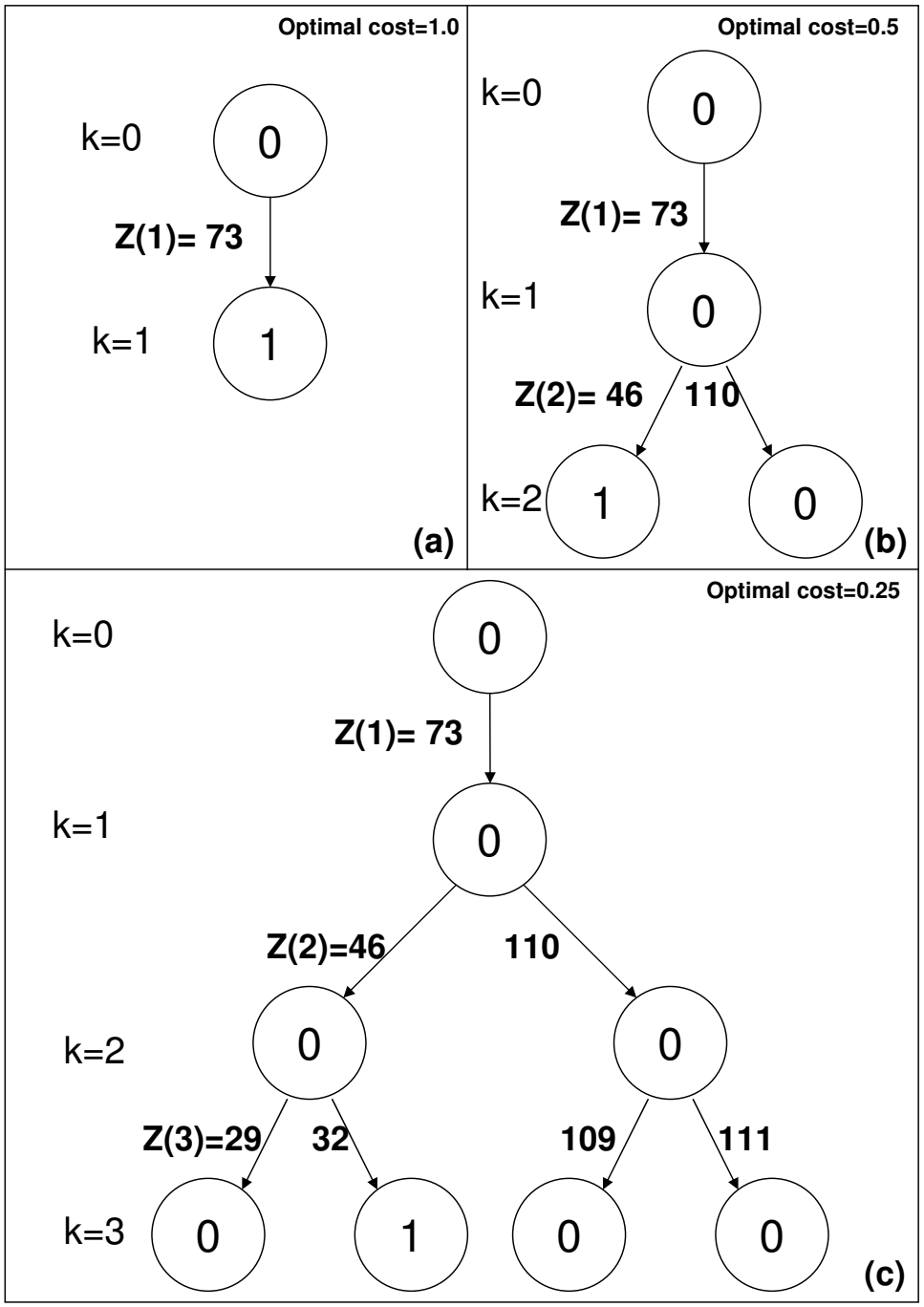

Fig. 26. Policy trees and optimal costs, for initial state $z(0) = 93$, π_0=[1/4, 1/4, 1/4, 1/4], $M = 2$(a), $M = 3$(b) and $M = 4$(c).

