# AN ENGINEERING APPROACH TOWARDS PERSONALIZED CANCER THERAPY

A Dissertation

by

GOLNAZ VAHEDI

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 2009

Major Subject: Electrical Engineering

# AN ENGINEERING APPROACH TOWARDS PERSONALIZED CANCER THERAPY

A Dissertation

by

GOLNAZ VAHEDI

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:

| | |
|---|---|
| Co-Chairs of Committee, | Edward Dougherty |
| | Jean-Francois Chamberland-Tremblay |
| Committee Members, | Aniruddha Datta |
| | Shankar Bhattacharyya |
| | Natarajan Sivakumar |
| Head of Department, | Costas Georghiades |

August 2009

Major Subject: Electrical Engineering

# ABSTRACT

An Engineering Approach Towards Personalized Cancer Therapy. (August 2009)

Golnaz Vahedi, B.S., Sharif University of Technology;

M.S., University of Alberta

Co–Chairs of Advisory Committee: Dr. Edward R. Dougherty
Dr. Jean-Francois Chamberland-Tremblay

Cells behave as complex systems with regulatory processes that make use of many elements such as switches based on thresholds, memory, feedback, error-checking, and other components commonly encountered in electrical engineering. It is therefore not surprising that these complex systems are amenable to study by engineering methods. A great deal of effort has been spent on observing how cells store, modify, and use information. Still, an understanding of how one uses this knowledge to exert control over cells within a living organism is unavailable. Our prime objective is "Personalized Cancer Therapy" which is based on characterizing the treatment for every individual cancer patient. Knowing how one can systematically alter the behavior of an abnormal cancerous cell will lead towards personalized cancer therapy. Towards this objective, it is required to construct a model for the regulation of the cell and utilize this model to devise effective treatment strategies. The proposed treatments will have to be validated experimentally, but selecting good treatment candidates is a monumental task by itself. It is also a process where an analytic approach to systems biology can provide significant breakthrough. In this dissertation, theoretical frameworks towards effective treatment strategies in the context of probabilistic Boolean networks, a class of gene regulatory networks, are addressed. These proposed analytical tools provide insight into the design of effective therapeutic interventions.

To my parents *Simin and Mehdi*, for their endless patience.

# ACKNOWLEDGMENTS

It is difficult to overstate my gratitude to my adviser, Dr. Edward R. Dougherty, whose constant support created an ideal environment in which to pursue my intellectual development, giving me the unique freedom to pursue unorthodox ideas. He is a coauthor in all my published work surveyed in this manuscript and his technical maturity and knowledge of the subject has been a great source of inspiration at every step.

I am also deeply indebted to my co-adviser, Dr. Jean-Francois Chamberland, without whose sincere care and guidance the very possibility of finishing this work would be a matter of great doubt. So much of one's experience in graduate school is colored by the personality of their advisor, and his relaxed and open approach to research is the source of a warm and friendly atmosphere. Moreover, his dedication to teaching and to his students is without compare. I must also thank my other coauthor and committee members, Dr. Aniruddha Datta and Dr. Shankar Bhattacharyya, for their invaluable contributions to this work. The administrative support provided by Tammy Carda and Jeanie Marshall has also been very valuable. Their efforts have allowed me to all but ignore the ceaseless paperwork that keeps the system running.

I give special thanks to my parents, Simin Mozaffari and Mehdi Vahedi, for their love and support through all of this. They raised me, supported me, taught me, and loved me. To them, I dedicate this dissertation.

At the end, I wish to express my deepest and most sincere love and gratitude to Babak. Without him, life in College Station would unquestionably be utter misery.

TABLE OF CONTENTS

CHAPTER Page

# LIST OF TABLES

TABLE Page

# LIST OF FIGURES

FIGURE Page

FIGURE Page

# CHAPTER I

## INTRODUCTION

Cells form complex systems and can be approached by engineering methods. Their behavior can be abstracted using many elements such as switches based on thresholds, memory, feedback, and other elements common in engineering systems. Nevertheless, studying cells can be challenging for an engineer. The primary problem is that cells are much more complex than man-made systems in terms of the numbers of inputs involved in any given decision. This situation is further exacerbated by the paucity of information that is available on the configuration of the regulatory networks and the dynamics of reconfiguration of these same networks as a cell responds. Although high-throughput technologies such as microarrays provide powerful tools to characterize genome or proteome, each only supplies a snap-shot of the state of cells. With these technologies, we are beginning to characterize information and its means of transmission within a biological system. Still, what we are still lacking is how to use this knowledge to exert control over biological systems. Our prime objective of our research is “Personalized Cancer Therapy” which is based on characterizing the treatment for every individual cancer patient. Knowing how one can systematically alter the behavior of an abnormal cancerous cell will bring us closer to personalized cancer therapy. Towards this objective, we are required to construct a model for the regulation of the cell and utilize this model to devise effective treatment strategies. The proposed treatments will have to be validated experimentally, but selecting good treatment candidates is a monumental task by itself. It is also a process where an analytic approach to systems biology can provide significant breakthrough.

---

This dissertation follows the style of *IEEE Transactions on Biomedical Engineering.*

In this dissertation, we address theoretical frameworks towards effective treatment strategies in the context of a given model of gene regulatory networks.

Most cellular processes involve many different molecules. The metabolism of a cell consists of many interlinked reactions. Products of one reaction will influence the next, thus forming the metabolic network [1]. Similarly, signaling molecules are interlinked and cross-talk between the different signalling cascades forms the signaling network. The same is true for regulatory relationships between genes and their products. All these networks are closely related, e.g. the regulatory network is influenced by extracellular signals. Our main interest is in transcription regulation networks and we will refer to them as "gene regulatory networks".

There have been numerous attempts to model the dynamical behavior of gene regulatory networks, ranging from deterministic to stochastic, using either discrete-time or continuous-time descriptions of gene interactions. Paradigms that have been considered in this context include Bayesian networks [2], Boolean networks [3], and, recently, probabilistic Boolean networks [4].

A deterministic model of a gene regulatory network can involve a number of different mechanisms that capture the collective behavior of the elements constituting the network. What deterministic models have in common is that there is no inherent notion of randomness or stochasticity in the model once it is specified [4]. A model system that has received much attention, not only from the biology community, but also in physics, is the Boolean network model, originally introduced by Kauffman [3]. In this model, gene expression is quantized to only two levels: ON and OFF. The expression level (state) of each gene is functionally related to the expression states of some other genes, using logical rules [5]. In a Boolean network, each (target) gene is "predicted" by several other genes by means of a Boolean function (predictor). Thus, after having inferred such a function from gene expression data, it could be concluded that if we observe the values of the predictive genes,

we know, with full certainty, the value of the target gene.

Conceptually, such an inherent determinism seems problematic as it assumes an environment with no uncertainty. However, the data used for the inference exhibits uncertainty on several levels. First, there is biological uncertainty: gene expression is inherently stochastic, not in the sense that it is totally random, but that it has a stochastic nature on account of intrinsic biological variability. Second, there is experimental noise due to the complex measurement process, ranging from hybridization conditions to microarray image processing techniques. Third, there may be interacting latent variables, such as proteins, various environmental conditions, or other genes that we simply do not measure, that further add to the variability of the measurements. Thus, we are in a position of having to infer a (deterministic) predictor under uncertainty. Probabilistic Boolean networks (PBNs) have been introduced to address such uncertainty. A number of additional justifications for introducing PBNs are contained in [5]. In this dissertation, we model the gene regulatory network dynamics as a probabilistic Boolean networks.

PBNs allow the incorporation of uncertainty into the inter-gene relationships [6]. This class of models offers a more flexible and powerful mathematical abstraction. PBN is capable of incorporating the effect of latent variables, not directly captured in a Boolean network. The dynamics of a PBN can be represented via a discrete-time Markov chain. A logical state of the corresponding Markov chain is a gene activity profile.

Our ultimate objective in modeling genetic regulatory networks is the identification of potential targets for therapeutic intervention [6]. For instance, in cancer, one can consider correlation between metastasis and the abundances of mRNA for certain genes. In this respect, the abundance of mRNA for the gene WNT5A has been found to be highly discriminating between cells with properties typically associated with high versus low metastatic competence [7]. Appropriate alteration in the expression of WNT5A can be perceived therapeutically, and it can therefore be used to search for an optimal intervention strategy [8].

In [9] and [4], several methods to design therapeutic interventions are discussed in the context of probabilistic Boolean networks. Some of these methods are intended to reduce the likelihood of the gene-expression profiles associated with aberrant cellular functions via manipulation of a control gene. In a nutshell, whenever changing the expression level of a control gene is perceived as a therapeutic option, these system-based therapies search for the most effective sequence of such changes to beneficially alter cell dynamics. The resulting intervention strategy specifies the appropriate expression of the control gene in order to reduce the likelihood of pathological cellular functions.

Major efforts have initially focused on manipulating external (control) variables to desirably affect dynamical evolution over a finite time horizon [10]. These short-term policies have been shown to change the dynamical performance of regulatory networks over a small number of stages; however, they are not necessarily effective in changing long-run network behavior. To address this issue, stochastic control has been employed via dynamic programming algorithms to find stationary control policies that affect the steady-state distributions of PBNs [11].

In this dissertation, we first address one practical issue in the inference of PBNs from biological data [4]. A significant effort has been put forth to infer PBNs. The inference problem depends on the kinds of data available. Data are often assumed to come from the steady-state distribution of the underlying biological network. This is typically the case for cancer patient data. We show how and why using the coefficient of determination (CoD) in the inference of PBNs can lead to artifacts in the structure of the network. We also propose an inference algorithm to avoid such artifacts.

Formulating the problem of intervention in a regulatory network as a classical infinite-horizon decision making process introduces an elegant analytical framework that may be instrumental to enhance our understanding of treatment discovery. Despite its conceptual benefits, the classical intervention fails to address many practical and technical issues. In

the past few years, the classical framework has been extended in several directions to improve system-based intervention schemes. To this end, we consider three control theoretic problems in the context of PBNs. These proposed analytical tools provide insight into the design of effective therapeutic interventions. These methods strive to address some of the practical concerns that are brought up by medical practitioners. In the following, we explain them in more details.

## A. Boolean Networks and Bidirectional Gene Relationships

The coefficient of determination (CoD) has been used to infer Boolean networks from steady-state biological data, in particular, to estimate the constituent Boolean networks for a probabilistic Boolean network. The advantage of the CoD method over design methods that emphasize graph topology or attractor structure is that the CoD produces a network based on strong predictive relationships between target genes and their predictor (parent) genes. The disadvantage is that spurious attractor cycles appear in the inferred network, so that there is poor inference relative to the attractor structure, that is, relative to the steady-state behavior of the network. An attractor is a set of states to which a Boolean network evolves after a long enough time. Given steady-state data, there should not be a significant amount of steady-state probability mass in the inferred network lying outside the mass of the data distribution; however, the existence of spurious attractor cycles creates a significant amount of steady-state probability mass not accounted for by the data.

Using steady-state data hampers design because the absence of temporal data causes the CoD method to suffer from a lack of directionality with regard to prediction. This may result in spurious bidirectional relationships among genes in which two genes are among the predictors for each other, when actually only one of them should be a predictor of the other, thereby creating a spurious attractor cycle. Chapter III characterizes the manner in

which bidirectional relationships affect the attractor structure of a Boolean network. Given this characterization, we propose a constrained CoD inference algorithm that outperforms unconstrained CoD inference in avoiding the creation of spurious non-singleton attractor. Algorithm performances are compared using a melanoma-based probabilistic Boolean network [12].

## B. Timing in Probabilistic Boolean Networks

Implementation of an intervention policy derived for probabilistic Boolean networks requires nearly continuous observation of the underlying biological system since precise application requires the observation of all transitions. In medical applications, as in many engineering problems, the process is sampled at discrete time intervals and a decision to intervene or not must be made at each sample point.

In this work, we construct a framework for gene interactions such that the model class: (i) incorporates rule-based dependencies among genes, (ii) allows the systematic study of global network dynamics, (iii) is able to cope with uncertainty, (iv) accounts for the sampling rate of temporal profiles, (v) remains robust to large estimation errors due to small samples. To this end, in Chapter IV, we extend the current definition of PBN and propose a discrete-time discrete-space model called sampling-rate-dependent PBN (SRD-PBN) [13].

## C. Optimal Cyclic Control Policy

We are able to exploit the biochemical differences between bacteria and human cells so as to achieve toxic drug concentrations in the former while sparing the latter. This selectivity largely contributes to the success in treating bacterial infections. Unfortunately, such high selectivity is at present elusive in the treatment of human cancers. Hence, great ef-

forts are required to determine dose schedules that maximize the benefit-to-toxicity ratio in cancer therapy [14]. Dose intensity is a measure of treatment delivery that looks at the amount of drug delivered per unit of time. To mitigate the detrimental side effects of a treatment in general, we should account for dose intensity in a system-based intervention method. Therapeutic intervention should avoid undesirable gene-expression profiles while accounting for the quantity or frequency of applied drugs. A higher drug dose intensity can be delivered by increasing the dose per cycle (dose escalation) or by reducing the interval between cycles (dose density).

To reduce the side-effects, certain types of cancer therapies, such as chemotherapy, are given in cycles with each treatment being followed by a recovery period. During the recovery period, the side effects tend to gradually subside. In Chapter V, we show how an optimal cyclic intervention strategy can be devised for a PBN. The effectiveness of optimal cyclic therapies is demonstrated through numerical studies for random networks. Furthermore, an optimal cyclic policy is derived to control the behavior of a regulatory model of the mammalian cell-cycle network [15].

### D. Mean-First-Passage-Time Control Policy

In general, dynamical programming algorithms can be problematic owing to their high computational complexity. Two additional computationally burdensome issues that arise in cancer therapy are the potential for controlling the network and identifying the best gene for intervention. Chapter VI proposes an algorithm based on mean first-passage time that assigns a stationary control policy for each gene candidate. It serves as an approximation to an optimal control policy and, because of its reduced computational complexity, can be used to predict the best control gene. Once the best control gene is identified, one can derive an optimal policy or simply utilize the approximate policy for this gene when the

network size precludes a direct application of dynamic programming algorithms. A salient point is that the proposed algorithm can be model-free. It can be directly designed from time-course data without having to infer the transition probability matrix of the network [16].

To set the stage, in Chapter II, we first introduce the background of this research in more details. Chapter III considers the bidirectional relationships in Boolean networks. In Chapter IV, we introduce how sampling-rate can be incorporated in probabilistic Boolean networks. We develop a novel framework to model cyclic cancer treatments such as chemotherapy in Chapter V. We propose a heuristic control design method based on mean-first-passage-times in Markov chains in Chapter VI.

# CHAPTER II

# BACKGROUND

## A. Boolean Networks

A Boolean network (BN) $G(V, F)$, [17], is defined by a sequence $V = \{x_i\}_{i=1}^n$ of $n$ nodes and a set of Boolean functions $\mathbf{f} = \{f_1, ..., f_n\}$where $x_i \in \{0, \cdots, d-1\}$ and $d$ denotes the quantization level. In gene regulatory modeling, $x_i$ represents the expression level of gene $i$, which can be either active (1) or inactive (0). As is commonly done, we will mix terminology by referring to the nodes as genes. The set of Boolean functions $\mathbf{f}$ represents the regulatory rules between genes. At time step $t+1$, the expression of gene $x_i$, called the target gene, is predicted by the expression of a set, $W_i$, of genes at time step $t$, based on the regulatory function $f_i$. The sequence of genes $W_i = \{x_{i_1}, ..., x_{i_{k_i}}\}$ is called the *predictor set* of $x_i$. The function $f_i$ is called the *predictor function* of $x_i$. We assume that there are no nonessential genes in a predictor set, meaning that the predictor function requires the full set as input. The cardinality of $W_i$, $|W_i|$, is called the *connectivity* of $x_i$ and the maximum connectivity in the network is called the connectivity of the network.

A state of the BN at time $t$ is a vector $(x_1(t), ..., x_n(t))$ of gene values which also referred to as *gene activity profile*. The possible states of the BN form its *state space*. Given an initial state, the network will eventually enter a set of states through which it will repeatedly cycle forever. Each such set is called an *attractor cycle*, and a *singleton attractor* is an attractor cycle of length $1$. The attractor cycles are mutually disjoint. The set of all states that transition into an attractor cycle is called the *basin* of that cycle. The family of basins partitions the state space.

## B. Probabilistic Boolean Networks

Probabilistic Boolean network (PBN) [6] consists of a sequence $V = \{x_i\}_{i=1}^{n}$ of $n$ nodes with $x_i \in \{0, \cdots, d-1\}$, together with a sequence $\{\mathbf{f}_c\}_{c=1}^{k}$ of vector-valued *network functions*. In the framework of gene regulation, each element $x_i$ represents the expression level of a gene. It is common to mix the terminology by referring to $x_i$ as the $i$th gene. The state vector $\mathbf{x}(t) = (x_1(t), \ldots, x_n(t))$ is called the *gene-activity profile* (GAP) at time $t$. Each network function $\mathbf{f}_c = (f_{c1}, \ldots, f_{cn})$ determines a constituent network (*context*) of the regulatory network. The function $f_{ci} : \{0, \ldots, d-1\}^n \rightarrow \{0, \ldots, d-1\}$ is the predictor of gene $i$, whenever network $c$ is selected. At each updating epoch a decision is made whether to switch the constituent network. This decision is based on a binary random variable $\xi$ with $P(\xi = 1) = q$. If $\xi = 0$, then the network is not switched, the model behaves like a fixed network and the values of all genes are synchronously updated according to the current constituent network. If $\xi = 1$, then a constituent network is randomly selected from among all constituent networks, including the current one, according to the selection probability distribution $\{p_c\}_{c=1}^{k}$ and, after selecting $\mathbf{f}_c$, the values of all genes are updated accordingly. If $q = 1$, so that a switch is permitted at every time point, the network is said to be *instantaneously random*; if $q < 1$, then the PBN will remain in a constituent network so long as $\xi$ remains equal to $0$, and the PBN is said to be *context-sensitive*.

Two quantization levels have thus far been used in practice. If $d = 2$ (binary), then the constituent networks are Boolean networks with $0$ or $1$ meaning OFF or ON, respectively. The case $d = 3$ (ternary) arises when we consider a gene to be $0$ (down-regulated), $2$ (up-regulated), and $1$ (invariant). This latter situation commonly occurs with cDNA microarrays, where a ratio is taken between the expression values on the test channel (red) and the base channel (green). In this work, we will develop the methodology for $d = 2$, so that gene values are either $0$ or $1$; however, the methodology is applicable to any finite

number of levels. For binary PBNs, there is a natural bijection between the GAP $\mathbf{x}(t)$ and its integer representation, $x(t)$, which takes values in $\mathcal{W} = \{0, 1, \ldots, 2^n{-}1\}$. We consider a PBN with perturbation, meaning that there is a binary random vector $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_n)$, independent of $\xi$, such that $P(\gamma_i = 1) = p$, and $\gamma_1, \gamma_2, \ldots, \gamma_n$ are independent. If $\gamma = \mathbf{0}$ the network transitions according to the network function, and if $\gamma \neq \mathbf{0}$ the value $x_i$ flips if and and only if $\gamma_i = 1$.

The dynamic behavior of an instantaneous PBN can be modeled by a Markov chain with state space of $\mathcal{W}$. Similarly, the dynamic behavior of a context-sensitive PBN can be modeled by a Markov chain whose states consist of (context, GAP) ordered pairs taking values in

$$\big\{(c, x) : c \in \{1, \ldots, k\}\ ,\ x \in \mathcal{W}\big\}.$$

In either frameworks, let $\mathbf{P}$ denote the transition probability matrix of the corresponding Markov chain where the state space is denoted as $\mathcal{S}$. The evolution of the network can be modeled by a stationary discrete-time equation

$$z(t+1) = f(z(t), w(t)) \qquad \text{for } t = 0, 1, \ldots,$$

where state $z(t) \in \mathcal{S}$. The disturbance $w(t)$ is the manifestation of uncertainties, due to either network switching or a change in gene-activity profile resulting from a random gene perturbation. Gene perturbation insures that all states in the Markov chain communicate with one another. Hence, the finite-state Markov chain is ergodic and has a unique steady-state distribution.

## C. Optimal Control in Probabilistic Boolean Networks

In the following, we describe how to devise an optimal control policy for a PBN. Let $\mathbf{P} = (p_{ij}; i, j \in \mathcal{S})$ denote the transition probability matrix of the Markov chain corresponding to the context-sensitive or instantaneous PBN. In the presence of external control, we suppose that there exists a binary control input, $u(t) \in \mathcal{C} = \{0, 1\}$. A control $u(t)$, which can take values $0$ or $1$ at each updating epoch $t$, specifies the action on the control gene. Treatment alters the status of the control gene, which can be selected among all genes in the network. If the control at updating epoch $t$ is on, $u(t) = 1$, then the state of the control gene is toggled; if $u(t) = 0$, then the state of the control gene remains unchanged. In the presence of external control, the system evolution is represented by a stationary discrete-time equation

$$z(t+1) = f(z(t), u(t), w(t)) \qquad \text{for } t = 0, 1, \ldots$$

where state $z(t)$ is an element of the state-space $\mathcal{S}$; and $w(t)$ is the manifestation of uncertainties in the model. The probability of transitioning from state $i$ to state $j$ under control $u$ is denoted by $p_{ij}(u)$, where $i, j \in \mathcal{S}$.

The problem of optimal intervention for a PBN is formulated as an optimal stochastic control problem. A cost-per-stage, $g(i, u, j)$, is associated to each intervention in the system. In general, a cost-per-stage may depend on the origin state $i$, the successor state $j$, and the control input $u$. We assume that the cost-per-stage is stationary and bounded for all $i$, $j$ in $\mathcal{S}$, and $u$ in $\mathcal{C} = \{0, 1\}$. We define the expected immediate cost in state $i$, when control $u$ is selected, by

$$\overline{g}(i, u) = \sum_{j \in \mathcal{S}} p_{ij}(u)\, g(i, u, j).$$

We consider the discounted formulation of the expected total cost. The discounting factor, $\alpha \in (0, 1)$, ensures convergence of the expected total cost over the long-run [18].

In the case of cancer therapy, the discounting factor emphasizes that obtaining treatment at an earlier stage is favored over later stages. The expected total discounted cost, given a policy $\pi$ and an initial state $i$, is denoted by

$$J_\pi(i) = \lim_{N\to\infty} E\left\{ \sum_{t=0}^{N-1} \alpha^t \, r(z(t), \mu_t(z(t)), z(t+1)) \,\middle|\, z(0) = i \right\}, \tag{2.1}$$

where $z(t), i \in \mathcal{S}$. A policy $\pi = \{\mu_{\mathbf{0}}, \mu_{\mathbf{1}}, \ldots\}$ is a sequence of decision rules $\mu_t : \mathcal{S} \to \mathcal{C}$, for each time step $t$. The vector $\mathbf{J}_\pi$ of the expected total costs is called the value function. In a stochastic control problem, we seek an intervention strategy $\pi^*$ among all the admissible intervention strategies $\Pi$ that minimizes the value function for each state $i$, i.e.,

$$\pi^*(i) = \arg \min_{\pi \in \Pi_g} J_\pi(i), \qquad \forall\, i \in \mathcal{S}. \tag{2.2}$$

For a finite time horizon, the dynamic programming algorithm describes how the optimal cost $J_{k+1}$ propagates backward in time to the optimal cost $J_k$

$$J_k(i) = \min_{u \in \mathcal{C}} \left[ \overline{g}(i,u) + \lambda \sum_{j=0}^{N-1} p_{i,j}(u) J_{k+1}(j) \right]. \tag{2.3}$$

The above equation motivates the introduction of the mapping $T : \mathcal{S} \mapsto \Re$ defined by

$$TJ(i) = \min_{u \in \mathcal{C}} \left[ \overline{g}(i,u) + \lambda \sum_{j=0}^{N-1} p_{i,j}(u) J(j) \right], \quad \forall i \in \mathcal{S}, \tag{2.4}$$

for any value function $J : \mathcal{S} \mapsto \Re$. Given the mapping of (2.4), the following propositions summarize how one can devise an optimal one-transition policy. Proofs of these statements can be found in [18].

**Proposition 1** *(Convergence of the discounted cost algorithm): For any $x \in \mathcal{S}$ and any bounded function $J : \mathcal{S} \mapsto \Re$, the optimal cost function satisfies*

$$J^*(x) = \lim_{M \longrightarrow \infty} \left(T^M J\right)(x), \qquad \forall\, x \in \mathcal{S}.$$

**Proposition 2** *(Bellman's optimality equation): The optimal cost function $J^*$ satisfies*

$$J^* = TJ^*. \tag{2.5}$$

*Furthermore, $J^*$ is the unique solution of this equation within the class of bounded functions.*

**Proposition 3** *(Necessary and sufficient condition for optimality): A stationary policy $\mu$ is optimal if and only if it attains the minimum in Bellman's optimality equation of* (2.5).

The three aforementioned propositions provide the basis for a method for determining an optimal one-transition policy. Proposition 2 asserts that the optimal cost function satisfies Bellman's optimality equation while Proposition 1 states that the optimal cost function can be iteratively determined by running the recursion equation

$$J_{k+1} = TJ_k, \quad k = 0, 1, 2, \ldots \tag{2.6}$$

for any bounded initial cost function $J_0 : \mathcal{S} \mapsto \Re$. Since this iteration is guaranteed to converge to $J^*$, one can continue the iteration until some stopping criterion is reached. By Proposition 3, the resulting optimal policy is also stationary. The procedure described in (2.6) is referred to as the value iteration algorithm since, at every stage, we are iterating on the value function. The optimal one-transition policy is obtained as the argument of the minimization step once the iterative procedure has converged.

## D. Continuous-time Markov Chain

Consider a continuous-time discrete-space stochastic process $\{Z(t), t \geq 0\}$ taking on values in the set of nonnegative integers $\mathcal{S}$. In analogy with a discrete-time Markov chain, we say that the process $\{Z(t), t \geq 0\}$ is a continuous-time Markov chain if $\forall s, t \geq 0$, and nonnegative integers $i, j, z(\nu) \in \mathcal{S}$, $0 \leq \nu < s$,

$$\Pr\{Z(t+s) = j | Z(s) = i, Z(\nu) = z(\nu), 0 \leq \nu < s\} =$$
$$\Pr\{Z(t+s) = j | Z(s) = i\}.$$

In other words, a continuous-time Markov chain is a stochastic process with the Markovian property. This means that the conditional distribution of the future state at time $t+s$, given the present state at time $s$ and all the preceding states, depends only on the present state and is conditionally independent of the states prior to the current state. The past given the present does not provide more information about the future.

If we let $\tau_i$ denote the amount of time that the process stays in state $i$ before making a transition into a different state, then the Markov property implies

$$\Pr\{\tau_i > s + t | \tau_i > s\} \quad = \quad \Pr\{\tau_i > t\}, \ \forall \ s, \ t \geq 0.$$

The random variable $\tau_i$ is memoryless and must therefore be exponentially distributed. In general, a continuous-time Markov chain is defined by a **Q**-matrix. A **Q**-matrix on $\mathcal{S}$ is a matrix $Q = (q_{ij}; i, j \in \mathcal{S})$ satisfying the following conditions [19]:

$$\begin{aligned} &(i) \ \ 0 \leq -q_{ii} < \infty, \ \forall i; \\ &(ii) \ \ q_{ij} \geq 0, \ \forall \ i \neq j; \\ &(iii) \ \ \textstyle\sum_{j \in I} q_{ij} = 0, \ \forall i. \end{aligned} \tag{2.7}$$

In the above, $q_{ij}$ is the rate of transitioning from $i$ to $j$ and $q_i = \sum_{i \neq j} q_{ij}$ is the rate of leaving state $i$. It is known that a matrix Q is a **Q**-matrix on $\mathcal{S}$ if and only if $\mathbf{P}(t) = e^{Qt}$ is a stochastic matrix, $\forall t \geq 0$ [19]. In particular, the transition probability from $i$ to $j$ after $t$ unit of time, the $(i, j)$ element of $\mathbf{P}(t)$, is given by

$$\Pr(X_t = j | X_0 = i) = p_{ij}^t = \left[e^{Qt}\right]_{ij}.$$

## E. Influence in Probabilistic Boolean Networks

Influence is a method for quantifying the relative impact of genes on other genes within the context of PBNs [5]. The influence $I_j(f)$ of gene $x_j$ on the function $f$, with respect to the probability distribution $D(x)$, $x \in \{0,1\}^n$, is defined as

$$I_j(f) = E_D\left[\frac{\partial f(x)}{\partial x_j}\right], \tag{2.8}$$

where $E_D[\cdot]$ is the expectation operator with respect to the distribution $D$, $\frac{\partial f(x)}{\partial x_j} = f(x^{(j,0)}) \oplus f(x^{(j,1)})$ is the partial derivative of the Boolean function $f$, the symbol $\oplus$ is addition modulo 2 (exclusive OR), and $x^{(j,k)} = (x_1, ..., x_{j-1}, k, x_{j+1}, ..., x_n)$ for $k \in \{0,1\}$. In other words, (2.8) gives the influence as the probability (under the distribution $D(x)$) that a toggle of the $j$th variable changes the value of the function. In the context of PBNs, the influence of gene $x_k$ on gene $x_i$ is given by $I_k(x_i) = \sum_{j=1}^{l(i)} I_k(f_j^{(i)}) \cdot p_j^{(i)}$ where $\{p_j^{(i)}\}_{j=1}^{l(i)}$ are the selection probabilities of the predictor functions of gene $i$ and $l(i)$ represents the number of predictor functions of gene $i$ [5]. To quantify the long-run influence, $D(x)$ is the stationary distribution of the PBN.

## F. Biological Data

In the theoretical frameworks we developed, we consider two gene regulatory networks that have been derived from biological data. In the following, these two networks are introduced in more details.

### 1. Melanoma Gene Regulatory Network

The steady-state data was collected in a profiling study of metastatic melanoma in which the abundance of messenger RNA for the gene WNT5A was found to be highly discriminating between cells with properties typically associated with high metastatic competence

versus those with low metastatic competence [7]. These findings were validated and expanded in a second study, in which 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 [20]. A further finding of interest in this study was that an intervention that blocked the WNT5A protein from activating its receptor, the use of an antibody that binds the WNT5A protein, can substantially reduce WNT5A's ability to induce a metastatic phenotype. This suggests control based on intervention that alters the contribution of the WNT5A gene to biological regulation. Disruption of this influence can potentially reduce the chance of a melanoma metastasizing, a desirable outcome. Ten genes, including the WNT5A gene, were selected in [21] based on the predictive relationships among 587 genes: WNT5A, pirin, S100P, RET1, MMP3, PHOC, MART1, HADHB, Synuclein, and STC3. We apply the design procedure proposed in [22] to generate a PBN possessing four constituent BNs. The method of [22] generates BNs with given attractor structures and the overall PBN is designed so that the data points, which are assumed to come from the steady-state distribution of the network, are attractors in the designed PBN. This approach is reasonable because our interest is in controlling the long-run behavior of the network. The control objective for this 10-gene network is to down-regulate the WNT5A gene, because WNT5A ceasing to be down-regulated is strongly predictive of the onset of metastasis. A number of other control studies based on the same data have aimed to down-regulate the WNT5A gene. This model has been used because the relation of WNT5A to metastasis is well established and the binary nature of the up or down regulation suits a binary model. A state is desirable if WNT5A = 0 and undesirable if WNT5A = 1. In this example, the use of the state WNT5A has resulted from biological knowledge relating the state of WNT5A to metastasis in melanoma tumors.

## 2. Mammalian Cell-Cycle Network

In this section, we construct a PBN that is a probabilistic version of the Boolean model for the mammalian cell cycle regulation proposed in [23]. This PBN postulates the mammalian cell cycle with a mutated phenotype.

During the late 1970s and early 1980s, yeast geneticists identified the cell-cycle genes encoding for new classes of molecules, including the cyclins (so-called because of their cyclic pattern of activation) and their cyclin dependent kinases (cdk) partners [23]. Our model is rooted in the work of Faure et al., who have recently derived and analyzed the Boolean functions of the mammalian cell cycle [23]. The authors have been able to quantitatively reproduce the main known features of the wild-type biological system, as well as the consequences of several types of mutations.

Mammalian cell division is tightly controlled. In a growing mammal, the cell division should coordinate with the overall growth of the organism. This coordination is controlled via extra-cellular signals. These signals indicate whether a cell should divide or remain in a resting state. The positive signals, or growth factors, instigate the activation of Cyclin D (CycD) in the cell.

The key genes in this model are CycD, retinoblastoma (Rb), and p27. Rb is a tumor-suppressor gene. This gene is expressed in the absence of the cyclins, which inhibits the Rb by phosphorylation. Whenever p27 is present, Rb can be expressed even in the presence of CycE or CycA. Gene p27 is active in the absence of the cyclins. Whenever p27 is present, it blocks the action of CycE or CycA. Hence, it stops the cell cycle. Table I summarizes the Boolean functions of the wild-type cell cycle network.

The preceding explanation represents the wild-type cell-cycle model. Following one of the proposed mutations in [23], we assume p27 is mutated and its logical rule is always zero (OFF). In this cancerous scenario, p27 can never be activated. As we mentioned

earlier, whenever p27 is present, Rb can be expressed even in the presence of CycE or CycA. For the mutated cell cycle network, p27 is always zero and Rb cannot be expressed in a case where CycD is not present but CycE or CycA are active [23]. This mutation introduces a situation where both CycD and Rb might be inactive. As a result, in this mutated phenotype, the cell cycles in the absence of any growth factor. In other words, we consider the logical states in which both Rb and CycD are down-regulated as 'undesirable states', when p27 is mutated. Table II summarizes the mutated Boolean functions.

The Boolean functions in Table II are used to construct the PBN model for the cell cycle. To this end, we assume that the extra-cellular signal to the cell-cycle model is a latent variable. The growth factor is not part of the cell and its value is determined by the surrounding cells. The expression of CycD changes independently of the cell's content and reflects the state of the growth factor. Depending on the expression status of CycD, we obtain two constituent Boolean networks for the PBN. The first constituent Boolean network is determined from Table II when the value of CycD is equal to zero. Similarly, the second constituent Boolean network is determined by setting the variable of CycD to one. Here, we set the perturbation probabilities equal to $10^{-3}$.

According to Table II, the cell-cycle PBN consists of nine genes: CycD, Rb, E2F, CycE, CycA, Cdc20, Cdh1, UbcH10, and CycB. The above order of genes is used in the binary representation of the logical states, with CycD as the most significant bit and CycB as the least significant bit. This order of genes in the logical states facilitates the presentation of our results and does not affect the computed control policies.

Having CycD and Rb as the most significant genes, we assume that the down regulations of the CycD and Rb, i.e. the cell growth in the absence of growth factors, is undesirable. Consequently, the state-space is partitioned into undesirable states and desirable states. Application of methods developed for control of gene regulatory networks requires the designation of desirable and undesirable states, and this depends upon the existence of

Table I. Boolean functions of normal mammalian cell cycle.

| Product | Predictors |
|---|---|
| $CycD$ | Input |
| $Rb$ | $(\overline{CycD} \wedge \overline{CycE} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \overline{CycD} \wedge \overline{CycB})$ |
| $E2F$ | $(\overline{Rb} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge \overline{Rb} \wedge \overline{CycB})$ |
| $CycE$ | $(E2F \wedge \overline{Rb})$ |
| $CycA$ | $(E2F \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge (\overline{Cdh1 \wedge UbcH10})) \vee (CycA \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge (\overline{Cdh1 \wedge UbcH10}))$ |
| $p27$ | $(\overline{CycD} \wedge \overline{CycE} \wedge \overline{CycA} \wedge \overline{CycB}) \vee (p27 \wedge (\overline{CycE \wedge CycA}) \wedge \overline{CycB} \wedge \overline{CycD})$ |
| $Cdc20$ | $CycB$ |
| $Cdh1$ | $(\overline{CycA} \wedge \overline{CycB}) \vee (Cdc20)$ |
| $UbcH10$ | $(\overline{Cdh1}) \vee (Cdh1 \wedge UbcH10 \wedge (Cdc20 \vee CycA \vee CycB))$ |
| $CycB$ | $(\overline{Cdc20} \wedge \overline{Cdh1})$ |

relevant biological knowledge. In the cell-cycle example when p27 is mutated, we consider the logical states in which both Rb and CycD are down-regulated as undesirable states. We assume that the cost of the logical states with down-regulated Rb and CycD is higher than that for the states in which these two genes are not simultaneously down-regulated.

Table II. Mutated Boolean functions of mammalian cell cycle.

| Product | Predictors |
|---|---|
| $CycD$ | Input |
| $Rb$ | $(\overline{CycD} \wedge \overline{CycE} \wedge \overline{CycA} \wedge \overline{CycB})$ |
| $E2F$ | $(\overline{Rb} \wedge \overline{CycA} \wedge \overline{CycB})$ |
| $CycE$ | $(E2F \wedge \overline{Rb})$ |
| $CycA$ | $(E2F \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge (\overline{Cdh1 \wedge UbcH10})) \vee (CycA \wedge \overline{Rb} \wedge \overline{Cdc20} \wedge (\overline{Cdh1 \wedge UbcH10}))$ |
| $Cdc20$ | $CycB$ |
| $Cdh1$ | $(\overline{CycA} \wedge \overline{CycB}) \vee (Cdc20)$ |
| $UbcH10$ | $(\overline{Cdh1}) \vee (Cdh1 \wedge UbcH10 \wedge (Cdc20 \vee CycA \vee CycB))$ |
| $CycB$ | $(\overline{Cdc20} \wedge \overline{Cdh1})$ |

# CHAPTER III

## BOOLEAN NETWORKS AND BIDIRECTIONAL GENE RELATIONSHIPS*

Various models have been proposed for gene regulatory networks [24] and great efforts have been made on the inference of these networks from gene expression data. Perhaps the key issue concerning network inference is the large space of networks from which a model must be selected in relation to the amount of data typically available. This dimensionality problem drives inference in two directions: (1) towards coarse-grained models that require less data for inference [25], and (2) application of biological constraints [26]. This Chapter concerns the inference of Boolean networks. For these networks, several inference methods have been proposed [27, 28, 5, 29]. These methods generally assume time-course data; however, here we are concerned with inference from time-independent data, the kind of data one typically obtains from microarray studies involving human subjects. In this context, it is generally assumed that the data come from the steady state of the network.

The long-run behavior of a Boolean network is characterized by its attractor cycles. The attractor cycles in Boolean networks modeling biological systems are typically associated with phenotypes and tend to be short [17, 30, 3], with biological state stability contributing to singleton attractors [31]. Singleton attractors have been associated with phenotypes such as cell proliferation and apoptosis [32]. For this reason, in the absence of time-course data to indicate the contrary, it is sometimes assumed that the data states represent singleton attractors. This assumption is enhanced when a Hamming-distance filter is applied to the data states to act as a noise filter, because the filter results in a small number of data states, each differing significantly among the components of the states [33].

---

* Reprinted with permission from "Inference of Boolean Networks under Constraint on Bidirectional Gene Relationships" by G. Vahedi, I. Ivanov, E. R. Dougherty, 2009, *IET Systems Biology*, 3, 191-202, Copyright 2009 by IET Systems Biology.

One method proposed for inference from steady-state data involves the coefficient of determination (CoD) [34]. Given a set of predictor variables and a target variable to be predicted, the CoD measures the relative decrease in prediction error when using the predictor variables in comparison to using the best estimate of the target in the absence of knowledge concerning other variables. The CoD was the first method used to infer probabilistic Boolean networks (PBNs) [5].

A fundamental issue is that, without time-course data, the CoD cannot provide information on the direction of prediction. This problem manifests itself in the situation where, if gene $a$ is a high CoD predictor of gene $b$, then gene $b$ is typically a high CoD predictor of gene $a$. We refer to this situation as a *bidirectional relationship* between genes $a$ and $b$. The presence of bidirectional relationships affects the attractor structure of a Boolean network, and this impacts the inference process with the result being that the inferred network possesses spurious attractor cycles. The problem is sufficiently troublesome that it has suppressed the use of CoD inference methods. The inference methods that have taken its place are primarily based on the attractor structure, with either secondary or no concern for the predictive relations between individual genes in the network [22, 33]. This kind of approach is natural when attractor structure is of primary interest.

In this chapter, we will accomplish both goals in network design: preservation of attractor structure and connectivity based on strong gene prediction. To accomplish this aim, we investigate the bidirectional effects for Boolean networks with connectivity $K = 1$ or $K = 2$, the connectivity of a Boolean network being the maximum number of variables allowed for a Boolean function. As a consequence of our analysis, we propose a novel constrained CoD-based inference algorithm that performs significantly better than unconstrained CoD inference relative to the attractor structure. We note that the number of attractor cycles and their average lengths in random Boolean networks has recently been addressed for the case of connectivity $K = 1$, and it is clear that even this seemingly simple

structure presents challenge for both analytical and computational approaches [35].

We will begin by defining the bidirectional relationship among two genes of a network. We then investigate the effect of such relationships on the attractor structure on specific classes of Boolean networks. After discovering how the bidirectional relationships influence the attractor structure of a Boolean network, and providing estimates of encountering such relationships and particular attractor structures, we discuss CoD-based inference. We then propose a novel algorithm that mitigates bidirectional relationships and we provide simulation results that support our analysis. Lastly, we present an application of the proposed algorithm to melanoma gene expression data and compare its performance to unconstrained CoD inference procedures.

## A. Bidirectional Relationships

Our particular interest is with how genes that are predictors of each other affect the attractor structure. As noted in the Introduction of this Chapter, when such pairs arise on account of network inference, they can lead to the existence of certain attractor structures. This motivates the following definition.

**Definition 1** *The genes $x_i$ and $x_j$ in a BN are said to have a **bidirectional relationship** iff $x_i \in W_j$ and $x_j \in W_i$. The relationship is said to be of **connectivity** $\mathbf{n}$ if $|W_i| = |W_j| = n$.*

To say that $x_i$ and $x_j$ have a bidirectional relationship of connectivity $n$ is to say that they have a bidirectional relationship and each has connectivity $n$. Alternatively, one might have defined the relationship to be of connectivity $n$ if $\max\{|W_i|, |W_j|\} = n$, or to be of connectivity $(m, n)$ if $|W_i| = m$ and $|W_j| = n$, the rationale behind the first alternative being to bound the complexity of the predictor relations and the second being to specify directly the predictor-set cardinalities. We have defined order $n$ as we have because it

characterizes the most complex case when one of the predictor sets has cardinality $n$. It is this maximum complexity that interests us.

We will investigate the effect of bidirectionality on the attractor structure, provide estimates of how often such bidirectional relationships happen, and derive a lower bound estimate for the probability of a BN with such relationships having at least one non-singleton attractor cycle. We first consider connectivity $1$ and show that there is at least one non-singleton attractor cycle in the BN. Next we consider connectivity $2$. There we will see that even such a minimal increase of the cardinality of the predictor sets complicates the analysis of the attractor structure.

### 1. Connectivity-1 Didirectionality

**Proposition 4** *If there are two genes in a BN having a bidirectional relationship of connectivity 1, then the BN has at least one non-singleton attractor cycle.*

**Proof**

Without loss of generality assume the two genes are $x_1$ and $x_2$. There are four possible transition pairs of predictor functions for these genes: (1) $f_1 \equiv x_2$ and $f_2 \equiv x_1$; (2) $f_1 \equiv x_2$ and $f_2 \equiv \overline{x_1}$; (3) $f_1 \equiv \overline{x_2}$ and $f_2 \equiv x_1$; and (4) $f_1 \equiv \overline{x_2}$ and $f_2 \equiv \overline{x_1}$, where the overbar denotes negation.

Consider the first possible pair: $f_1 \equiv x_2$ and $f_2 \equiv x_1$. If the transitions start from the point $01\mathbf{y}$, then after finitely many transitions, the BN will enter an attractor $01\mathbf{x_0}$ or $10\mathbf{y_0}$, where $\mathbf{y}$, $\mathbf{y_0}$ and $\mathbf{x_0}$ denote vectors of the remaining gene values. Assume that the first visited attractor state is $01\mathbf{x_0}$ (the other possibility $10\mathbf{y_0}$ can be considered in the same way). Because $x_1$ and $x_2$ depend only on each other and $01\mathbf{x_0}$ is an attractor state, from this point on the network must follow a transition sequence of the form $01\mathbf{x_0}$,$10\mathbf{x_1}$,$01\mathbf{x_2}$,..., $01\mathbf{x}_k$, where $\mathbf{x}_k = \mathbf{x_0}$ and $\mathbf{x}_r \neq \mathbf{x_0}$ for $1 \leq r < k$. Thus, the sequence forms an attractor cycle

of length $k = 2m > 1$. It is straightforward to show that similar cycles are formed when $f_1 \equiv \overline{x_2}$ and $f_2 \equiv \overline{x_1}$.