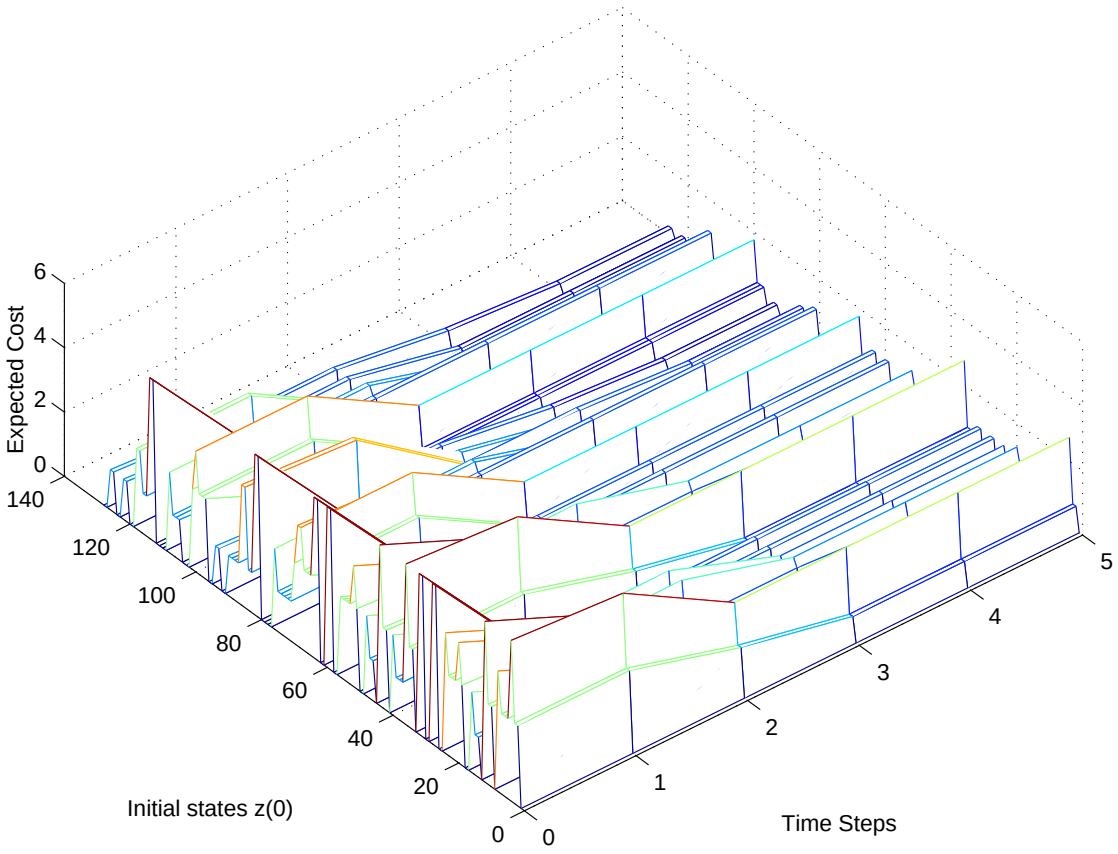

Fig. 27. Expected cost as a function of horizon M and initial state $z(0)$

CHAPTER VII

CONCLUSIONS

In this work, we discussed several approaches that have been recently developed for addressing the issue of intervention in 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 thesis find application in actual clinical practice. We next discuss some of the issues that we are aware of at the current time:

A. Choice of Control Input

In the case of the melanoma cell line study presented in this thesis, 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* [28] may yield some useful insights. Another possibility is to use the concept of gene *influence* introduced in [4], an approach that we have preliminarily explored in chapter V.