Next, we consider the predictor functions pair: $f_1 \equiv x_2$ and $f_2 \equiv \overline{x_1}$. If the transitions start from any point of the form $x_1x_2\mathbf{y}$, then after finitely many transitions the BN will enter an attractor state that is of one of the following forms: $00\mathbf{x_0}$, $01\mathbf{y_0}$, $10\mathbf{z_0}$ or $11\mathbf{u_0}$. Here we consider the case when the first visited attractor state is $00\mathbf{x_0}$ (the other possibilities can be considered similarly). Because $x_1$ and $x_2$ depend only on each other and $00\mathbf{x_0}$ is an attractor state, from this point on the network must follow a transition sequence of the form, $00\mathbf{x_0}$,$01\mathbf{x_1}$,$11\mathbf{x_2}$,$10\mathbf{x_3}, \ldots, 00\mathbf{x}_k$, where $\mathbf{x}_k = \mathbf{x_0}$ and $\mathbf{x}_r \neq \mathbf{x_0}$ for $1 \leq r < k$. Thus, the sequence forms an attractor cycle of length $k = 4m > 1$. It is straightforward to show that similar cycles are formed when $f_1 \equiv \overline{x_2}$ and $f_2 \equiv x_1$. □

## 2. Connectivity-2 Bidirectionality

Suppose $x_1$ and $x_2$ have a bidirectional relationship of order 2 with $W_1 = \{x_2, x_4\}$ and $W_2 = \{x_1, x_3\}$. Because all predictor variables are essential, the following conditions cannot occur (refer to the truth tables for $f_1$ and $f_2$):

1. ($a_1 = c_1$ and $b_1 = d_1$) or ($a_2 = c_2$ and $b_2 = d_2$)
2. ($a_1 = b_1$ and $c_1 = d_1$) or ($a_2 = b_2$ and $c_2 = d_2$)

Table III. Truth tables for $f_1$ and $f_2$.

| $x_2$ | $x_3$ | $f_1$ |
|---|---|---|
| 0 | 0 | $a_1$ |
| 0 | 1 | $b_1$ |
| 1 | 0 | $c_1$ |
| 1 | 1 | $d_1$ |

| $x_1$ | $x_4$ | $f_2$ |
|---|---|---|
| 0 | 0 | $a_2$ |
| 0 | 1 | $b_2$ |
| 1 | 0 | $c_2$ |
| 1 | 1 | $d_2$ |

Moreover, any combination of $f_1$ and $f_2$ belongs to at least one of the following (not mutually exclusive) classes:

$F_1$: $a_1 = \overline{c_1}$ and $a_2 = \overline{c_2}$

$F_2$: $a_1 = \overline{c_1}$ and $b_2 = \overline{d_2}$

$F_3$: $b_1 = \overline{d_1}$ and $a_2 = \overline{c_2}$

$F_4$: $b_1 = \overline{d_1}$ and $b_2 = \overline{d_2}$

**Proposition 5** *If a BN possesses a pair of genes that have a bidirectional relastionship of connectivity* $2$*, then at least* $\frac{1}{8}$ *of the states in its state space cannot be singleton attractors of the network.*

**Proof**

Without loss of generality, suppose $x_1$ and $x_2$ have a bidirectional relationship of order $2$ with $W_1 = \{x_2, x_4\}$ and $W_2 = \{x_1, x_3\}$. To prove the proposition, we consider the following four cases: (a) $(f_1, f_2) \in F_1$ and $x_3x_4 = 00$; (b) $(f_1, f_2) \in F_2$ and $x_3x_4 = 01$; (c) $(f_1, f_2) \in F_3$ and $x_3x_4 = 10$; and (d) $(f_1, f_2) \in F_4$ and the states of the BN were $x_3x_4 = 11$.

For case (a), first consider $(f_1, f_2) \in F_1$ such that $a_1 = \overline{c_1} = 0$ and $a_2 = \overline{c_2} = 0$. Examination of the truth tables of $f_1$ and $f_2$, Table III, where $a_1 = \overline{c_1} = 0$ and $a_2 = \overline{c_2} = 0$, together with the assumed constant values of $x_3$ and $x_4$, shows that any state with $x_3x_4 = 00$ and $x_1 = \overline{x_2}$ cannot be a singleton attractor. A simple counting argument shows that the states where $x_3x_4 = 00$ and $x_1 = \overline{x_2}$ account for exactly $1/8$ of all of the states in state space. Reasoning in the same way, one can check that when $(f_1, f_2) \in F_1$ with $a_1 = \overline{c_1} = 1$ and $a_2 = \overline{c_2} = 1$, the states with $x_3x_4 = 00$ and $x_1 = x_2$ cannot be singleton attractors, and that there are exactly $1/8$ such states in the state space. To complete the analysis of case (a), consider the situation where $(f_1, f_2) \in F_1$ with $a_1 = \overline{c_1}$, $a_2 = \overline{c_2}$, and $a_1 = \overline{a_2}$. In this case, examination of the the truth tables of $f_1$ and $f_2$ shows that all of the states where

$x_3x_4 = 00$ cannot be singleton attractors. It is straightforward to count that exactly $1/4$ of the states in the state space are of this type.

Using similar arguments and symmetry considerations, one can show that the proposition holds for cases (b), (c) and (d). □

## B. Algorithm

### 1. CoD-based Inference of BNs

The *coefficient of determination* (*CoD*) is a general non-linear statistical method to select a set of predictors for a given gene. It measures the degree to which the transcriptional levels of an observed (predictor) gene set can be used to improve the prediction of the transcriptional level of a target gene relative to the best prediction in the absence of observations. If $x_i$, $W_i$, and $f_i$ are the target gene, the predictor set, and the predictor function for the target gene, respectively, then the CoD for the target gene $x_i$ is given by

$$\theta^i = \frac{\varepsilon_0 - \varepsilon(x_i, f_i(W_i))}{\varepsilon_0}$$

where $\varepsilon_0$ is the error of the best estimate of $x_i$ in the absence of any conditional variables and $\varepsilon(x_i, f_i(W_i))$ is the prediction error of the target gene according to the observations of the predictor set $W_i$ [34]. For minimum mean-square error estimation, $\varepsilon_0$ is the error of the prediction of $x_i$ with its mean.

The previous propositions explain why very often the CoD-inferred BNs possess spurious non-singleton attractors. We propose an algorithm to correct this undesirable behavior. We make the typical assumption that the data come from the steady state, and we apply the constraint that each data point is a singleton attractor.

Since the predictor function of each target gene is estimated from the steady-state data, not time-series data, each gene is a perfect estimator of itself (CoD equal to 1). To eliminate

this trivial case, no gene can be a member of its own predictor set. Therefore, for $n$ genes, for each target gene, there are $\sum_{m=1}^{k_i} C_m^{n-1}$ possible combinations for $W_i$, where $k_i$ is the maximum cardinality of $W_i$.

We employ a method called *full-logic* to estimate the predictor function and consequently the CoD for all possible combinations of predictor sets of each target gene. The CoD estimates a predictor function from the highest occurrence frequency of the target gene based on the values of all of the possible sets of predictor genes in the data set. More details regarding the full-logic method can be found in [36]. Note that there may be more than one high CoD predictor set for a target gene.

### 2. Singleton Attractor CoD Inference Algorithm

Based upon our analysis of bidirectional relationships, in particular, their effect on the attractor structure of a BN, we have formulated an algorithm that limits the number of such bidirectional relationships when predictor sets are chosen using the CoD method.

The algorithm's input is the binary gene expression data. The outcome of the algorithm is a BN with no non-singleton attractors. The following parameters are set in advance: (1) a threshold, $T_{CoD}$, for the CoD ($T_{CoD} = 0.7$ in our study); (2) the maximum number, $M_{BR}$, of bidirectional relationships allowed (keeping in mind that, as we have shown, there is a substantial probability of there being at least two genes with bidirectional relationships in an arbitrary BN, $M_{BR} = 3$ in our study); and (3) the minimum number, $m_A$, of points in the sample that appear as singleton attractors in the inferred BN ($m_A = 3$ in our study). Any predictor function that exceeds $T_{CoD}$ is called a *high CoD predictor function*. We now describe the Singleton Attractor CoD (SA-CoD) algorithm.

**Singleton Attractor CoD (SA-CoD) Inference Algorithm**

1. Estimate the CoD and $f_i$ for all the combinations of predictor sets $W_i$, for $i =$

$1, 2, \ldots, n$.

2. Save all $W_i$ and $f_i$ with CoD exceeding $T_{CoD}$. For each target gene, save the high CoD predictor sets and their associated predictor functions into two columns. The length, $C_i$, $i = 1, \ldots, n$, of both columns, depends on $T_{CoD}$.

3. Form a BN from $W_i$ and $f_i$ in step 2 such that the bidirectional relationships does not exceed $M_{BR}$. The algorithm never allows bidirectional relationships for connectivity 1 since this case guarantees the formation of non-singleton attractor cycles.

4. If there is a non-singleton attractor in the BN, then go to step 3; otherwise, continue.

5. If the number of data points appearing as singleton attractors in the BN is less than $m_A$, then go to step 3; otherwise, STOP.

The steps of the algorithm accomplish certain goals: step 3 limits bidirectional relationships, thereby limiting spurious attractor cycles resulting from bidirectional relationships; step 4 checks to see if any non-singleton attractor cycles have "slipped through" step 2; and step 5 insures that some minimal number of data points appears as singleton attractors in the inferred BN. The algorithm does not guarantee that the inferred BN will not contain singleton attractors that are not data points, but it does guarantee that there will be no non-singleton attractors. It is spurious non-singleton attractors that are ubiquitous in unconstrained CoD design. The algorithm does not guarantee that all data points will be singleton attractors, although it guarantees a minimum number, $m_A$, of these.

The algorithm can be run a number of times to produce a number of BNs, with each data point appearing in one or more BNs as a singleton attractor. This is somewhat similar to the design of PBNs under the requirement of contextual data consistency [33], where every data point must appear as a singleton attractor in at least one constituent BN of the PBN. There are, however, two key differences. First the method of [33] does not involve

the CoD, but instead involves a constrained optimization relative to the data distribution in the sample, and second, the number of BNs is determined by the data and it is theoretically certain that each data point will appear in at least one of the BNs as a singleton attractor. Nonetheless, the analogy is useful because the PBN design method first proposed in [5] applied CoD inference without constraint and then took combinations of high CoD predictor functions to construct the BNs forming the PBN, with the threshold ultimately determining the number of constituent BNs.

Regarding algorithm complexity, the total number of BNs that can be generated from high CoD predictor functions is $N = \prod_{i=1}^{n} C_i$, where $n$ is the total number of genes and $C_i$ is the number of high CoD predictor functions. Thus, the search space has $N$ members. In worst-case scenario, step 3 will be repeated $N$ times.

## C. Results and Discussion

### 1. Comparison of SA-CoD Algorithm with Unconstrained CoD Design

We have applied the preceding BN design procedure using gene-expression profiles from a study of 31 malignant melanoma samples explained in Chapter 1. The 7 genes used for the model are pirin, WNT5A, S100P, RET1, MART1, HADHB and STC2 (this being their order in the state space) and they were chosen from a set of $587$ genes from the data-set that have been subjected to an analysis of their ability to cross predict each other's state in a multivariate setting [21]. Table IV gives the 7-gene profiles for the 18 distinct data points and their corresponding frequencies. The assumption is that the data points correspond to the steady state of the underlying gene regulatory system.

The SA-CoD algorithm is applied 500 times to the gene expression data to generate 500 BNs. Based on the specifications of the algorithm, the BNs possess no non-singleton attractors and there are at least three data points as singleton attractors in each of them. We

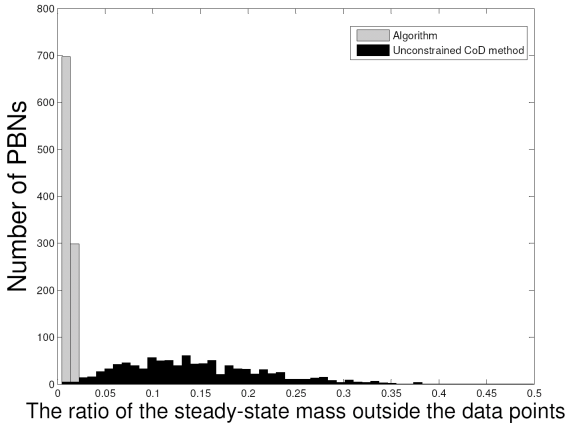



Fig. 1. Light histogram shows the proportion of the steady-state mass outside the data states in 1000 PBNs when the SA-CoD algorithm is used. Dark histogram shows the proportion of the steady-state mass outside the data states in 1000 PBNs when the unconstrained CoD method is used.

randomly choose 10 BNs from the pool of 500 BNs. Setting the perturbation probability equal to $0.01$, we generate a PBN from these 10 BNs. The PBN is run sufficiently long so that its steady-state distribution can be estimated, and the proportion of the steady-state mass lying outside the data states is computed. This procedure is repeated 1000 times to generate 1000 PBNs, in each case the proportion of the steady-state mass outside the data states being computed. These 1000 proportions are used to form the light histogram in Figure 1. The mass of this histogram is concentrated very close to 0.

To compare the performance of the SA-CoD algorithm with the unconstrained CoD method, we repeat the same experiment with the predictor sets and predictor functions with high CoD chosen without the constraint of the SA-CoD algorithm. Proceeding without constraint, 500 BNs are generated and 1000 PBNs composed of 10 BNs randomly chosen from the 500 are generated and run into their steady states. The dark histogram in Figure 1 is formed from the proportions of mass of the 1000 steady-state distributions lying outside

the data states. These are well dispersed between 0 and 0.35. By eliminating spurious attractors, the SA-CoD algorithm puts a much higher concentration of the steady-state mass on the data points.

A key issue for PBN design is to compose a PBN with enough BNs so that each data state appears as an attractor in the PBN (that is, appears as a an attractor in one of the constituent BNs) but not to include so many BNs that there is a large number of spurious attractors. To compare the SA-CoD algorithm with unconstrained design in this regard, in the next experiment we compare the number of data points appearing as attractors with the number of attractors that are not data points in a collection of $n$ BNs generated by the either the SA-CoD algorithm or unconstrained CoD design. Let $D$ be the number of distinct points in the data, $N$ be the number of data points appearing as attractors in the generated BNs, and $M$ be the number of non-data-point attractors appearing in the generated BNs. A reasonable measure of performance for the desired comparison is:

$$R = a(D - N) + (1 - a)M$$

where $0 \leq a \leq 1$, $a$ being chosen depending on what we want to emphasize. Smaller $R$ means better performance.

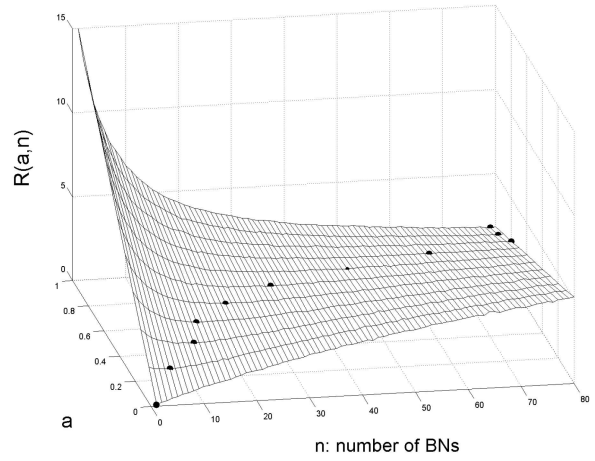
Since $N$ and $M$ are functions of the number $n$ of BNs, $R$ is a function of $n$. We compute $R(a, n)$ for $n = 1, 2, ..., 80$ and $0 \leq a \leq 1$ by taking $R(a, n)$ to be the average of $1000$ trials of computation of $R$, each trial involving randomly choosing $n$ BNs from a pool of $500$ designed BNs. Figure 2(a) shows the surface graph of $R(a, n)$ when using the SA-CoD algorithm. The dots on the surface indicate the minimum value of $R(a, n)$ for a given value of $a$, the value of $n$ for the minimum being the optimal number of BNs relative to the measure $R$. For small $a$, the emphasis is on avoiding spurious attractors and hence the optimum $n$ is smaller. For large $a$, the emphasis is on recovering data points as

attractors and hence the optimum $n$ is larger.

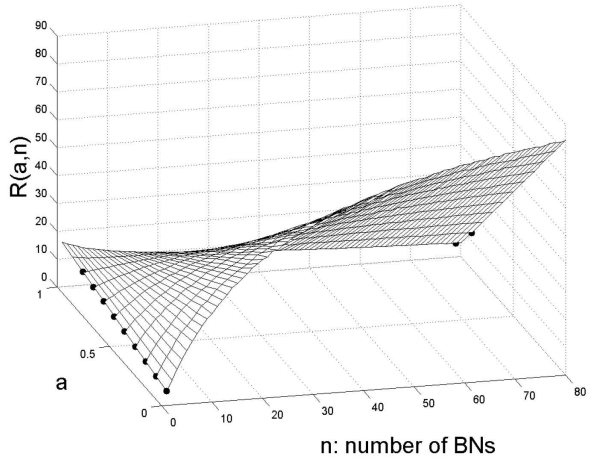
The differences between Figure 2(a) and Figure 2(b) demonstrate the benefits of the SA-CoD algorithm. First, we should point out the different scales of the graphs. The values of $R$ for unconstrained CoD design tend to greatly exceed those for the SA-CoD algorithm. Second, in Figure 2(b), the optimal number of BNs is 1 for all but very large values of $a$. This observation validates the point that, if we are concerned about spurious attractors, then unconstrained CoD design performs poorly.

Table IV. Expression profiles for melanoma.

| $Profile\#$ | $Gene$ | | | | | | | $Count$ |
|---|---|---|---|---|---|---|---|---|
| | pirin | WNT5A | S100P | RET1 | MART1 | HADHB | STC2 | |
| 1 | 1 | 0 | 0 | 1 | 1 | 1 | 1 | 2 |
| 2 | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 3 | 1 | 0 | 1 | 0 | 1 | 1 | 1 | 1 |
| 4 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 2 |
| 5 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 |
| 6 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 |
| 7 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 |
| 8 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 4 |
| 9 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |
| 10 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
| 11 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |
| 12 | 1 | 0 | 1 | 1 | 1 | 1 | 0 | 2 |
| 13 | 1 | 0 | 1 | 0 | 1 | 1 | 0 | 8 |
| 14 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| 15 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 16 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |
| 17 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 |
| 18 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 |




**(a)**




**(b)**

Fig. 2. Value of *R(a,n)* for *a* from 0 to 1 and *n* from 1 to 80 (a) SA-CoD algorithm, (b) unconstrained CoD method.

# CHAPTER IV

## TIMING IN PROBABILISTIC BOOLEAN NETWORKS

A major concern of translational genomics is to use the knowledge of gene regulation to design therapeutic strategies. Gene network modeling facilitates this effort by producing dynamical systems to serve as the mathematical basis for the derivation of optimal intervention strategies over time. To date, intervention has mainly focused on the external control of probabilistic Boolean networks (PBNs) via the associated discrete-time discrete-space Markov processes [11]. Given the accuracy of the model, there are two practical impediments to PBN-based intervention, both related to temporal issues. One of these concerns the lack of information regarding the sojourn time in any given state and the other concerns the practical problem of sampling. The first issue, the effect of sojourn time on the control, has been studied in [37]. In this work, we focus on the effect of discrete sampling.

While the physical evolution of the biological gene network occurs over continuous time, the PBN records only state transitions and contains no information on the time between transitions. The PBN model inherits this property from the original Boolean model, from which it was generalized [3]. Hence, the problem can be explained in the framework of the Boolean model. Fig. 3 shows the directed graph of a 3-gene Boolean network, where each 3-gene state corresponds to a gene-activity profile (GAP). Fig. 4 shows two continuous-time realizations that are equivalent from the perspective of the model of Fig. 3. In both Fig. 4(a) and (b), the initial state is “100”. We observe the evolution “100”→“010”→“001”, at which point there are no other changes because “001” is an attractor of the network. While equivalent from the perspective of the Boolean model, from the perspective of continuous time, the realizations of Fig. 4 (a) and (b) are not the same. For instance, in the second realization, the sojourn time in state “010” is much longer than in the first realization. If we are only interested in tracking the transitions, this may be of

no concern. On the other hand, suppose we are considering intervention and penalizing undesirable states. Then, if "010" is an undesirable state, the penalty should be greater in the second realization; that is, the penalty needs to consider the sojourn time in a state. This problem has been addressed in the framework of asynchronous PBNs by considering the process to be defined over continuous time and treating it as a semi-Markov process [37].

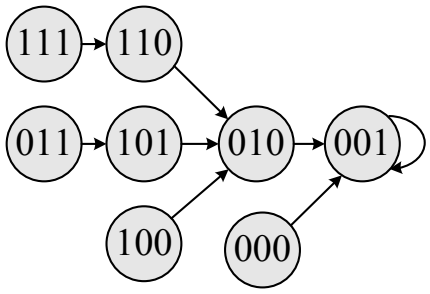



Fig. 3. Presentation of a directed graph for an arbitrary 3-gene Boolean network.

Whether one considers the original synchronous PBNs or asynchronous PBNs, implementation of the intervention policy requires nearly continuous observation because precise application requires the observation of all transitions. However, this is not generally the case in medical applications; rather, as with many engineering problems, the process is sampled at discrete time intervals and a decision to intervene or not must be made at each sample point. Since the process is not observed outside the sample points, it is impossible to know if, or how many, transitions have taken place between consecutive sample points.

In Fig. 4, the discrete-time process $\{Y_n, n \geq 0\}$ given by $Y_n = Z_{t_n}$ is called the jump chain of the continuous-time process $\{Z_t, t \geq 0\}$. Both synchronous and asynchronous PBNs deal with the jump chain under the assumption that the jumps (i.e. $t_0, t_1, \ldots$) are observed. The jump chains corresponding to realizations of Figs. 4(a) and (b) are equivalent. Fig. 4 also shows the sampled processes corresponding to each realization. The sampled process corresponding to Fig. 4(a) is "100"→"100"→"001"→"001" →"001"; for Fig. 4(b), it is "010"→"010" →"001"→"001"→"001". On account of sampling, "010" is

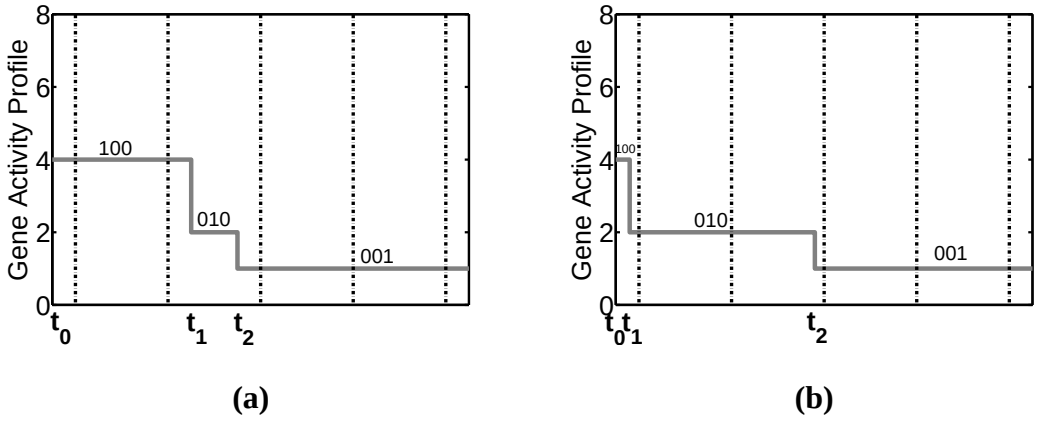



Fig. 4. Two examples of temporal gene activity profiles (GAP) for Fig. 3. The dash-dot vertical lines represent the sampling times.

missed in Fig. 4(a) and "100" is missed in Fig. 4(b). Whereas in a standard Boolean network self-transitions only occur for singleton attractors, the sampled process has self-transitions. Moreover, sojourn time is implicitly contained in the sampled process on account of these self-transitions. As with any sampling procedure, the sampling rate is crucial. The faster the rate, the less transitions will be missed and the more accurate will be the sojourn time estimates; the slower the rate, the more transitions will be missed and the less accurate will be the sojourn time estimates. In any event, in the presence of sampling, neither the synchronous or asynchronous PBN models will adequately reflect the dynamics of the network from the perspective of the decision process required for intervention. In this chapter, we propose a framework for gene regulatory networks, a sampling-rate-dependent PBN (SRD-PBN), that is capable of incorporating the sampling rate of temporal profile. Below, we mathematically define SRD-PBNs and expose a methodology to obtain optimal intervention strategies for such systems. We introduce SRD-PBNs in Section A. In Section B, we derive an optimal policy for SRD-PBNs with various properties for synthetic networks. We also consider a network obtained from melanoma gene-expression data.

## A. Sampling-Rate-Dependent Probabilistic Boolean Networks

A context-sensitive PBN disregards the information about the sojourn time in states present in temporal data. From another point of view, a context-sensitive PBN models the jump chain corresponding to the continuous-time process of interest. This means that in an arbitrary temporal profile such as Fig. 5, the observer can only apply intervention at instants $t_0, t_1, \ldots$ However, in medical applications, it is not known in advance when a transition (i.e. a jump) will occur. As such, a model based on applying treatment when a transition occurs may not conform with the reality and limitations of patient treatment. Time samples and state changes are unlikely to coincide perfectly and an intervention strategy must focus on the former not the latter.

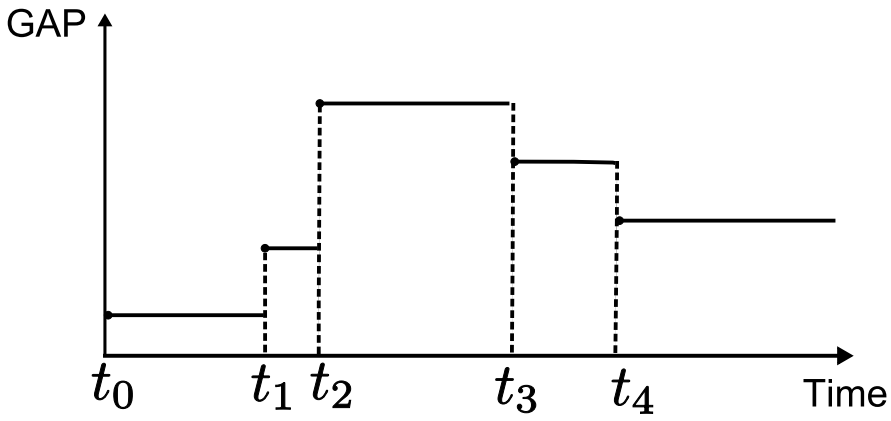



Fig. 5. An example of temporal gene activity profiles

Our objective in this work is to propose a discrete-time discrete-space model based on context-sensitive PBNs such that (i) it can embody the sojourn time of states into the network dynamics, (ii) it allows us to incorporate the sampling rate into the network's dynamics. A transition probability matrix must be derived for the state-space of a SRD-PBN under specific assumptions. Similar to other Markovian models, the transition probability

matrix derived for an SRD-PBN is sufficient to describe its dynamics. The task of finding the most effective intervention strategy can then be formulated as a sequential decision making problem via the associated transition probability matrix.

Let us first briefly introduce the underlying structure of an SRD-PBN. The states of the SRD-PBN take values in $\mathcal{S}$, as we defined for a context-sensitive PBN in Chapter II. Logical rules of different contexts determine the probability of jumps among GAPs. To coarsely capture the rate of change in the underlying biological system, the proposed framework requires two parameters, which are either known a priori or can be estimated from temporal data. These two parameters are the maximum rate of change among GAPs and the maximum rate of change among contexts. The rate of change between any two states, i.e. the average number of transitions between these two states in every unit of time, depends on the probability of jumps between these two states, the sampling period, the maximum rate of change among GAPs, and the maximum rate of change among contexts. Employing these parameters, we construct a **Q**-matrix on the state-space $\mathcal{S}$ of the SRD-PBN. This matrix is the generator of a continuous-time Markov chain. We are interested in the state of the continuous process only at discrete observation instants. The memoryless property of the continuous-time Markov chain allows us to model the dynamics of the sampled process as a discrete-time Markov chain. The transition probability matrix of this Markov chain is the transition probability matrix of the SRD-PBN. Below, we define the SRD-PBN in more details.

Given Boolean functions of context $c$, the probability of jumping from state $(c, x)$ to state $(c, x')$ is

$$P_{(c,x),(c,x')} \;=\; p^{D(x,x')}(1-p)^{n-D(x,x')} \;+\; (1-p)^n \mathbf{1}\left(\boldsymbol{f_c}(x) = x'\right), \tag{4.1}$$

where $p$ is the perturbation probability in the Boolean network. The Hamming distance be-

tween GAPs $x$ and $x'$ is denoted by $D(x, x')$. We use $\mathbf{1}(\cdot)$ to denote the indicator function. The successor state of GAP $x$ according to the Boolean functions of context $c$ is denoted by $\boldsymbol{f_c}(x)$. The first part of (4.1) corresponds to the transition probability due to gene perturbation. The probability of transitioning between GAPs $x$ and $x'$ based on the selected context $\boldsymbol{f_c}$ is presented as the second part of (4.1).

To include timing in our proposed model, given (4.1), we introduce matrix $Q$ which shows the rate of transitions among states in $\mathcal{S}$. We denote the maximum rate of change among GAPs by $\lambda$ and the maximum rate of change among contexts by $\gamma$. In practice, $\lambda$ can be estimated from temporal data. Knowledge of the ratio $\frac{\lambda}{\gamma}$, provided by experiments, would determine the value of $\gamma$. Matrix $Q$ is the generator of a continuous-time Markov chain. Let $Q = (q_{(c,x),(c',x')}; c, c' \in \{1, \ldots, k\}, x, x' \in \mathcal{W})$ denote the $\mathbf{Q}$-matrix of the continuous-time Markov chain $\{Z(t), t \geq 0\}$ whose state-space is $\mathcal{S}$. Elements of the $\mathbf{Q}$-matrix show the rate of change among states and can be computed in the following manner.

At any updating epoch, there are two independent processes: (i) a process that updates the GAP in the current context, (ii) a process that updates the context. There is null probabilities for both processes to occur at the same time. For the first process, we can compute the rate of change among GAPs $x$ and $x'$ in context $c$ as the product of $\lambda$, the maximum rate of change between GAPs, times the probability to jump from GAP $x$ to $x'$, i.e. $P_{(c,x),(c,x')}$. For the second process, we can compute the rate of change between contexts $c$ and $c'$ as the product of $\gamma$, the maximum rate of change between contexts, times the selection probability of context $c'$. Furthermore, in order to have a valid $\mathbf{Q}$-matrix (2.7), all diagonal elements of $Q$ should be defined such that the sum of elements in each row is zero. Thus, the rate of change between any two states $(c, x)$ and $(c', x')$ in $\mathcal{S}$ is defined as

$$q_{(c,x),(c',x')} = \begin{cases} \lambda P_{(c,x),(c,x')} & \text{if } c = c' \text{ and } x \neq x', \\ \gamma p_{c'} & \text{if } c \neq c' \text{ and } x = x', \\ 0 & \text{if } c \neq c' \text{ and } x \neq x', \\ -\sum_{X \neq x} \sum_{C \neq c} q_{(c,x),(C,X)} & \text{if } c = c' \text{ and } x = x', \end{cases} \tag{4.2}$$

where $p_{c'}$ is the selection probability of context $c'$.

We define $p_{ij}^t$ to be the probability that the continuous-time process $\{Z_t, t \geq 0\}$ associated to the SRD-PBN makes a transition from current state $i$ to successor state $j$ after $t$ units of time. Using this notation, $p_{ij}^t$ corresponds to $(i, j)$ entry in matrix $\mathbf{P}(t)$, where

$$\mathbf{P}(t) = e^{Qt}.$$

From the intervention perspective, we are interested in the dynamical behavior of the SRD-PBN at discrete observation instants, i.e. every $T$ units of time. Such a discrete-time model yields more information for the decision making process. Employing the memoryless property of the continuous-time Markov chain, we obtain a discrete-time Markov chain by taking samples from the continuous-time Markov chain at every $T$ units of time. This discrete-time model describes the dynamics of the SRD-PBN. For a given sampling period $T$, the transition probability matrix that expresses the dynamics of the SRD-PBN is computed as

$$\mathbf{P}(T) = e^{QT}, \tag{4.3}$$

where elements of $Q$ are defined in (4.2). We note that the transition probability matrix associated to the SRD-PBN is a function of the sampling period $T$. Optimal intervention strategies can then be derived for this SRD-PBN using the corresponding transition probability matrix.

**Example:** To illustrate the details of an SRD-PBN, we produce a simple 3-gene, 2-context example. Given the logical rules of each constituent Boolean network, one can draw the directed graphs corresponding to each Boolean network. Fig. 6 shows the directed graphs of the constituent Boolean networks in our simple example. The transition probability matrix corresponding to the context-sensitive PBN constructed based on these Boolean networks is shown in Table V. This transition probability matrix is computed following the methodology described in [37]. The switching probability $q$ is chosen to be 0.01 and there exists a gene perturbation probability of 0.01. It is clear that most of the states have zero self-transition probabilities. To construct the transition probability matrix of the SRD-PBN model, we first select $\lambda$ and $\gamma$ to be 0.1 and 0.05, respectively. The rate matrix $Q$ is computed based on (4.2). The transition probability matrix of the SRD-PBN corresponding to this matrix Q for sampling period of $T = 2$ is computed based on (4.3) and is shown in Table VI. A similar procedure is repeated for $T = 4$ and the transition probability matrix of the SRD-PBN is shown in Table VII. It is evident that the self-transition probabilities in Tables VI and VII are not zero. These values are different for $T = 2$ and $T = 4$. Intuitively, we expect a higher self-transition probability for a smaller sampling period and a lower self-transition probability for a larger sampling period. It can be seen that self-transition probabilities are larger in Table VI for $T = 2$ compared to Table VII for $T = 4$.

## B. Results and Discussion

Our prime goal of modeling gene regulatory networks from temporal gene expression data is to derive effective intervention strategies and beneficially alter the long-run behavior of the inferred model. From a practical point of view, at every observation point, this strategy decides which action should be applied to the underlying biological system. Provided that

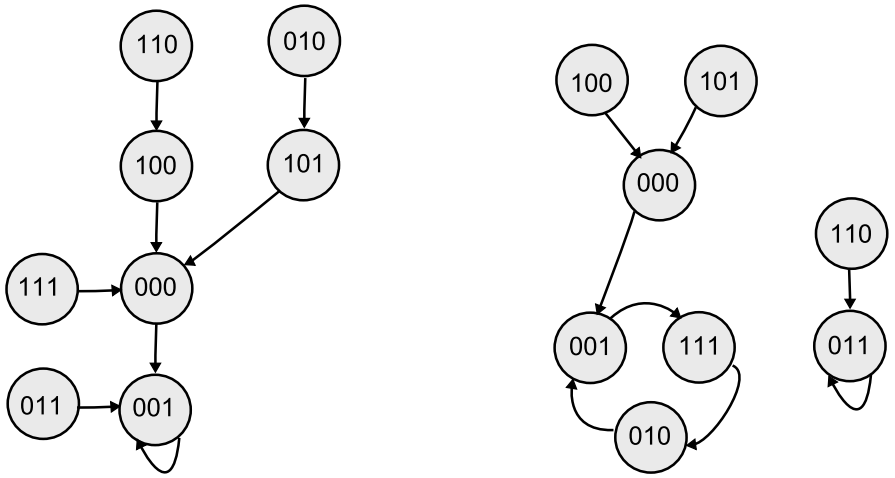



Fig. 6. Directed graphs of Boolean networks corresponding to the toy example

the model framework captures the dynamics of gene regulatory networks accurately, the derived intervention strategy would favorably alter the behavior of aberrant cells.

In this section, through numerical studies, we provide supporting evidence for the need to extend the original PBN framework. In the following simulations, the target gene, the gene responsible for aberrant behavior of the cell, is chosen to be the most significant gene in the GAP. We assume the up-regulation of the target gene is undesirable. Consequently, the state-space is partitioned into desirable states, $\mathcal{D}$, and undesirable states, $\mathcal{U}$. Since our objective is to down-regulate the target gene, a higher cost is assigned to destination states having an up-regulated target gene. Moreover, for a given status of the target gene for a destination state, a higher cost is assigned when the control is applied, versus when it is not. In practice, the cost values will have to mathematically capture the benefits and costs of intervention and the relative preference of states. These cost values will eventually be set with the help of physicians in accordance with their clinical judgement. Although this is not feasible within current medical practice, we do believe that such an approach will

Table V. Transition probability matrix of the context-sensitive PBN

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.73507 | 0.0073508 | 7.425e-005 | 0.0073508 | 7.425e-005 | 7.425e-005 | 7.5e-007 | 0 | 0.24502 | 0.0024503 | 2.475e-005 | 0.0024503 | 2.475e-005 | 2.475e-005 | 2.5e-007 |
| 0.0073508 | 0.72772 | 7.425e-005 | 0.0073508 | 7.425e-005 | 0.0073508 | 7.5e-007 | 7.425e-005 | 0.0024503 | 0 | 2.475e-005 | 0.0024503 | 2.475e-005 | 0.0024503 | 2.5e-007 | 0.2426 |
| 0.0073508 | 7.425e-005 | 0 | 0.0073508 | 7.425e-005 | 0.72773 | 0.0073508 | 7.425e-005 | 0.0024503 | 0.2426 | 0 | 0.0024503 | 2.475e-005 | 2.5e-007 | 0.0024503 | 2.475e-005 |
| 7.425e-005 | 0.73507 | 0.0073508 | 0 | 7.5e-007 | 7.425e-005 | 7.425e-005 | 0.0073508 | 2.475e-005 | 0.0024503 | 0.0024503 | 0.24257 | 2.5e-007 | 2.475e-005 | 2.475e-005 | 0.0024503 |
| 0.73507 | 7.425e-005 | 7.425e-005 | 7.5e-007 | 0 | 0.0073508 | 0.0073508 | 7.425e-005 | 0.24502 | 2.475e-005 | 2.475e-005 | 2.5e-007 | 0 | 0.0024503 | 0.0024503 | 2.475e-005 |
| 0.7278 | 0.0073508 | 7.5e-007 | 7.425e-005 | 0.0073508 | 0 | 7.425e-005 | 0.0073508 | 2.475e-005 | 0.0024503 | 2.5e-007 | 2.475e-005 | 0.0024503 | 0 | 0.2426 | 0.0024503 |
| 7.425e-005 | 7.5e-007 | 0.0073508 | 7.425e-005 | 0.73507 | 7.425e-005 | 0 | 0.0073508 | 0.2426 | 2.5e-007 | 0.0024503 | 2.475e-005 | 0.0024503 | 2.475e-005 | 0 | 0.0024503 |
| 0.72773 | 7.425e-005 | 7.425e-005 | 0.0073508 | 7.425e-005 | 0.0073508 | 0.0073508 | 0 | 2.5e-007 | 2.475e-005 | 0.2426 | 0.0024503 | 2.475e-005 | 0.0024503 | 0.0024503 | 0 |
| 0 | 0.24502 | 0.0024503 | 2.475e-005 | 0.0024503 | 2.475e-005 | 2.475e-005 | 2.5e-007 | 0 | 0.73507 | 0.0073508 | 7.425e-005 | 0.0073508 | 7.425e-005 | 7.425e-005 | 7.5e-007 |
| 0.0024503 | 0.24257 | 2.475e-005 | 0.0024503 | 2.475e-005 | 0.0024503 | 2.5e-007 | 2.475e-005 | 0.0073508 | 0 | 7.425e-005 | 0.0073508 | 7.425e-005 | 0.0073508 | 7.5e-007 | 0.7278 |
| 0.0024503 | 2.475e-005 | 0 | 0.0024503 | 2.475e-005 | 0.24258 | 0.0024503 | 2.475e-005 | 0.0073508 | 0.7278 | 0 | 0.0073508 | 7.425e-005 | 7.5e-007 | 0.0073508 | 7.425e-005 |
| 2.475e-005 | 0.24502 | 0.0024503 | 0 | 2.5e-007 | 2.475e-005 | 2.475e-005 | 0.0024503 | 7.425e-005 | 0.0073508 | 0.0073508 | 0.72772 | 7.5e-007 | 7.425e-005 | 7.425e-005 | 0.0073508 |
| 0.24502 | 2.475e-005 | 2.475e-005 | 2.5e-007 | 0 | 0.0024503 | 0.0024503 | 2.475e-005 | 0.73507 | 7.425e-005 | 7.425e-005 | 7.5e-007 | 0 | 0.0073508 | 0.0073508 | 7.425e-005 |
| 0.2426 | 0.0024503 | 2.5e-007 | 2.475e-005 | 0.0024503 | 0 | 2.475e-005 | 0.0024503 | 7.425e-005 | 0.0073508 | 7.5e-007 | 7.425e-005 | 0.0073508 | 0 | 0.7278 | 0.0073508 |
| 2.475e-005 | 2.5e-007 | 0.0024503 | 2.475e-005 | 0.24502 | 2.475e-005 | 0 | 0.0024503 | 0.7278 | 7.5e-007 | 0.0073508 | 7.425e-005 | 0.0073508 | 7.425e-005 | 0 | 0.0073508 |
| 0.24258 | 2.475e-005 | 2.475e-005 | 0.0024503 | 2.475e-005 | 0.0024503 | 0.0024503 | 0 | 7.5e-007 | 7.425e-005 | 0.7278 | 0.0073508 | 7.425e-005 | 0.0073508 | 0.0073508 | 0 |

Table VI. Transition probability matrix of SRD-PBN for sampling period T = 2

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.37738 | 0.29685 | 0.0025193 | 0.0008308 | 0.0022317 | 0.0014735 | 4.8756e-005 | 0.0026109 | 0.17417 | 0.11432 | 0.0040765 | 0.00044188 | 0.0010277 | 0.00048008 | 0.0001188 | 0.021425 |
| 0.0051341 | 0.66138 | 0.0014114 | 0.0030301 | 5.637e-005 | 0.0031601 | 6.0717e-005 | 0.0097065 | 0.0015781 | 0.23749 | 0.011406 | 0.0016716 | 2.2949e-005 | 0.0013831 | 0.0003802 | 0.062127 |
| 0.062988 | 0.025801 | 0.37564 | 0.002299 | 0.0015365 | 0.21034 | 0.003429 | 0.0019038 | 0.01262 | 0.054741 | 0.17501 | 0.0013961 | 0.00037188 | 0.050619 | 0.010816 | 0.010485 |
| 0.0018478 | 0.28873 | 0.002432 | 0.38578 | 2.3813e-005 | 0.0014393 | 4.4532e-005 | 0.0036407 | 0.00040847 | 0.063794 | 0.00299 | 0.2356 | 6.0167e-006 | 0.00036679 | 8.8608e-005 | 0.012803 |
| 0.22187 | 0.079318 | 0.00072494 | 0.00015764 | 0.37664 | 0.0024982 | 0.002244 | 0.00048444 | 0.10249 | 0.031863 | 0.00084154 | 8.1732e-005 | 0.17387 | 0.0011084 | 0.0012378 | 0.0045695 |
| 0.21324 | 0.079173 | 0.00070957 | 0.00018836 | 0.0040734 | 0.37565 | 0.0083255 | 0.002574 | 0.060811 | 0.024153 | 0.00092305 | 8.5038e-005 | 0.0014544 | 0.17354 | 0.050526 | 0.0045683 |
| 0.071732 | 0.017277 | 0.0024065 | 6.3607e-005 | 0.21273 | 0.0013257 | 0.376 | 0.0024078 | 0.070736 | 0.014919 | 0.0015871 | 4.9862e-005 | 0.051673 | 0.00034222 | 0.17372 | 0.0030368 |
| 0.21154 | 0.077929 | 0.0089785 | 0.002416 | 0.0012851 | 0.0036983 | 0.0022689 | 0.37582 | 0.051094 | 0.031437 | 0.051227 | 0.0014448 | 0.00031749 | 0.001226 | 0.0013476 | 0.17797 |
| 0.17565 | 0.12565 | 0.0024739 | 0.00039997 | 0.0010368 | 0.00074174 | 6.7046e-005 | 0.010044 | 0.37689 | 0.22611 | 0.014053 | 0.00099915 | 0.0022213 | 0.00084704 | 0.00032511 | 0.062497 |
| 0.011417 | 0.23929 | 0.0098272 | 0.001504 | 8.64e-005 | 0.0027633 | 0.00034266 | 0.050829 | 0.0039058 | 0.39764 | 0.060426 | 0.003911 | 5.5836e-005 | 0.0030169 | 0.001496 | 0.21348 |
| 0.012655 | 0.064714 | 0.17491 | 0.0014051 | 0.00035111 | 0.050907 | 0.0011899 | 0.0099297 | 0.0050204 | 0.21462 | 0.38697 | 0.0038826 | 0.00010197 | 0.009037 | 0.003685 | 0.060622 |
| 0.0004758 | 0.076766 | 0.0013999 | 0.23545 | 6.5807e-006 | 0.00040455 | 2.2788e-005 | 0.0015328 | 0.00012971 | 0.015123 | 0.0040482 | 0.65904 | 1.9695e-006 | 0.00011107 | 6.8701e-005 | 0.0054156 |
| 0.10266 | 0.035066 | 0.000488 | 7.4731e-005 | 0.17398 | 0.001155 | 0.0011304 | 0.0015062 | 0.22178 | 0.065926 | 0.0024392 | 0.00018174 | 0.37605 | 0.0023679 | 0.002877 | 0.012319 |
| 0.062368 | 0.016439 | 0.00047643 | 4.6578e-005 | 0.011113 | 0.17357 | 0.050527 | 0.0015158 | 0.069453 | 0.017142 | 0.0018487 | 8.9707e-005 | 0.0041756 | 0.37542 | 0.2103 | 0.0055079 |
| 0.061308 | 0.02493 | 0.0015044 | 7.6473e-005 | 0.051743 | 0.00043038 | 0.17358 | 0.0024897 | 0.21367 | 0.063818 | 0.0051748 | 0.00022306 | 0.011114 | 0.00023948 | 0.37546 | 0.014239 |
| 0.052715 | 0.023646 | 0.050781 | 0.0014047 | 0.00037883 | 0.010866 | 0.0013362 | 0.17493 | 0.010096 | 0.063371 | 0.21202 | 0.0038826 | 0.00010502 | 0.0035566 | 0.0036239 | 0.38728 |

become feasible when engineering approaches are integrated into translational medicine.