B. 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 indica-

tive of the cell type and phenotypic behavior of the cell. Consequently, a reasonable objective of therapeutic intervention could be to explore intervention by altering the attractor landscape in the associated Boolean network.

C. PBN Design Issues

The optimal control results presented in this thesis 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 thesis correspond to the 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.

REFERENCES

[1] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Theoretical Biology*, vol. 22, pp. 437–467, 1969.

[2] S. A. Kauffman and S. Levin, "Towards a general theory of adaptive walks on rugged landscapes," *Theoretical Biology*, vol. 128, pp. 11–45, 1987.

[3] S. A. Kauffman, *The Origins of Order: Self-Organization and Selection in Evolution*, Oxford Univ. Press, New York, 1993.

[4] 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, pp. 261–274, 2002.

[5] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, pp. 1319–1331, 2002.

[6] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Control of stationary behaviour in probabilistic Boolean networks by means of structural intervention," *Biological Systems*, vol. 10, pp. 431–446, 2002.

[7] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," *Machine Learning*, vol. 52, pp. 169–191, 2003.

[8] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks: The imperfect information case," *Bioinformatics*, vol. 20, pp. 924–930, 2004.

[9] R. Pal, A. Datta, M. L. Bittner, and E. R. Dougherty., "Intervention in context-sensitive probabilistic Boolean networks," *Bioinformatics*, vol. 21, pp. 1211–1218, 2005.

[10] R. Pal, A. Datta, and E. R. Dougherty, "Optimal infinite horizon control for probabilistic Boolean networks," *IEEE Transactions on Signal Processing*, vol. 51, 2006, in press.

[11] A. Choudhary, A. Datta, M. L. Bittner, and E. R. Dougherty, "Intervention in family of networks," *Bioinformatics*, vol. 22, pp. 226–232, 2006.

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

[13] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, and V. Sondak, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, pp. 536–540, 2000.

[14] A. T. Weeraratna, Y. Jiang, G. Hostetter, K. Rosenblatt, P. Duray, M. Bittner, and J. M. Trent, "Wnt5a signalling directly affects cell motility and invasion of metastatic melanoma," *Cancer Cell*, vol. 1, pp. 279–288, 2002.

[15] S. Kim, H. Li, E. R. Dougherty, N. Chao, Y. Chen, M. L. Bittner, and E. B. Suh, "Can Markov chain models mimic biological regulation?," *Biological Systems*, vol. 10, pp. 437–458, 2002.

[16] E. R. Dougherty, S. Kim, and Y. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Process.*, vol. 80, pp. 2219–2235, 2000.

[17] 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," *Journal of Biomedical Optics*, vol. 5, pp. 411–424, 2000.

[18] S. Kim, E. 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.

[19] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Athena Scientific, Belmont Massachusetts, 2nd edition, 2001.

[20] R. D. Smallwood and E. J. Sondik, "Optimal control of partially observable Markov processes over a finite horizon," *Operations Research*, vol. 21, pp. 1071–1088, 1973.

[21] A. Choudhary, A. Datta, M. Bittner, and E. R. Dougherty, "Assignment of terminal penalties in controlling genetic regulatory networks," *Proc. American Control Conference*, pp. 417–422, 2005.

[22] A. Choudhary, "Assignment of terminal penalties in controlling genetic regulatory networks, supplementary information and detailed figures," http://ee.tamu.edu/∼cdry/acc05.pdf, 2005.

[23] R. Pal, I. Ivanov, A. Datta, M. L. Bittner, and E. R. Dougherty, "Generating Boolean networks with a prescribed attractor structure," *Bioinformatics*, vol. 21, pp. 4021–4025, 2005.

[24] X. Zhou, X. Wang, R. Pal, I. Ivanov, M. L. Bittner, and E. R. Dougherty, "A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks," *Bioinformaticss*, vol. 20, pp. 2918–2927, 2004.