We postulate the following cost-per-stage in state $j$ under control $u$:

$$g(u,j) = \begin{cases} 0 & \text{if } u = 0 \text{ and } j \in \mathcal{D}, \\ 5 & \text{if } u = 0 \text{ and } j \in \mathcal{U}, \\ c & \text{if } u = 1 \text{ and } j \in \mathcal{D}, \\ 5{+}c & \text{if } u = 1 \text{ and } j \in \mathcal{U}, \end{cases} \tag{4.4}$$

where $c$ denotes the cost of control. We study the effect of $c$ in our simulations. A cost minimization framework is used to effectively trade-off the number of interventions and the likelihood of the network being in an undesirable state. An optimal control policy with regard to the cost values can be found via dynamic programming.

Table VII. Transition probability matrix of SRD-PBN for sampling period T = 4

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.17868 | 0.35925 | 0.0057011 | 0.0019278 | 0.0021134 | 0.0035642 | 0.00021701 | 0.016212 | 0.1333 | 0.20118 | 0.020986 | 0.0018779 | 0.0015748 | 0.0017692 | 0.00074308 | 0.070901 |
| 0.014692 | 0.50027 | 0.0090648 | 0.0040594 | 0.00019922 | 0.0057762 | 0.00043069 | 0.033177 | 0.0050446 | 0.25957 | 0.040292 | 0.0041558 | 8.4702e-005 | 0.0031117 | 0.0015077 | 0.11857 |
| 0.10288 | 0.090581 | 0.17324 | 0.0026056 | 0.0041568 | 0.1762 | 0.0090185 | 0.0089919 | 0.040905 | 0.10373 | 0.14019 | 0.003032 | 0.0014774 | 0.076391 | 0.030785 | 0.035828 |
| 0.0057851 | 0.33727 | 0.004429 | 0.20531 | 8.54e-005 | 0.0031632 | 0.00016202 | 0.011467 | 0.0018686 | 0.12417 | 0.013755 | 0.24697 | 3.2803e-005 | 0.0013484 | 0.00053547 | 0.043661 |
| 0.20516 | 0.17535 | 0.0022614 | 0.00067721 | 0.17329 | 0.0032648 | 0.0022213 | 0.0054509 | 0.15542 | 0.10424 | 0.0073772 | 0.00063446 | 0.13147 | 0.0021403 | 0.002346 | 0.028711 |
| 0.18835 | 0.1639 | 0.0021694 | 0.00066995 | 0.010183 | 0.17221 | 0.02385 | 0.0063215 | 0.10749 | 0.082778 | 0.0066183 | 0.00056804 | 0.003877 | 0.13076 | 0.076045 | 0.024191 |
| 0.13102 | 0.075247 | 0.003231 | 0.00029166 | 0.17834 | 0.0026047 | 0.17213 | 0.0048031 | 0.13651 | 0.061893 | 0.0059905 | 0.00034666 | 0.078014 | 0.0012002 | 0.1312 | 0.017175 |
| 0.1811 | 0.16663 | 0.025842 | 0.0030037 | 0.0022348 | 0.010116 | 0.0023946 | 0.17635 | 0.078679 | 0.10179 | 0.082779 | 0.0033655 | 0.00096658 | 0.0034649 | 0.0029033 | 0.15838 |
| 0.14216 | 0.24037 | 0.010431 | 0.001545 | 0.0016752 | 0.0039442 | 0.00038594 | 0.031819 | 0.17574 | 0.23275 | 0.04076 | 0.0026171 | 0.0020726 | 0.0021998 | 0.0012051 | 0.11032 |
| 0.035076 | 0.27152 | 0.029857 | 0.0033663 | 0.00049728 | 0.011092 | 0.0012222 | 0.079698 | 0.011315 | 0.24615 | 0.099865 | 0.0060475 | 0.00018346 | 0.0050184 | 0.0032555 | 0.19583 |
| 0.041474 | 0.13533 | 0.13886 | 0.003053 | 0.0013976 | 0.078705 | 0.0032107 | 0.030301 | 0.014961 | 0.20188 | 0.20765 | 0.0058504 | 0.00046527 | 0.025556 | 0.0099638 | 0.10134 |
| 0.0023676 | 0.17371 | 0.0032769 | 0.24628 | 3.8474e-005 | 0.0016271 | 8.52e-005 | 0.0049515 | 0.00071434 | 0.05067 | 0.0082044 | 0.49004 | 1.2965e-005 | 0.00055224 | 0.00024337 | 0.017234 |
| 0.15752 | 0.12513 | 0.0030728 | 0.00053267 | 0.13185 | 0.0026918 | 0.0019382 | 0.0096008 | 0.20416 | 0.12842 | 0.012724 | 0.00086452 | 0.17237 | 0.0027507 | 0.0033873 | 0.042997 |
| 0.11652 | 0.069673 | 0.0020666 | 0.00028091 | 0.03164 | 0.13106 | 0.075975 | 0.0051168 | 0.12448 | 0.05981 | 0.0065406 | 0.00039252 | 0.010492 | 0.17138 | 0.17558 | 0.01899 |
| 0.11116 | 0.094641 | 0.0045875 | 0.00047552 | 0.078157 | 0.001985 | 0.13066 | 0.010669 | 0.19213 | 0.11775 | 0.015097 | 0.00087665 | 0.026864 | 0.0010117 | 0.17157 | 0.042379 |
| 0.088771 | 0.087311 | 0.078238 | 0.0030548 | 0.0013755 | 0.031372 | 0.0027699 | 0.13944 | 0.029664 | 0.11819 | 0.18652 | 0.0058631 | 0.00046275 | 0.0095599 | 0.005974 | 0.21143 |

In our simulation studies, our objective is to show that an optimal policy derived for the current definition of context-sensitive PBN will no longer be optimal if we include the timing information of temporal data into the dynamics of gene regulatory networks. To this end, we generate synthetic SRD-PBNs and corresponding context- sensitive PBNs. We compute the cost induced by the optimal policy derived for the context-sensitive PBN and the cost induced by the optimal policy derived for the SRD-PBN, when both are applied to a sequence of data generated from the SRD-PBN. These two cost values are compared in our simulation studies. An SRD-PBN accommodates the sampling rate, which is in this simulation identical to the intervention rate. The goal of this study is to measure how costly it is to apply an optimal policy derived for a context-sensitive PBN to a sequence of data generated based on an SRD-PBN. In the following, we first consider synthetically generated SRD-PBNs. Furthermore, we study the gene regulatory network inferred from metastatic melanoma gene expression data.

### 1. Synthetic Networks

We generate SRD-PBNs in the following manner. Each SRD-PBN consists of 2 constituent Boolean networks. Each Boolean function of a Boolean network is randomly generated with a random bias. Given a set of Boolean networks, we generate various SRD-PBNs.

We let $\gamma = 0.01$. We vary the value of $\lambda$ from $0.05$ to $4$ with step-size $0.2$. We choose the gene perturbation probability of $0.01$. The constituent Boolean networks are selected with equal probabilities. Furthermore, for the given set of Boolean networks, we generate the corresponding context sensitive PBNs for switching probability $q = 0.01$. We let the observation period to be every $1$ unit of time, i.e. $T = 1$.

Using dynamic programming, given the cost-per-stage defined in (4.4), we derive an optimal intervention policy $\mu^*_{\text{srd}}$ for an SRD-PBN. Our goal is to estimate the average total discounted cost induced by $\mu^*_{\text{srd}}$ for a sequence of data generated from the SRD-PBN. To this end, we generate synthetic time-course data for 1000 time-steps from the SRD-PBN model while $\mu^*_{\text{srd}}$ is applied. We estimate the discounted cost by accumulating the discounted cost of each state given the action at that state. This procedure is repeated $10,000$ times for random initial states and the average of the induced discounted cost is computed. Likewise, an optimal policy $\mu^*_{\text{cs}}$ for a context-sensitive PBN is derived. Following a similar procedure, $\mu^*_{\text{cs}}$ is applied to the SRD-PBN, which we already described, and the average discounted cost is computed. Moreover, we compute the average discounted cost of a sequence of time-course data for an SRD-PBN in the absence of intervention.

In summary, for each set of Boolean networks, we have the following: ($\bar{C}^{\text{srd}}$) average total discounted cost induced by $\mu^*_{\text{srd}}$ on the SRD-PBN; ($\bar{C}^{\text{cs}}$) average total discounted cost induced by $\mu^*_{\text{cs}}$ on the SRD-PBN; ($\bar{C}^{\text{woc}}$) average total discounted cost induced in the absence of any intervention on SRD-PBN. The preceding procedure is repeated for $1000$ random Boolean networks, thereby yielding 1000 values for each statistic: $\bar{C}^{\text{srd}}_1, \ldots, \bar{C}^{\text{srd}}_{1000}; \bar{C}^{\text{cs}}_1, \ldots, \bar{C}^{\text{cs}}_{1000}; \bar{C}^{\text{woc}}_1, \ldots, \bar{C}^{\text{woc}}_{1000}$.

Using these, we compare the effect of $\mu^*_{\text{srd}}$ and $\mu^*_{\text{cs}}$ on an SRD-PBN by the empirical averages $\mathbf{M}[C^{\text{srd}}]$ of $\bar{C}^{\text{srd}}_1, \ldots, \bar{C}^{\text{srd}}_{1000}$; $\mathbf{M}[C^{\text{cs}}]$ of $\bar{C}^{\text{cs}}_1, \ldots, \bar{C}^{\text{cs}}_{1000}$; and $\mathbf{M}[C^{\text{woc}}]$ of $\bar{C}^{\text{woc}}_1, \ldots, \bar{C}^{\text{woc}}_{1000}$. We define the gain obtained by each intervention policy as the difference between the average discounted cost before and after intervention. $G_{\text{srd}}$, the gain of

policy $\mu^*_{\text{srd}}$, is $\mathbf{M}[C^{\text{woc}}] - \mathbf{M}[C^{\text{srd}}]$ and $G_{\text{cs}}$, the gain of policy $\mu^*_{\text{cs}}$ applied to an SRD-PBN, is $\mathbf{M}[C^{\text{woc}}] - \mathbf{M}[C^{\text{cs}}]$. We are interested in $\mathbf{M}[C^{\text{cs}}] - \mathbf{M}[C^{\text{srd}}]$, which we refer to as $\Delta G$.

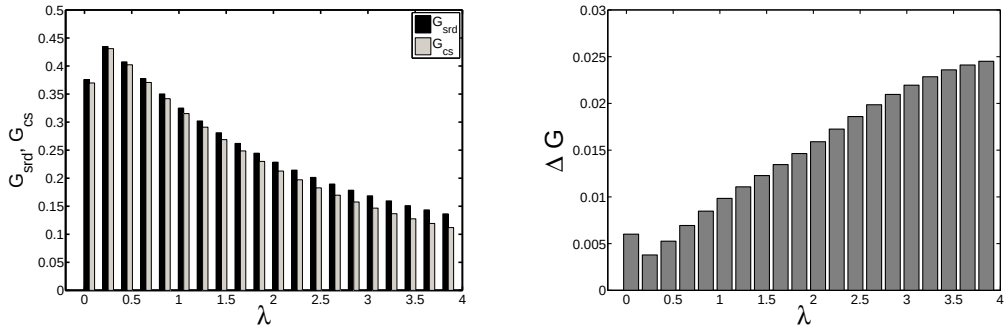
Figs. 7, 8, and 9 demonstrate the outcome of the above experiment for various values of cost of control $c$. It is evident that the intervention gains $G_{\text{srd}}$ and $G_{\text{cs}}$ are larger for smaller cost of intervention. The structure of a context-sensitive PBN is such that there is a transition to a new state after each unit of time, which corresponds to one change at every unit of time on average. When $\lambda$ is substantially smaller or larger than 1, $\Delta G$ is larger compare to the case where $\lambda$ is closer to 1, as is shown in Figs. 7, 8, 9. We should point out that the value of $\lambda$ for which $\Delta G$ attains its minimum depends on many factors, such as $\gamma$, the switching probability $q$ in context-sensitive PBN, and the cost of control. It is also observed that $\Delta G$ increases for larger cost of control.

We emphasize that this simulation study compares the gains obtained by two policies, the policy optimal for the SRD-PBN and the policy optimal for the context-sensitive PBN, when each is applied to SRD-PBN. Our objective is to show how poor the effect of an intervention policy derived for a context-sensitive PBN might be if the rate of change among observations is substantially different from 1.

### 2. Melanoma Gene Expression

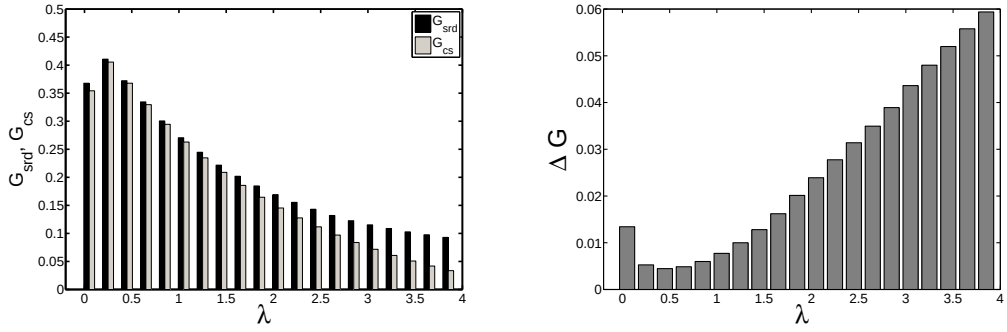
In this section, we consider the gene network corresponding to metastatic melanoma explained in Chapter II. We postulate the cost-per-stage in (4.4) with the cost of control $c$ being $0.1$. Since our objective is to down-regulate the WNT5A gene, a higher penalty is assigned for destination states having WNT5A up-regulated. Also, for a given WNT5A status for the destination state, a higher penalty is assigned when the control is active versus when it is not.

We generate the SRD-PBN corresponding to the melanoma-based data. Similar to the procedure explained in the previous section, we compute $\mu^*_{\text{cs}}$ and $\mu^*_{\text{srd}}$. Both policies are



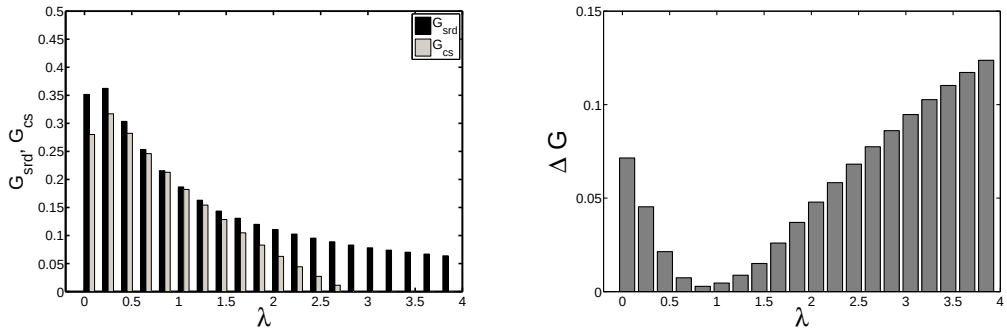

**(a)** **(b)**

Fig. 7. (a) $G_{\text{srd}}$, gain obtained by the policy optimal for SRD-PBN, and $G_{\text{cs}}$, the gain obtained by the policy optimal for context-sensitive PBN, when both are applied to SRD-PBN for various $\lambda$. (b) Difference between the gains, $\Delta G$, for various $\lambda$. The cost of control is 0.1.




**(a)** **(b)**

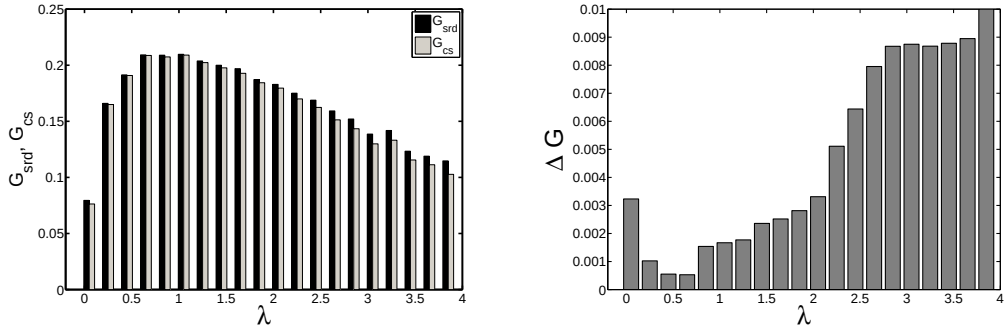
Fig. 8. (a) $G_{\text{srd}}$, gain obtained by the policy optimal for SRD-PBN, and $G_{\text{cs}}$, the gain obtained by the policy optimal for context-sensitive PBN, when both are applied to SRD-PBN for various $\lambda$. (b) Difference between the gains, $\Delta G$, for various $\lambda$. The cost of control is 1.0.




**(a)** **(b)**

Fig. 9. (a) $G_{\text{srd}}$, gain obtained by the policy optimal for SRD-PBN, and $G_{\text{cs}}$, the gain obtained by the policy optimal for context-sensitive PBN, when both are applied to SRD-PBN for various $\lambda$. (b) Difference between the gains, $\Delta G$, for various $\lambda$. The cost of control is 3.

applied to a sequence of data generated from the SRD-PBN and the cost corresponding to the policy is estimated. Fig. 10 shows the outcome of this experiment. Similar results are observed as $\lambda$ varies. We should emphasize that Fig. 10 corresponds to one network representing the melanoma gene expression data while Figs. 7, 8, and 9 consider average behavior of 1000 synthetically generated networks.




**(a)** **(b)**

Fig. 10. Simulation results corresponding to Melanoma study (a) $G_{\text{srd}}$, gain obtained by the policy optimal for SRD-PBN, and $G_{\text{cs}}$, the gain obtained by the policy optimal for context-sensitive PBN, when both are applied to SRD-PBN for various $\lambda$. (b) Difference between the gains, $\Delta G$, for various $\lambda$. The cost of control is 0.1.

# CHAPTER V

## OPTIMAL CYCLIC CONTROL POLICY*

Successful treatment of bacterial infections is largely a result of our ability to exploit the biochemical differences between bacteria and human cells so as to achieve toxic drug concentrations in the former while sparing the latter. Unfortunately, such high selectivity is at present elusive in the chemotherapy of human cancers. Hence, great efforts are required to determine dose schedules that maximize the benefit to toxicity ratio in cancer treatment [14]. To this end, chemotherapy is generally given in *cycles*: each treatment is followed by a *recovery phase*. During a recovery period, the side effects tend to gradually subside. Dose intensity is a measure of chemotherapy delivery that looks at the amount of drug delivered per unit of time. A higher drug dose intensity can be delivered by increasing the dose per cycle (dose escalation) or by reducing the interval between cycles (dose density).

For a given integrated drug effect, the chance of eradicating the tumor is maximized by delivering the most effective dose level of drug over as short a time as possible. Tumors given less time to grow between treatments are more likely to be eradicated. Administering high quantities of drugs at the beginning of a chemotherapy cycle might fail for two reasons. First, levels higher than a certain concentration may not increase the killing rate of cancer cells. Second, even if they did, the toxicity could be intolerable to the patient. In practice, optimizing the schedule means determining a way to give the maximum integrated effect over as short a time as possible, consistent with reasonable quality of life [14].

A prime objective of modeling genetic regulatory networks is to develop therapies based on gene regulation, in particular, the disruption or mitigation of aberrant gene func-

---

* Reprinted with permission from "Optimal Intervention Strategies for Cyclic Therapeutic Methods" by G. Vahedi, B. Faryabi, J.-F. Chamberland, A. Datta, and E. R. Dougherty, 2009, *IEEE Transactions on Biomedical Engineering*, 56, 281-291, Copyright 2009 by IEEE.

tion contributing to the pathology of a disease. Engineering therapeutic tools involve synthesizing nonlinear dynamical networks, analyzing these networks to characterize gene regulation, and developing intervention strategies to modify dynamical behaviors [38]. In this chapter, we derive an optimal cyclic intervention strategy for gene regulation in the context of probabilistic Boolean networks.

For intervention strategies proposed earlier [10, 11, 39, 16], at every state transition of the system, the intervention strategy dictates whether to apply treatment or not. In this chapter, our objective is to devise an effective intervention strategy under the constraint that intervention is permitted only every $W$ transitions, where $W \in \mathbb{N}$ denotes the length of the recovery period. An intervention strategy that is optimal for the case where intervention is permitted at every transition is not necessarily optimal (i.e. may not minimize the expected total discounted cost) if one is only permitted to apply treatment every $W$ transitions. We will refer to a policy that is optimal when intervention is permitted every transition as an *optimal one-transition policy*. Similarly, we refer to the policy that is optimal when intervention is permitted every $W$ transitions as an *optimal W-transition policy*.

We define a *treatment window* to be every $W$ transitions of the system. Intervention is permitted at the beginning of a treatment window. Thereafter, the system transitions $W - 1$ steps without intervention. To incorporate the cyclic constraint on interventions, we construct a Markov chain with an augmented state space based on the original Markov chain. An optimal cyclic intervention policy, i.e. optimal W-transition policy, can be found by solving the stochastic control problem for the Markov chain with the augmented state space via dynamic programming algorithms. However, this procedure maybe prohibitive due to the size of the augmented state space. We show that the augmented state space can be collapsed resulting in a compressed space of size equal to the original state space. We accomplish this reduction in the size of the state space by accumulating the expected cost of the system progressing during a period. The new cost function is used to select the

proper action when intervention is permitted. We establish the convergence of the dynamic programming algorithm and show how the optimal W-transition intervention strategy can be found. Furthermore, we compare the performance of an optimal W-transition policy to that of an optimal one-transition policy when intervention is applied every $W$ transitions. We show that although this may not be true in general, in our intervention framework, optimal one-transition policy can be used as an approximation of optimal W-transition policy.

## A. Optimal Control Strategy for Cyclic Therapeutic Methods

Our objective is to find an effective intervention policy when we are allowed to apply treatment only every $W$ transitions, in other words at times $t = 0, W, 2W, \ldots$ To incorporate this cyclic constraint in our mathematical framework, we construct a Markov chain with an augmented state space based on the original Markov chain. The new (augmented) state space is defined as

$$\tilde{\mathcal{S}} = \{(i, j) | i \in \{0, \ldots, N-1\}, j \in \{0, \ldots, W-1\}\},$$

where $N$ is the size of the original state space $\mathcal{S}$. There are two types of states in the augmented state space: state $(i, j)$ with $j = 0$, represented as $(i, 0)$, where intervention is permitted, and state $(i, j)$ with $j \neq 0$, where intervention is not permitted. In the augmented state space, the control $u$ is constrained to take values in $U(i, j)$, a given nonempty subset of $\mathcal{C}$. For the first type of states, $(i, 0)$, we have $U(i, 0) = \{0, 1\}$, while for the second type of states, $(i, j)$ where $j \neq 0$, we have $U(i, j) = \{0\}$.

The transition probabilities in the augmented state space are defined as a function of control $u$. For state $(i, 0)$, we define the probability of transitioning to state $(i', j')$ given

control $u$ as

$$p_{(i,0)(i',j')}(u) = \begin{cases} p_{i,i'}(u) & \text{if } j' = 1, \\ 0 & \text{otherwise}, \end{cases}$$

where $p_{i,i'}(u)$ denotes the probability of transitioning from state $i$ to state $i'$ under control $u$. On the other hand, for states $(i, j)$ where $j \neq 0$, control $u$ only admits one value, $u \in \{0\}$. For these states, the transition probability is defined as

$$p_{(i,j)(i',j')}(u = 0) = \begin{cases} p_{i,i'}(u = 0) & \text{if } j' = (j + 1) \bmod W \\ 0 & \text{otherwise}, \end{cases}$$

where $p_{i,i'}(u = 0) = p_{i,i'}$ denotes the uncontrolled probability of transitioning from state $i$ to state $i'$. It should be noted that $(j' = (j + 1) \bmod W)$ is true if either $(j' = j + 1)$ or $(j = W - 1$ and $j' = 0)$ is true. Considering that $u \in U(i, j)$, the probability of transitioning from state $(i, j)$ to state $(i', j')$ can be compactly defined as

$$p_{(i,j)(i',j')}(u) = \begin{cases} p_{i,i'}(u) & \text{if } j' = (j + 1) \bmod W, \\ 0 & \text{otherwise}. \end{cases} \tag{5.1}$$

Let us now consider an example to explain how the above definition simulates the cyclic intervention scenario. Assume that at time $t = 0$ we observe state $i$. At this time, we are allowed to apply control $u \in \{0, 1\}$. The augmented state corresponding to state $i$ at time $t = 0$ is $(i, 0)$. From augmented state $(i, 0)$, under control $u$, the system transitions to the augmented state $(i', 1)$ with probability $p_{i,i'}(u)$, where $u \in \{0, 1\}$. The probability of transitioning to any other state $(i', j)$, where $j \neq 1$, is zero. At time $t = 1$ and from augmented state $(i', 1)$, the system transitions to state $(i'', 2)$ with probability $p_{i',i''}(0)$ since $u \in \{0\}$. Likewise, one can consider transitions for $t = 2, \ldots, W - 2$. Similarly, assume that we observe state $k$ at time $t = W-1$. The probability of transitioning to the augmented state $(k', 0)$ is $p_{k,k'}(0)$. The probability of transitioning to any other state $(k', j)$, where

$j \neq 0$, is zero.

The cost-per-stage for transitioning from augmented state $(i, j)$ to augmented state $(i', j')$, given control $u$, is defined as

$$g(i,j,i',j',u) = \begin{cases} C+c & i' \in \mathcal{U} \text{ and } \{j=0 \text{ and } j'=1\} \text{ and } u=1, \\ C & i' \in \mathcal{U} \text{ and } \{j' = (j+1) \bmod W\} \text{ and } u=0, \\ c & i' \in \mathcal{D} \text{ and } \{j=0 \text{ and } j'=1\} \text{ and } u=1, \\ 0 & i' \in \mathcal{D} \text{ and } \{j' = (j+1) \bmod W\} \text{ and } u=0, \\ 0 & \text{otherwise}, \end{cases} \tag{5.2}$$

where $C$ and $c$ represent the cost of undesirable states and the cost of treatment (control), respectively. Given $u = 1$, we assign a cost to a transition from state $(i, j)$ to state $(i', j')$ only when $j = 0$ and $j' = 1$. In this case, if $i'$ is an undesirable state, the corresponding cost is $C + c$; if $i'$ is a desirable state, the only cost incurred is $c$. When $u = 0$, it is possible to transition to $(i', j')$ if $j' = (j + 1) \bmod W$ is true. In this case, if $i'$ is an undesirable state, the corresponding cost is $C$; if $i'$ is a desirable state, no cost is incurred. For all the other cases, no cost is assigned.

Based on (5.2), we define the expected immediate cost at state $(i, j)$ when control $u$ is selected by

$$\begin{aligned} \overline{g}(i,j,u) &= \sum_{i'=0}^{N-1} \sum_{j'=0}^{W-1} p_{(i,j)(i',j')}(u) g(i,j,i',j',u) \\ &= \sum_{i'=0}^{N-1} [p_{(i,j)(i',0)}(u) g(i,j,i',0,u) + \\ &\qquad \cdots + p_{(i,j)(i',W-1)}(u) g(i,j,i',W-1,u)]. \end{aligned}$$

In this equation, for each value of $i'$, only one term inside the brackets is non-zero (based

on the definition of the transition probabilities in (5.1)). Hence,

$$\overline{g}(i,j,u) = \sum_{i'=0}^{N-1} p_{(i,i')}(u) g(i,j,i',j',u),$$

where $j' = (j+1) \bmod W$ is true. Using the definition of $g(i,j,i',j',u)$ in (5.2), we have

$$\overline{g}(i,j,u) = \begin{cases} C \sum_{i' \in \mathcal{U}} p_{i,i'}(u=1) + c & \text{if } u = 1, \\ C \sum_{i' \in \mathcal{U}} p_{i,i'}(u=0) & \text{if } u = 0. \end{cases} \tag{5.3}$$

From (5.3), it is clear that $\overline{g}(i,j,u)$ does not depend on $j$, i.e. $\overline{g}(i,j,u) = \overline{g}(i,u)$.

As we explained in Chapter II, the dynamic programming algorithm captures how the optimal cost at $J_{k+1}$ propagates backward in time to the optimal cost $J_k$. For the augmented state space, we have

$$J_k(i,j) = \min_{u \in U(i,j)} \left[ \overline{g}(i,j,u) + \lambda \sum_{i'=0}^{N-1} \sum_{j'=0}^{W-1} p_{(i,j)(i',j')}(u)\ J_{k+1}(i',j') \right] \tag{5.4}$$
$$\forall (i,j) \in \tilde{\mathcal{S}}.$$

Since $\overline{g}(i,j,u) = \overline{g}(i,u)$, we can rewrite (5.4) as

$$J_k(i,j) = \min_{u \in U(i,j)} \left[ \overline{g}(i,u) + \lambda \sum_{i'=0}^{N-1} \sum_{j'=0}^{W-1} p_{(i,j)(i',j')}(u)\ J_{k+1}(i',j') \right], \tag{5.5}$$
$$\forall (i,j) \in \tilde{\mathcal{S}}.$$

Our goal is to derive the value functions for the original state space, i.e. $\mathcal{S}$, based on (5.5). To this end, for every treatment window starting with $i \in \mathcal{S}$, we accumulate the total discounted cost of all states in the window where no control can be applied and add it to the average cost of state $i$. We then show how the accumulated cost at the beginning of the $(s+1)$th window affects the accumulated cost at the beginning of the $s$th window, where $s = 0, 1, 2, \ldots$ This approach is in accord with the dynamic programming technique that ranks decisions based on the sum of the present cost and the expected future cost,

assuming optimal decision making for subsequent stages. This manipulation of the value function lets us collapse the augmented state space $\tilde{\mathcal{S}}$ to the much smaller space $\mathcal{S}$. We prove the convergence of the discounted cost algorithm in this framework and show how an optimal W-transition control policy can be found using standard dynamic programming algorithms.

Assume $\mathbf{P}$ is the transition probability matrix of the uncontrolled Markov chain. For $i, j \in \mathcal{S}$, let $p_{i,j}^{(r)}$ be the probability of going from state $i$ to state $j$ in $r$ steps, i.e. the $(i, j)$th entry of the matrix $\mathbf{P}^{(r)}$. The objective is to compute the recursive relation of the value function starting at time $t = sW$, given the cost value at time $t = (s + 1)W$.

Without loss of generality, we assume $s = 0$. In the augmented state space $\tilde{\mathcal{S}}$, we are not allowed to apply any control at state $(i, W - 1)$, hence from (5.1) and (5.5)

$$\begin{aligned} J_{W-1}(i, W-1) &= \min_{u \in U(i,W-1)} \left\{ \overline{g}(i,u) + \lambda \sum_{j=0}^{N-1} p_{i,j}(u) J_W(j,0) \right\} \\ &= \quad \overline{g}(i,0) + \lambda \sum_{j=0}^{N-1} p_{i,j} J_W(j,0). \end{aligned} \tag{5.6}$$

Given $J_{W-1}$, one can compute $J_{W-2}$ as

$$\begin{aligned} J_{W-2}(i, W-2) &= \min_{u \in U(i,W-2)} \left\{ \overline{g}(i,u) + \lambda \sum_{j=0}^{N-1} p_{i,j}(u) J_{W-1}(j, W-1) \right\} \\ &= \overline{g}(i,0) + \lambda \sum_{j=0}^{N-1} p_{i,j} J_{W-1}(j, W-1). \end{aligned}$$

Replacing $J_{W-1}$ from (5.6), we have $J_{W-2}(i, W-2)$ as a function of $J_W(k, 0)$ for all $k \in \mathcal{S}$,

$$J_{W-2}(i,W-2) = \overline{g}(i,0) + \lambda \sum_{j=0}^{N-1} p_{i,j} \left( \overline{g}(j,0) + \lambda \sum_{k=0}^{N-1} p_{j,k} J_W(k,0) \right)$$
$$= \overline{g}(i,0) + \lambda \sum_{j=0}^{N-1} p_{i,j} \overline{g}(j,0) + \lambda^2 \sum_{k=0}^{N-1} p_{i,k}^{(2)} J_W(k,0).$$

Similarly we can compute $J_{W-3}$ as

$$J_{W-3}(i,W-3) = \sum_{j=0}^{N-1} \left( p_{i,j}^{(0)} + \lambda p_{i,j}^{(1)} + \lambda^2 p_{i,j}^{(2)} \right) \overline{g}(j,0) + \lambda^3 \sum_{k=0}^{N-1} p_{i,k}^{(3)} J_W(k,0).$$

One can recursively evaluate the value function for the last state in a treatment window where no control is allowed, i.e. $J_1(i,1)$, as follow:

$$J_1(i,1) = \sum_{j=0}^{N-1} \left( \sum_{r=0}^{W-2} \lambda^r p_{i,j}^{(r)} \right) \overline{g}(j,0) + \lambda^{W-1} \sum_{k=0}^{N-1} p_{i,k}^{(W-1)} J_W(k,0). \tag{5.7}$$

Finally, at time $0$, intervention is allowed and the following minimization problem leads to

$$J_0(i,0) = \min_{u \in U(i,0)} \left\{ \overline{g}(i,u) + \lambda \sum_{j=0}^{N-1} p_{i,j}(u) J_1(j,1) \right\}. \tag{5.8}$$

Using (5.7) and (5.8), we obtain

$$J_0(i,0) = \min_{u \in U(i,0)} \left\{ \overline{g}(i,u) + \lambda \sum_{j=0}^{N-1} p_{i,j}(u) \left( \sum_{k=0}^{N-1} \left( \sum_{r=0}^{W-2} \lambda^r p_{j,k}^{(r)} \right) \overline{g}(k,0) \right.\right.$$
$$\left.\left. + \lambda^{W-1} \sum_{k=0}^{N-1} p_{j,k}^{(W-1)} J_W(k,0) \right) \right\}. \tag{5.9}$$

We can rewrite (5.9) as

$$J_{sW}(i,0) = \min_{u \in U(i,0)} \{ \overline{g}(i,u) + \lambda \sum_{j=0}^{N-1} p_{i,j}(u) \left( \sum_{k=0}^{N-1} \left( \sum_{r=0}^{W-2} \lambda^r p_{j,k}^{(r)} \right) \overline{g}(k,0) \right.$$
$$\left. + \lambda^{W-1} \sum_{k=0}^{N-1} p_{j,k}^{(W-1)} J_{(s+1)W}(k,0) \right) \Bigg\}, \tag{5.10}$$

for $s = 0, 1, 2, \ldots$ This equation reveals how the cost at the beginning of the $(s+1)$th window affects the cost at the beginning of the $s$th window. This equation ranks decisions based on the sum of the present cost and the expected future cost considering the cost of all the states where no control is allowed between two treatment times. This manipulation of the value function lets us collapse the state space from $\tilde{\mathcal{S}}$ to $\mathcal{S}$ and leads to

$$J_s(i) = \min_{u \in U(i)} \left\{ \overline{g}(i,u) + \lambda \sum_{j=0}^{N-1} p_{i,j}(u) \left( \sum_{k=0}^{N-1} \left( \sum_{r=0}^{W-2} \lambda^r p_{j,k}^{(r)} \right) \overline{g}(k,0) + \lambda^{W-1} \sum_{k=0}^{N-1} p_{j,k}^{(W-1)} J_{(s+1)}(k) \right) \right\}, \tag{5.11}$$

for $s = 0, 1, 2, \ldots$ It should be noted that the above backward propagation of costs applies to every $W$ transitions of the Markov chain in which we are permitted to apply intervention and $U(i) = \{0, 1\}$.

Similar to Chapter II, the following proposition discusses how an optimal W-transition stationary control policy can be devised. In Proposition 6, we prove the convergence of the discounted cost algorithm as it is defined in this work. Propositions 2 and 3 can be restated for the following operator $T$.

**Proposition 6** *(Convergence of the discounted cost algorithm): For any $i \in \mathcal{S}$, bounded function $J : \mathcal{S} \mapsto \Re$, and $T : \mathcal{S} \mapsto \Re$, where*

$$TJ(i) = \min_{u \in U(i)} \left\{ \overline{g}(i,u) + \lambda \sum_{j=0}^{N-1} p_{i,j}(u) \left( \sum_{k=0}^{N-1} \left( \sum_{r=0}^{W-2} \lambda^r p_{j,k}^{(r)} \right) \overline{g}(k,0) + \lambda^{W-1} \sum_{k=0}^{N-1} p_{j,k}^{(W-1)} J(k) \right) \right\}, \tag{5.12}$$

*the optimal cost function satisfies*

$$J^*(x) = \lim_{j' \longrightarrow \infty} \left( T^M J \right)(x), \qquad \forall\, x \in \mathcal{S}.$$

**Proof**

We define $\tilde{g}(i,u)$ as

$$\tilde{g}(i,u) = \overline{g}(i,u) + \lambda \sum_{j=0}^{N-1} p_{i,j}(u) \left( \sum_{k=0}^{N-1} \left( \sum_{r=0}^{W-2} \lambda^r p_{j,k}^{(r)} \right) \overline{g}(k,0) \right). \tag{5.13}$$

The function $\tilde{g}(i,u)$ collects the average cost of state $i$ with control $u$ and the accumulated cost of $W$ transitions from state $i$. For every positive integer $K$, initial state $x_0 \in \mathcal{S}$, and policy $\pi = \{\mu_0, \mu_1, \dots\}$, we break down the cost $J_\pi(x_0)$ into the portions incurred over the first $K$ stages and over the remaining stages

$$\begin{aligned} J_\pi(x_0) &= \lim_{M \longrightarrow \infty} E \left\{ \sum_{k=0}^{M-1} \lambda^{kW} \ \tilde{g}(x_k, \mu_k(x_k)) \right\} \\ &= E \left\{ \sum_{k=0}^{K-1} \lambda^{kW} \ \tilde{g}(x_k, \mu_k(x_k)) \right\} \\ &\quad + \lim_{M \longrightarrow \infty} E \left\{ \sum_{k=K}^{M-1} \lambda^{kW} \ \tilde{g}(x_k, \mu_k(x_k)) \right\}. \end{aligned}$$

Since the cost-per-stage is bounded, it is straightforward to see that $\tilde{g}(i,u)$ in (5.13) is bounded. Let us assume $|\tilde{g}(i,u)| < L$. We also obtain

$$\left| \lim_{M \longrightarrow \infty} E \left\{ \sum_{k=K}^{M-1} \lambda^{kW} \ \tilde{g}(x_k, \mu_k(x_k)) \right\} \right| \leq L \sum_{k=K}^{\infty} \lambda^{kW} = \frac{\lambda^{WK}}{1-\lambda^W} L.$$

It follows that

$$J_\pi(x_0) \leq E \left\{ \sum_{k=0}^{K-1} \lambda^{kW} \ \tilde{g}(x_k, \mu_k(x_k)) \right\} + \frac{\lambda^{WK}}{1-\lambda^W} L.$$

From the inequalities above, one can conclude that

$$
\begin{aligned}
& E\left\{\sum_{k=0}^{K-1} \lambda^{kW}\ \tilde{g}(x_k, \mu_k(x_k)) + \lambda^{WK} J(x_K)\right\} \\
&\geq\ E\left\{\left|\sum_{k=0}^{K-1} \lambda^{kW}\ \tilde{g}(x_k, \mu_k(x_k))\right| - \left|\lambda^{WK} J(x_K)\right|\right\} \\
&\geq\ J_\pi(x_0) - \frac{\lambda^{WK}}{1-\lambda^W} L - \lambda^{WK} \max_{x\in\mathcal{S}} |J(x)|.
\end{aligned}
$$

Similarly,

$$
\begin{aligned}
& E\left\{\sum_{k=0}^{K-1} \lambda^{kW}\ \tilde{g}(x_k, \mu_k(x_k)) + \lambda^{WK} J(x_K)\right\} \\
&\leq\ E\left\{\left|\sum_{k=0}^{K-1} \lambda^{kW}\ \tilde{g}(x_k, \mu_k(x_k))\right| + \left|\lambda^{WK} J(x_K)\right|\right\} \\
&\leq\ J_\pi(x_0) + \frac{\lambda^{WK}}{1-\lambda^W} L + \lambda^{WK} \max_{x\in\mathcal{S}} |J(x)|.
\end{aligned}
$$

Hence,

$$
\begin{aligned}
& J_\pi(x_0) - \frac{\lambda^{WK}}{1-\lambda^W} L - \lambda^{WK} \max_{x\in\mathcal{S}} |J(x)| \\
&\leq\ E\left\{\sum_{k=0}^{K-1} \lambda^{kW}\ \tilde{g}(x_k, \mu_k(x_k)) + \lambda^{WK} J(x_K)\right\} \\
&\leq\ J_\pi(x_0) + \frac{\lambda^{WK}}{1-\lambda^W} L + \lambda^{WK} \max_{x\in\mathcal{S}} |J(x)|.
\end{aligned}
$$

We need to show that $E\left\{\sum_{k=0}^{K-1} \lambda^{kW}\ \tilde{g}(x_k, \mu_k(x_k)) + \lambda^{WK} J(x_K)\right\}$ is equal to $(T^K J)(x_0)$. From the definition of $\tilde{g}(i,u)$ in (5.13) and (5.12), we have

$$TJ(i) = \min_{u \in U(i)} \left\{ \tilde{g}(i,u) + \lambda^W \sum_{j=0}^{N-1} p_{i,j}(u) \left( \sum_{k=0}^{N-1} p_{j,k}^{(W-1)} J(k) \right) \right\}$$
$$= \min_{u \in U(i)} \left\{ \tilde{g}(i,u) + \lambda^W \sum_{k=0}^{N-1} \left( \sum_{j=0}^{N-1} p_{i,j}(u) p_{j,k}^{(W-1)} J(k) \right) \right\}.$$

We denote $\sum_{k=0}^{N-1} \sum_{j=0}^{N-1} p_{i,j}(\mu) p_{j,k}^{(W-1)}$ by $\sum_{k=0}^{N-1} p_{i,k}^{(W)}(\mu)$. When optimal policy $\mu_k(x_k)$ is applied, the above equation changes to

$$T_{\mu_k} J(x_k) = \tilde{g}(x_k, \mu_k(x_k)) + \lambda^W \sum_{x_{k+1}=0}^{N-1} p_{x_k,x_{k+1}}^{(W)}(\mu_k(x_k)) J(x_{k+1}).$$

Further application of $T_{\mu_{k+1}}$ leads us to

$$T_{\mu_{k+1}} T_{\mu_k} J(x_k) = \tilde{g}(x_k, \mu_k(x_k)) + \lambda^W \sum_{x_{k+1}=0}^{N-1} p_{x_k,x_{k+1}}^{(W)}(\mu_k(x_k)) \tilde{g}(x_{k+1}, \mu_{k+1}(x_{k+1}))$$
$$+ \lambda^{2W} \sum_{x_{k+1}=0}^{N-1} p_{x_k,x_{k+1}}^{(W)}(\mu_k(x_k)) \sum_{x_{k+2}=0}^{N-1} p_{x_{k+1},x_{k+2}}^{(W)}(\mu_{k+1}(x_{k+1})) J(x_{k+2}).$$

By induction, one can show that

$$T_{\mu_K} T_{\mu_{K-1}} \ldots J(x_0) = E \left\{ \sum_{k=0}^{K-1} \lambda^{kW} \ \tilde{g}(x_k, \mu_k(x_k)) + \lambda^{WK} J(x_K) \right\}.$$

By taking the minimization over $\pi$, we obtain

$$J^*(x_0) - \frac{\lambda^{WK}}{1-\lambda^W} L - \lambda^{WK} \max_{x \in \mathcal{S}} |J(x)|$$
$$\leq \ (T^K J)(x_0)$$
$$\leq \ J^*(x_0) + \frac{\lambda^{WK}}{1-\lambda^W} L + \lambda^{WK} \max_{x \in \mathcal{S}} |J(x)|,$$

for all $x_0$ and $K$. By taking the limit as $K \to \infty$, the result follows. □

Propositions 2, 3 (in Chapter II), and 6 provide the basis for computational algorithms to determine an optimal W-transition policy. Proposition 2 asserts that the optimal cost function satisfies Bellman's optimality equation, while Proposition 6 states that the optimal cost function can be iteratively determined by running the recursion

$$J_{s+1} = TJ_s, \quad s = 0, 1, 2, \ldots \tag{5.14}$$

for any bounded initial cost function $J_0 : \mathcal{S} \mapsto \Re$. An optimal W-transition policy is found when the iteration converges to the optimal value of the cost function.