[25] A. Choudhary, "Proof of concept ", http://ee.tamu.edu/∼cdry/proof.ppt, 2005.

[26] L. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

[27] A. Choudhary, "Additional information for intervention in family of Boolean networks," http://ee.tamu.edu/∼edward/tree, 2005.

[28] R. E. Kalman, "Canonical structure of linear dynamical systems," *Proc. Natl Acad. Sci. USA*, vol. 48, pp. 596–600, 1962.

APPENDIX A

Proof of (4.3)

Now for any $j = 1, 2, \cdots, 2^n$, we have

$$\begin{aligned} p_{k+1}^{j} &= Pr\left\{z_{k+1} = j | I_{k+1}\right\} \\ &= Pr\left\{z_{k+1} = j | I_k, v_k, \theta_{k+1}\right\} \\ &= \frac{Pr\left\{z_{k+1} = j, \theta_{k+1} | I_k, v_k\right\}}{Pr\left\{\theta_{k+1} | I_k, v_k\right\}} \\ &= \frac{Pr\left\{z_{k+1} = j, \theta_{k+1} | I_k, v_k\right\}}{\sum_{j=1}^{2^n} Pr\left\{z_{k+1} = j, \theta_{k+1} | I_k, v_k\right\}} \end{aligned} \tag{A.1}$$

We next evaluate the numerator of the above expression:

$$\begin{aligned} & Pr\left\{z_{k+1} = j, \theta_{k+1} | I_k, v_k\right\} \\ = & \sum_{i=1}^{2^n} Pr\left\{z_{k+1} = j, z_k = i, \theta_{k+1} | I_k, v_k\right\} \\ = & \sum_{i=1}^{2^n} Pr\left\{z_k = i | I_k, v_k\right\} . Pr\left\{z_{k+1} = j, \theta_{k+1} | I_k, v_k, z_k = i\right\} \\ = & \sum_{i=1}^{2^n} Pr\left\{z_k = i | I_k\right\} . Pr\left\{z_{k+1} = j | I_k, v_k, z_k = i\right\} \\ & . Pr\left\{\theta_{k+1} | z_{k+1} = j, I_k, v_k, z_k = i\right\} \\ & \quad \text{(since } z_k \text{ does not depend on } v_k\text{)} \\ = & \sum_{i=1}^{2^n} Pr\left\{z_k = i | I_k\right\} . Pr\left\{z_{k+1} = j | z_k = i, v_k\right\} \\ & . Pr\left\{\theta_{k+1} | z_{k+1} = j, v_k\right\} \\ & \quad \text{(since } z_{k+1} \text{ given } z_k \text{ and } v_k \text{ does not depend on } I_k\text{; and } \theta_{k+1} \text{ given} \\ & \quad z_{k+1} \text{ and } v_k \text{ does not depend on } I_k \text{ or } z_k\text{)} \\ = & \sum_{i=1}^{2^n} p_k^i . a_{ij}(v_k) . r_{j,\theta_{k+1}}^{v_k} \end{aligned} \tag{A.2}$$

Substituting (A.2) into (A.1), we obtain

$$p_{k+1}^{j} = \frac{\sum_{i=1}^{2^n} p_k^i a_{ij}(v_k).r_{j,\theta_{k+1}}^{v_k}}{\sum_{j=1}^{2^n}\sum_{i=1}^{2^n} p_k^i a_{ij}(v_k).r_{j,\theta_{k+1}}^{v_k}},\ j = 1, 2, \cdots, 2^n$$

VITA

Ashish Choudhary received the B.Tech degree in electrical engineering from Indian Institute of Technology Bombay, Mumbai, India in 2001 and the M.S. degree in electrical engineering from Texas A&M University in 2003. He received his Ph.D in electrical engineering from Texas A&M University in 2006. His research areas are genomic signal processing, morphological image processing, statistical learning and control. He can be reached at 9 Zachry Engineering Center, Texas A&M University, College Station, TX 77843-3128, USA.