## B. Results and Discussion

An optimal one-transition policy is no longer optimal, i.e. does not necessarily minimize the expected total discounted cost, if one is restricted to apply treatment only every $W$ transitions. Nevertheless, we can apply an optimal one-transition policy every $W$ transitions and compare the effect and cost of such a policy to the ones of an optimal W-transition policy, which truly minimizes the expected total discounted cost.

We anticipate an effective control policy to reduce the likelihood of visiting undesirable states compared to a network without intervention by modifying the long-run behavior of the network. The effectiveness of a control policy can be measured by the amount of change (*shift*) in the aggregated probability of undesirable states before and after intervention. We should emphasize that an optimal policy does not necessarily result in a maximum shift in the steady-state distribution, as explained above, since we are minimizing the expected total discounted cost. The amount of shift in the aggregated probability of undesirable states before and after intervention can be computed as

$$\Delta P^W = \frac{\sum_{i \in \mathcal{U}} \pi_i - \sum_{i \in \mathcal{U}} \tilde{\pi}_i^W}{\sum_{i \in \mathcal{U}} \pi_i}. \tag{5.15}$$

In (5.15), $\tilde{\pi}_i^W$ is the probability of being in undesirable state $i$ in the long run using a policy that is applied every $W$ transitions. In this equation, $\pi_i$ is the probability of being in undesirable state $i$ in the long run when there is no intervention. In other words, given a Markovian gene regulatory network, one can shift the aggregated probability of undesirable states to desirable ones through appropriately altering the expression of the control gene every $W$ time instants.

Formulation of $\Delta P^W$ requires the computation of $\tilde{\pi}^W$, i.e. the steady-state distribution of the Markov chain under a W-transition policy $\mu_W$, a policy that is applied every $W$ transitions. To this end, we derive the transition probability matrix of the system when a W-transition policy $\mu_W$ is applied. In general, $W$ possible cases can happen for the transition of state $i$ to state $j$ in $W$ steps under a cyclic policy depending on the instants in which states $i$ and $j$ are observed with respect to the treatment times. Let us denote the transition probability matrix under the W-transition policy $\mu_W$ by $\mathbf{P}_{\mu_W}$. In the first case, there are $W-1$ uncontrolled transitions and the corresponding transition probability matrix is $\mathbf{P}^{W-1}$. Afterward, in $W$th transition, policy $\mu_W$ decides whether to apply control or not. The system transitions to state $j$ and the corresponding transition probability matrix is $\mathbf{P}_{\mu_W}$. Consequently, the transition probability matrix corresponding to the first case is $\mathbf{P}^{(W-1)}\mathbf{P}_{\mu_W}$. In the second case, starting from state $i$, there are $W-2$ uncontrolled transitions and the corresponding transition probability matrix is $\mathbf{P}^{W-2}$. At next transition, policy $\mu_W$ decides whether to apply control or not and the system transitions according to the transition probability matrix $\mathbf{P}_{\mu_W}$. Thereafter, the system transitions to state $j$ according to the original transition probability matrix $\mathbf{P}$. The transition probability matrix corresponding to the second case is $\mathbf{P}^{(W-2)}\mathbf{P}_{\mu_W}\mathbf{P}$. Likewise, the transition probability matrix for $W-2$ other cases can be derived. Fig. 11 demonstrates an example for $W=4$. As this figure suggests, 4 possible cases can happen depending on when state $i$ is observed with respect to treatment times.

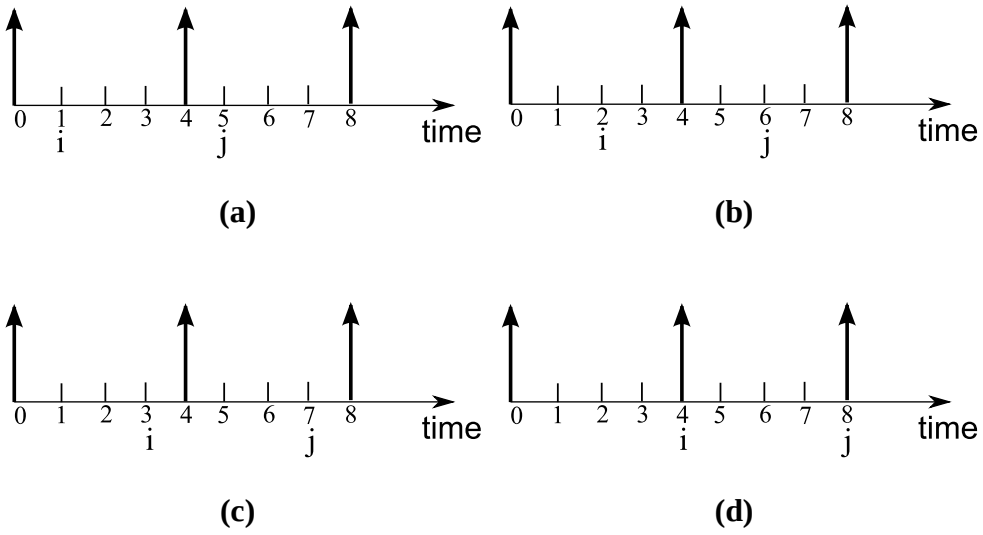



Fig. 11. An example of cyclic intervention strategy for $W = 4$. Arrows represent treatment times. Subfigures a to d show the four possible cases that can happen depending on the instants in which states $i$ and $j$ are observed with respect to treatment times.

To find the transition probability matrix of the Markov chain under optimal W-transition policy, one should consider the possibility of these cases. Since each of these cases are equally probable, the following transition probability matrix represents the probabilities of transitions among states when the W-transition intervention policy $\mu_W$ is applied

$$\hat{\mathbf{P}}_{\mu_W} = \frac{1}{W} \sum_{w=1}^{W} \mathbf{P}^{(W-w)} \mathbf{P}_{\mu_W} \mathbf{P}^{(w-1)}. \tag{5.16}$$

The steady-state distribution $\tilde{\pi}^W$ is the invariant distribution of $\hat{\mathbf{P}}_{\mu_W}$.

In the following sections, we first derive optimal one-transition and W-transition policies for synthetic networks. We generate random PBNs with various properties. We vary the values of bias and connectivity of the PBNs. The bias of a PBN is the probability that each constituent Boolean function takes on the value 1 and the connectivity corresponds to the maximum number of predictors for each Boolean function. Since the bias and con-

nectivity affect the dynamical properties of randomly generated BNs [4], we take them as parameters in our simulations. Whenever not specified, the connectivity of the PBN is 3. Furthermore, we investigate the effect of the cost of control on each type of policy. We provide some of these in the sequel. We then present a similar investigation for the network obtained from the mammalian cell-cycle network explained in Chapter II.

### 1. Synthetic Networks

We generate random PBNs with 7 genes. Each PBN consists of $4$ constituent BNs. For each PBN, the probability transition matrix of the corresponding Markov chain is computed [5]. Without loss of generality, the target gene is chosen to be the most significant gene in the states. We assume that the up-regulation of the target gene is undesirable. Consequently, the state space is partitioned into desirable states, $\mathcal{D} = \{0, \ldots, N/2 - 1\}$, and undesirable states, $\mathcal{U} = \{N/2, \ldots, N - 1\}$, where $N$ represents the total number of states. Since our objective is to down-regulate the target gene, a higher cost is assigned to destination states having an up-regulated target gene. We postulate the following cost-per-stage:

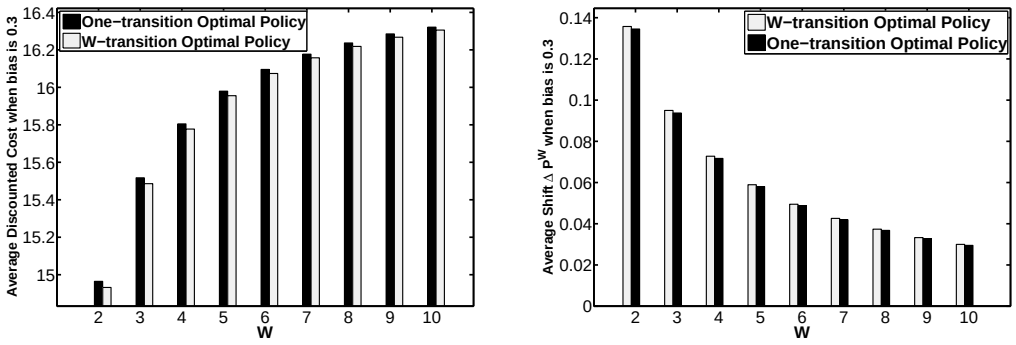
$$g(u, j) = \begin{cases} 0 & \text{if } u = 0 \text{ and } j \in \mathcal{D}, \\ 5 & \text{if } u = 0 \text{ and } j \in \mathcal{U}, \\ \text{c} & \text{if } u = 1 \text{ and } j \in \mathcal{D}, \\ 5 + \text{c} & \text{if } u = 1 \text{ and } j \in \mathcal{U}, \end{cases} \tag{5.17}$$

where $c$ represents the cost of control. Whenever it is not specified, the cost of control is selected to be zero.

For each PBN, we vary the value of $W$ from 1 to 10. For each $W$, the optimal W-transition policy is derived and the corresponding $\Delta P^W$ is computed from (5.16). Given the optimal W-transition policy, we estimate the average total discounted cost induced by this policy. To this end, we generate synthetic time-course data for $1000$ time-steps from

each PBN model while the optimal W-transition policy is applied. Using this synthetic time-course data, we estimate the discounted cost by accumulating the discounted cost of each state given the policy at that state. This procedure is repeated $10,000$ times for random initial states and the average of the induced discounted cost is computed. Furthermore, the optimal one-transition policy is applied every $W$ transitions and the corresponding $\Delta P^W$ is computed from (5.16). To compute the average discounted cost of the optimal one-transition policy when it is applied every $W$ transitions, we generate synthetic time-course data as explained above and the average total discounted cost of the optimal one-transition policy is similarly computed. In summary, for each PBN model, we have the following: ($\bar{C}^W$) average total discounted cost resulting from the optimal W-transition policy; ($\Delta P^W$) the value of $\Delta P^W$ resulting from the optimal W-transition policy; ($\bar{C}^{W,1}$) the average total discounted cost of the optimal one-transition policy when it is applied every $W$ transitions; and ($\Delta P^{W,1}$) the value of $\Delta P^W$ resulting from the optimal one-transition policy applied every $W$ transitions. The preceding procedure is repeated for $1000$ random PBNs, thereby yielding 1000 values for each statistic: $\bar{C}_1^W, \ldots, \bar{C}_{1000}^W$;$\Delta P_1^W, \ldots, \Delta P_{1000}^W$;$\bar{C}_1^{W,1}, \ldots, \bar{C}_{1000}^{W,1}$; $\Delta P_1^{W,1}, \ldots, \Delta P_{1000}^{W,1}$. Using these, we compare the optimal W-transition and one-transition polices via the empirical averages $\mathbf{M}[C^W]$ of $\bar{C}_1^W, \ldots, \bar{C}_{1000}^W$; $\mathbf{M}[C^{W,1}]$ of $\bar{C}_1^{W,1}, \ldots, \bar{C}_{1000}^{W,1}$; $\mathbf{M}[\Delta P^W]$ of $\Delta P_1^W, \ldots, \Delta P_{1000}^W$; and $\mathbf{M}[\Delta P^{W,1}]$ of $\Delta P_1^{W,1}, \ldots, \Delta P_{1000}^{W,1}$. In addition, for each value of $W$, the histograms of the differences $\bar{C}_i^{W,1} - \bar{C}_i^W$ and $\Delta P_i^W - \Delta P_i^{W,1}$, $i = 1, \ldots, 1000$, are also found. We will see that the means tend to be close, $\mathbf{M}[C^W] \approx \mathbf{M}[C^{W,1}]$ and $\mathbf{M}[\Delta P^W] \approx \mathbf{M}[\Delta P^{W,1}]$, but that the histograms of the differences have long tails to the right, indicating that there are cases for which using the optimal one-transition policy can have strongly detrimental effects.

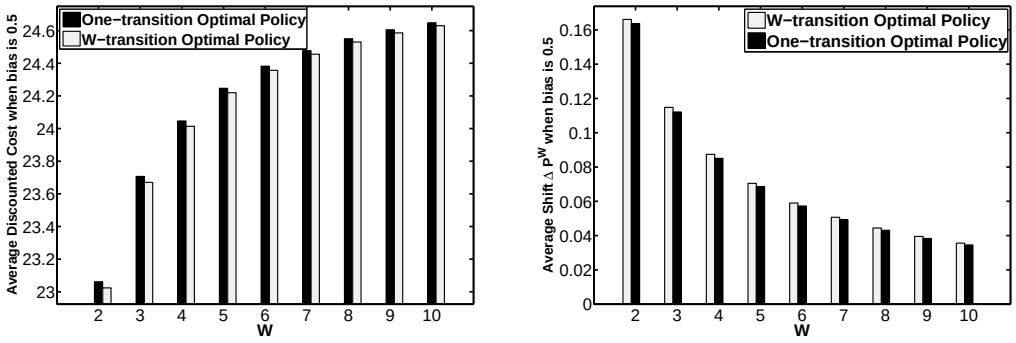
In the first set of experiments, each constituent BN is randomly generated with a bias, the bias being the probability that a Boolean function takes on the value 1. We randomly select the bias $b$ of a BN from a beta distribution. The mean of the beta distribution is




**(a)** **(b)**

Fig. 12. Comparison of optimal W-transition and one-transition policies based on the average values of $\Delta P^W$ and average total discounted cost for $W \in \{1, \ldots, 10\}$ for random PBNs with bias mean = 0.3. (a) Average of discounted cost, (b) Average of $\Delta P^W$

chosen to be 0.3, 0.5, or 0.7. The variance of the beta distribution, $\sigma^2$, is set to a constant value $0.0001$. The average values of $\Delta P^W$ and the average total discounted costs for both one-transition and W-transition policies are shown in Figs. 12 and 13 for bias values of 0.3 and 0.5, respectively. Similarly, the histograms of the differences of the two policies in terms of $\Delta P^W$ and the average total discounted costs are shown in Figs. 14 and 15.




**(a)** **(b)**

Fig. 13. Comparison of optimal W-transition and one-transition policies based on the average values of $\Delta P^W$ and average total discounted cost for $W \in \{1, \ldots, 10\}$ for random PBNs with bias mean = 0.5. (a) Average of discounted cost, (b) Average of $\Delta P^W$

We observe that the average of $\Delta P^W$ for both policies decreases as $W$ increases. This behavior is in accordance with the intuition that treatments which are further apart in time are less effective. As we stated in the Introduction, tumors given less time to grow between treatments are more likely to be eradicated [14]. In the long run, less treatment is applied for a larger $W$ and consequently more cost is induced. Hence, for a fixed bias, the average discounted costs of both one-transition and W-transition policies increase as $W$ increases. On average, the optimal W-transition policy results in lower discounted cost and higher $\Delta P^W$ compared to the optimal one-transition policy. The histograms of the differences of the two policies in terms of $\Delta P^W$ and average discounted show how often they generate similar outcomes and how often the effect of the two policies differ. Note that the differences are not positive for all PBNs. This is because the optimal policies minimize the 'expected' total discounted cost. Hence, the W-transition control policy can induce a larger average discounted cost compared to the one-transition control policy, but rarely.

In the second set of experiments, we generate constituent BNs with connectivities $2$, $3$, and $4$. For each connectivity, predictors and Boolean functions are randomly generated with a bias $b$, randomly selected from a beta distribution with mean $0.5$. Similar to the previous experiment, we observe that the optimal W-transition policy results in lower average discounted cost and higher $\Delta P^W$ compared to the optimal one-transition policy. In the third set of experiments, we repeat the simulations for the cost of control being 0, 0.1, and 0.5. Results of this experiment for control cost of 0.1 are shown in Figs. 16 and 17.

### 2. Mammalian Cell-Cycle Network

The value of $\Delta P^W$ and the average total discounted cost for both optimal one-transition and W-transition policies derived for the cell-cycle network are shown in Fig. 18. In the long run, less treatment is applied for a larger $W$ and consequently more cost is induced. Hence, the average discounted costs of both optimal one-transition and W-transition policies increase as $W$ increases. It should be noted that the previous experiments show the average behavior of 1000 random PBNs while this experiment considers the behavior of one network, i.e. the mammalian cell-cycle network. In this instance, the optimal one-transition and W-transition policies are close to parity for the cell-cycle network.

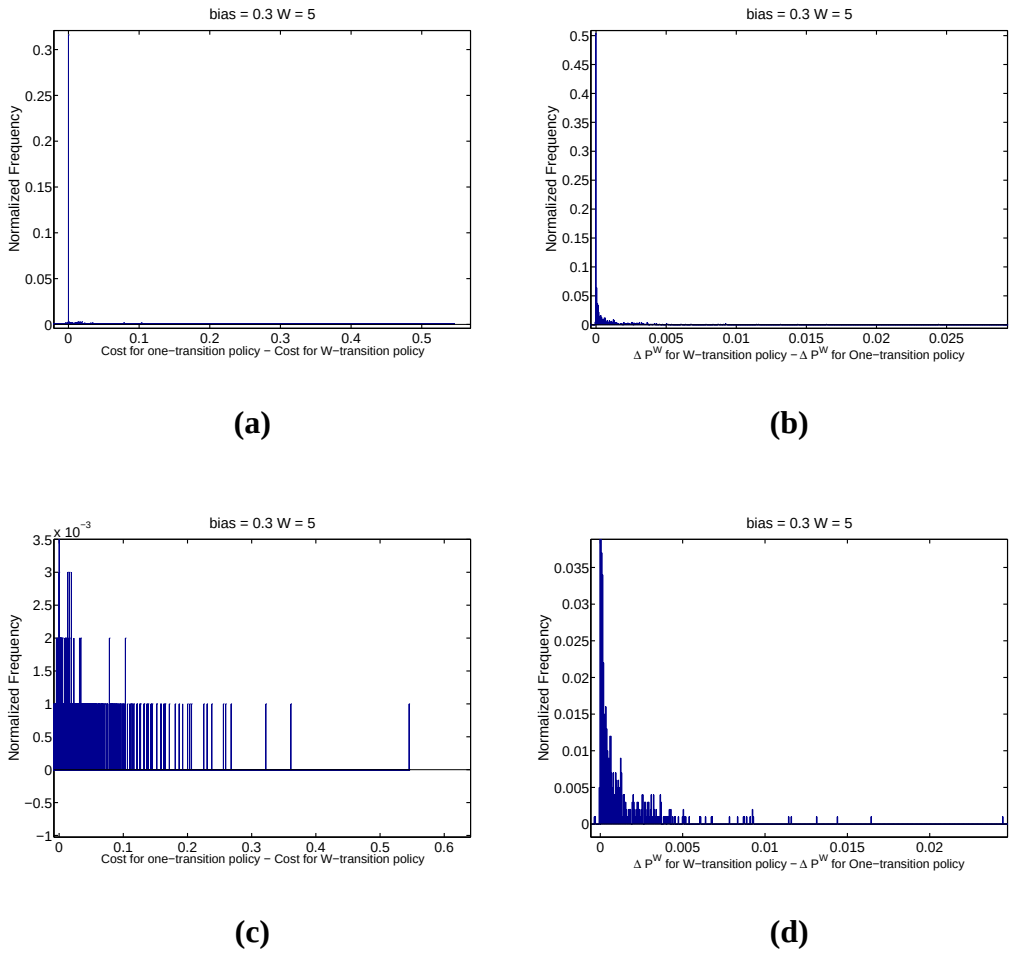



Fig. 14. Comparison of optimal W-transition and one-transition policies based on the histogram of difference of W-transition and optimal one-transition policies for $W = 5$ on random PBNs with bias mean = 0.3. (a) Histogram of $\Delta P^W$ associated to optimal W-transition policy minus $\Delta P^W$ associated to optimal one-transition policy, (b) Histogram of the average discounted cost associated to optimal one-transition policy minus the average discounted cost associated to optimal W-transition policy, (c) enlarged view of (a), (d) enlarged view of (b)

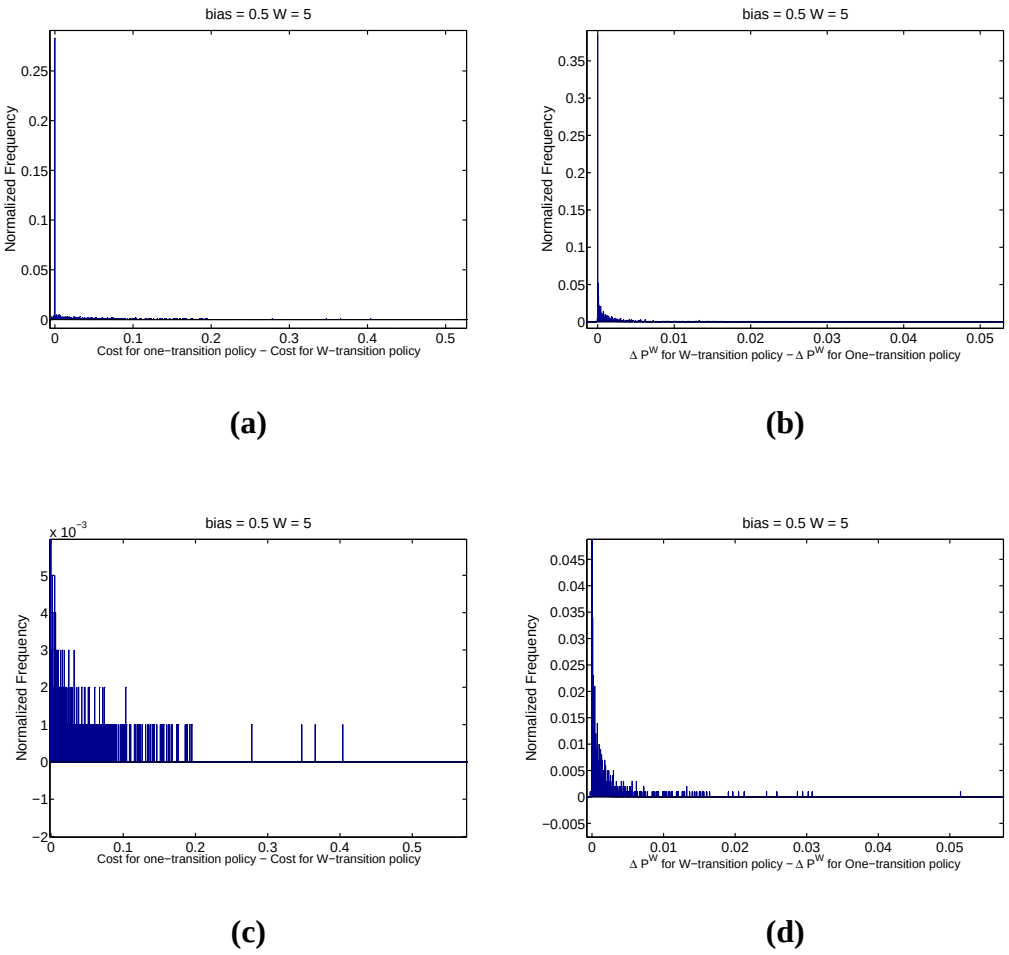
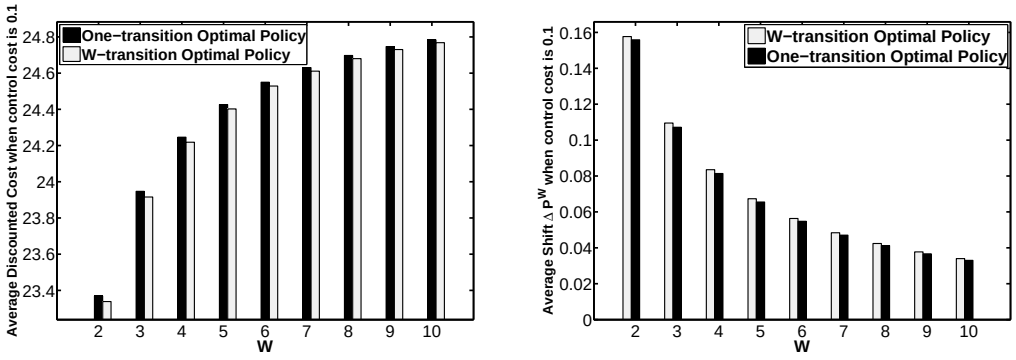



Fig. 15. Comparison of optimal W-transition and one-transition policies based on the histogram of difference of W-transition and optimal one-transition policies for $W = 5$ on random PBNs with bias mean = 0.5. (a) Histogram of $\Delta P^W$ associated to optimal W-transition policy minus $\Delta P^W$ associated to optimal one-transition policy, (b) Histogram of the average discounted cost associated to optimal one-transition policy minus the average discounted cost associated to optimal W-transition policy, (c) enlarged view of (a), (d) enlarged view of (b)




**(a)** **(b)**

Fig. 16. Comparison of optimal W-transition and one-transition policies based on the average values of $\Delta P^W$ and average total discounted cost for $W \in \{1, \ldots, 10\}$ for random PBNs with control cost = 0.1. (a) Average of $\Delta P^W$, (b) Average of discounted cost.

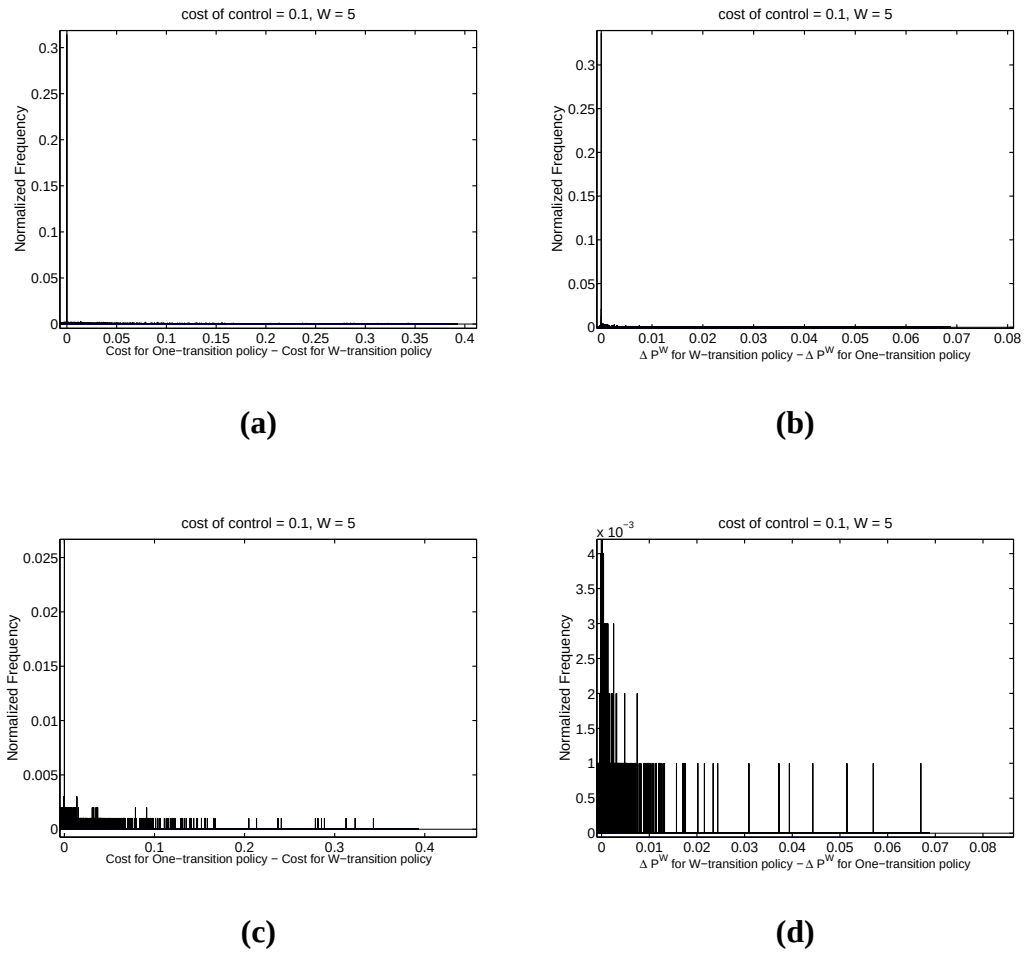



Fig. 17. Comparison of optimal W-transition and one-transition policies based on the histogram of difference of optimal W-transition and one-transition policies for $W = 5$ on random PBNs when cost of control is 0.1. (a) Histogram of $\Delta P^W$ associated to optimal W-transition policy minus $\Delta P^W$ associated to optimal one-transition policy, (b) Histogram of the average discounted cost associated to optimal one-transition policy minus the average discounted cost associated to optimal W-transition policy, (c) enlarged view of (a), (d) enlarged view of (b)

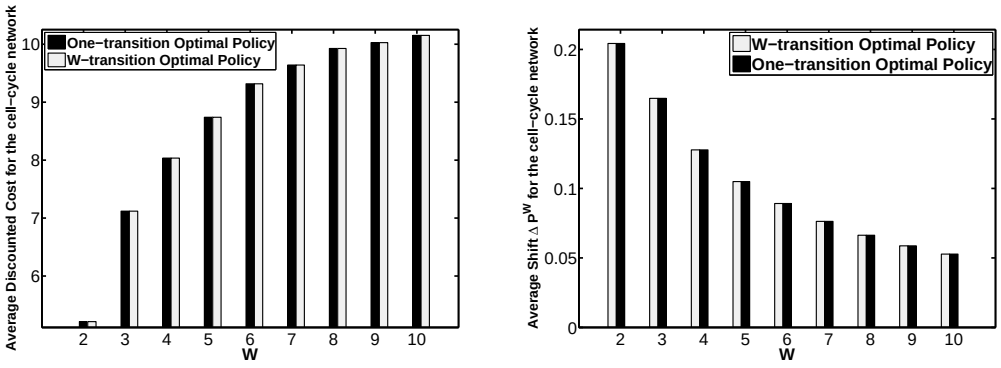



Fig. 18. Comparison of optimal W-transition and one-transition policies based on the values of $\Delta P^W$ and average total discounted cost for $W \in \{1, \ldots, 10\}$ for the mammalian cell-cycle network. (a) Average of total discounted cost, (b) $\Delta P^W$

# CHAPTER VI

## MEAN-FIRST-PASSAGE-TIME CONTROL POLICY*

The study of infinite-horizon intervention strategies poses two fundamental questions. First, is it possible to beneficially affect a network by applying the optimal stationary control policy? This translates into assessing the *controllability* of the network. In practice, a physician would like to predict the effectiveness of a certain treatment at different stages of a disease and on different patients. Investigating the effect of a certain type of control on various networks is equivalent to questioning the controllability of the network. To date, there has been no investigation on this important topic in the context of gene regulatory networks. Second, can we identify the best intervening gene? In other words, which gene is the best potential "lever point," to borrow the terminology from [32], in the sense of having the greatest possible impact on the desired network behavior? In principle, solving an optimal control problem for each candidate gene and comparing the performance of the system for these various controls would answer these questions; however, this process is a computationally demanding procedure. The complexity of dynamic programming algorithms can be vast and increases exponentially with the number of genes [40].

In their early papers, Shmulevich et al. employ two methods for selecting a candidate gene for intervention: *mean first-passage time* and *influence* [5, 6]. The following biological example, borrowed from [4], explains the intuition behind using mean first-passage times for selecting the best control gene. In biology, there are numerous cases where the (in)activation of a certain gene or protein can lead more quickly (or with higher probability) to a particular cellular functional state or phenotype than the (in)activation of another gene

* Reprinted with permission from " Intervention in Gene Regulatory Networks via a Stationary Mean-First-Passage-Time Control Policy" by G. Vahedi, B. Faryabi, J.-F. Chamberland, A. Datta, and E. R. Dougherty, 2008, *IEEE Transactions on Biomedical Engineering*, 55, 2319-2331, Copyright 2008 by IEEE.

or protein. For instance, in a stable cancer cell line, in the absence of intervention, the cells will keep proliferating. This behavior can be reversed by controlling the expression of certain genes. Assume that the goal of the intervention is to push the cell into programmed cell death (apoptosis). Further assume that we can achieve this intervention with two candidate genes: p53 and telomerase. The p53 gene is the most well-known tumor suppressor gene [41][42][43]. The telomerase gene encodes telomerase, which maintains the integrity of the end of chromosomes (telomeres) in germ cells. Germ cells are responsible for propagating the complete genetic material to the following generation. Telomerase also maintains the integrity of the end of chromosomes in progenitor cells. Progenitor cells are responsible for replenishing cells during the normal cell turnover (homeostasis). In somatic cells, the telomerase gene is turned off, resulting in telomere shortening each time the cell divides – a key reason for the limited life-span of normal cells [44]. In the majority of tumor cells, telomerase is activated, which is believed to contribute to the prolonged life-span of the tumor cells [45]. This worsens prognosis for cancer patients [46][47]. Extensive experimental results indicate that when p53 is activated in the cells, for example in response to radiation, the cells undergo rapid growth inhibition and apoptosis in as short as a few hours [48]. In contrast, inhibition of the telomerase gene also leads to cell growth inhibition, differentiation, and cell death, but only after cells go through a number of cell divisions (allowing telomere shortening). Cell death takes a longer time through this latter process than via p53 activation. The use of mean first-passage times for finding the best control gene is intuitive; however, it focuses on a one-step control scenario.

The influence method distinguishes genes that have a major impact on a predictor function from those that have only a minor impact. This method was introduced to reflect the extent to which a set of genes is capable of determining the value of a target gene [5]. It has been used as a criterion to select a control gene with the suggestion that the gene with the largest influence on the target gene is likely to be a good control gene in the finite

and infinite-horizon control of PBNs [49][11]; however, no research has been done on the overall performance of this heuristic measure.

Capitalizing on the biological intuition behind mean first-passage time, we propose an algorithm based on mean first-passage times that assigns a stationary control policy for each candidate gene. We call this algorithm the *Mean First-Passage Time (MFPT)* algorithm and refer to the corresponding stationary control policy as the *Mean First-Passage Time (MFPT)* control policy. The proposed algorithm selects the MFPT control policy based on two heuristics: (1) it is preferable to *reach* desirable states as early as possible; (2) it is preferable to *leave* undesirable states as early as possible. The MFPT algorithm can be employed in four main applications.

First, the MFPT algorithm can be used for predicting the best control gene. The MFPT algorithm enables the computation of the MFPT control policies for all the genes in the network with a manageable complexity. The control gene with the highest desirable effect on the long-run behavior of the network upon the application of the corresponding MFPT control policy is likely the most effective gene for controlling the biological system. Second, to reduce the complexity of the optimal stochastic control problem, the MFPT control policy can be used as an approximate solution. Contrary to optimal algorithms, the MFPT algorithm finds policies with constant complexity. Third, the MFPT algorithm can be used to measure the controllability of a network. Since the MFPT control policy is an approximation for the optimal control policy, one can define a network to be controllable if the effect of the MFPT control policy is greater than a desired threshold.

Finally, the MFPT algorithm can be used to design a control policy without requiring network inference. The optimal stochastic control polices proposed thus far require perfect knowledge of the probability transition matrix governing the network, which must be derived from the PBN or inferred directly. These procedures are prone to modeling errors and suffer from problems of computational complexity for both network inference and finding

the optimal control solutions. To achieve model-free intervention, the MFPT control policy can be designed based on estimates of the mean first-passage times. The model-free intervention method has low complexity, is robust to modeling errors, and adapts to changes in the underlying biological system.

## A. Mean-First-Passage-Time Algorithm

In this section, we first elaborate on how the MFPT algorithm is designed based on the mean first-passage time. We then summarize the MFPT algorithm. Application of the MFPT algorithm requires the designation of desirable and undesirable states, and this depends upon the existence of relevant biological knowledge. Intervention is performed by flipping (toggling) the expression status of a particular gene from ON to OFF or vice-versa, the intent being to externally guide the time evolution of the network towards more desirable states. If $g$ is the control gene, then applying the control (intervention) in state $\mathbf{x}$ translates into flipping the value of $g$ in that state (the control gene $g$ changes to $0$ if its value is $1$ and vice-versa). Consequently, the network resumes its transition from the new state $\tilde{\mathbf{x}}$, which we call the *flipped-state*. In the context of therapy, the state-space of a PBN can be partitioned into desirable and undesirable states. Given a control gene, when a desirable state reaches the set of undesirable states on average faster than its flipped-state, it is reasonable to intervene and transition into the flipped-state. Similarly, if an undesirable state reaches the set of desirable states on average faster than its flipped-state, it is reasonable not to intervene. These insights motivate the use of mean first-passage times for designing intervention strategies.

Without loss of generality we can assume that the transition probability matrix **P** of the Markov chain (representing a PBN) is partitioned according to

$$\mathbf{P} = \begin{pmatrix} P_{\mathcal{D},\mathcal{D}} & P_{\mathcal{D},\mathcal{U}} \\ P_{\mathcal{U},\mathcal{D}} & P_{\mathcal{U},\mathcal{U}} \end{pmatrix},$$

where $\mathcal{D}$ and $\mathcal{U}$ are the subsets of desirable and undesirable states, respectively. The mean first-passage times are computed by solving the following systems of linear equations [19]:

$$K_{\mathcal{U}} = e + P_{\mathcal{D},\mathcal{D}} \cdot K_{\mathcal{U}}, \tag{6.1}$$

$$K_{\mathcal{D}} = e + P_{\mathcal{U},\mathcal{U}} \cdot K_{\mathcal{D}}, \tag{6.2}$$

where $e$ is a column vector of 1's with appropriate length, $K_{\mathcal{U}}$ is a vector containing the mean first-passage times from each state in the subset of desirable states $\mathcal{D}$ to undesirable states in set $\mathcal{U}$, and $K_{\mathcal{D}}$ is a vector containing the mean first-passage times from each state in the subset of undesirable states $\mathcal{U}$ to the desirable states in set $\mathcal{D}$.

A control policy $\mu_g$ corresponding to control gene $g$ is a vector of size $2^n$, the number of states in the network. The decision rule $\mu_g : \mathcal{S} \rightarrow \mathcal{C}$ specifies the control action for each state $\mathbf{x}$ in $\mathcal{S}$. Having $\mu_g(\mathbf{x}) = 0$ for state $\mathbf{x}$ means that, whenever the network reaches state $\mathbf{x}$, no control is applied and the system continues its transition based on the transition probabilities of state $\mathbf{x}$. On the other hand, having $\mu_g(\mathbf{x}) = 1$ implies that, whenever the network reaches state $\mathbf{x}$, the control is applied and the system continues its evolution based on the transition probabilities of state $\tilde{\mathbf{x}}$, the flipped-state of $\mathbf{x}$.

The goal of the MFPT algorithm is to design the MFPT control policies $\{\hat{\mu}_g\}_{g=1}^{n}$. The objective is to choose a control value $u$ for every state in $\mathcal{S}$ such that the network evolves towards more desirable states. The MFPT algorithm selects the control policy for control gene $g$ in the following manner. Assume state $\mathbf{x}$ is an undesirable state. We compare the mean first-passage times from state $\mathbf{x}$ to $\mathcal{D}$ and from the flipped-state $\tilde{\mathbf{x}}$ to $\mathcal{D}$. In other words, we would like to know on average which one of these two states, $\mathbf{x}$ and $\tilde{\mathbf{x}}$, *hits* the set of desirable states for the first time faster than the other one. The algorithm

chooses $\hat{\mu}_g(\mathbf{x}) = 1$ if the difference between the mean first-passage times of state $\mathbf{x}$ and the flipped-state $\tilde{\mathbf{x}}$ to the set of desirable states, i.e. $K_{\mathcal{D}}(\mathbf{x}) - K_{\mathcal{D}}(\tilde{\mathbf{x}})$, is greater than a tuning parameter $\gamma$ (to be discussed). Otherwise, $\hat{\mu}_g(\mathbf{x}) = 0$. Analogously, if state $\mathbf{x}$ is desirable, then $\hat{\mu}_g(\mathbf{x}) = 1$ if the difference between the mean first-passage times of state $\mathbf{x}$ and the flipped-state $\tilde{\mathbf{x}}$ to undesirable states, i.e. $K_{\mathcal{U}}(\tilde{\mathbf{x}}) - K_{\mathcal{U}}(\mathbf{x})$, is greater than $\gamma$. Otherwise, $\hat{\mu}_g(\mathbf{x}) = 0$. These comparisons are repeated for all states. Algorithm 1 summarizes the proposed procedure.

The threshold $\gamma$ in the MFPT algorithm is a tuning parameter chosen based on the ratio of the cost of control to the cost of undesirable states. When the cost of applying treatment in a state is high compared to the cost of undesirable states, an optimal control policy is less likely to apply the control frequently. Thus, $\gamma$ is set to a larger value when this ratio is higher, the intent being to apply control less frequently. We explain after the following definitions how one can set this parameter.

An effective control policy reduces the likelihood of visiting undesirable states compared to a network without intervention by modifying the long-run behavior of the network. The effectiveness of a control policy can be measured by the amount of change (*shift*) in the aggregated probability of undesirable states before and after intervention. As a performance measure we define

$$\Delta P_g = \frac{\sum_{i \in \mathcal{U}} \pi_i - \sum_{i \in \mathcal{U}} \pi_i^g}{\sum_{i \in \mathcal{U}} \pi_i},$$

where $\pi_i^g$ is the probability of being in undesirable state $i$ in the long-run after intervening with control gene $g$, and $\pi_i$ is the probability of being in undesirable state $i$ in the long-run when there is no intervention. The ratio $\Delta P_g$ measures the proportion of reduction in the total probability of undesirable states in the steady state when the control gene $g$ is selected. We denote this proportion by $\Delta P_g^{\text{opt}}$ when an optimal control policy $\mu_g^*$ is applied. In other words, in the optimal case one can shift the aggregated probability of undesirable states to

**Algorithm 1** MFPT algorithm

Partition the state-space into undesirable $\mathcal{U}$ and desirable $\mathcal{D}$ subsets.

Compute $K_{\mathcal{U}}$ and $K_{\mathcal{D}}$.

$g \leftarrow 1$.

**repeat**

**for** All states $\mathbf{x}$ in $\mathcal{U}$ **do**

$\tilde{\mathbf{x}} \leftarrow$ flip control gene $g$ in $\mathbf{x}$.

**if** $K_{\mathcal{D}}(\mathbf{x}) - K_{\mathcal{D}}(\tilde{\mathbf{x}}) > \gamma$ **then**

$\mu_{\mathbf{g}}(\mathbf{x}) = 1$;

**else**

$\mu_{\mathbf{g}}(\mathbf{x}) = 0$;

**end if**

**end for**

**for** All states $\mathbf{x}$ in $\mathcal{D}$ **do**

$\tilde{\mathbf{x}} \leftarrow$ flip control gene $g$ in $\mathbf{x}$.

**if** $K_{\mathcal{U}}(\tilde{\mathbf{x}}) - K_{\mathcal{U}}(\mathbf{x}) > \gamma$ **then**

$\mu_{\mathbf{g}}(\mathbf{x}) = 1$;

**else**

$\mu_{\mathbf{g}}(\mathbf{x}) = 0$;

**end if**

**end for**

$g \leftarrow g + 1$.

**until** $g >$ number of genes

desirable states by $\Delta P_g^{\text{opt}}$ through appropriately altering the expression of the control gene $g$. Similarly, the shift obtained by the MFPT control policy $\hat{\mu}_g^{\gamma}$ is denoted by $\Delta P_g^{\text{MFPT}(\gamma)}$, where $\gamma$ is the tuning parameter.

We define *the probability of the execution of control* as

$$\Gamma_g = \sum_{j=0}^{2^n-1} \pi_j \cdot \mathbf{1}(\mu_g(j) = 1), \tag{6.3}$$

where $n$ is the number of genes, $\pi_j$ is the steady-state probability of state $j \in \mathcal{S}$, $\mu_g(j)$ is the value of the control policy in state $j$, and $\mathbf{1}(\cdot)$ is the indicator function. The purpose of introducing this probability is to have a fair evaluation of the performance of the MFPT control policy in terms of the number of control executions, which for the optimal policy is related to the cost of control. For each control gene $g$, one can define $\Gamma_g^{\text{opt}}$ as the probability of the execution of control when the optimal control policy is applied. Similarly, $\Gamma_g^{\text{MFPT}(\gamma)}$ is the probability of the execution of control when the MFPT control policy with the parameter $\gamma$ is applied.

We numerically find the value of the parameter $\gamma$ for each control cost. We generate random intervention problems and calculate the averages of $\Gamma_g^{\text{opt}}$ and $\Gamma_g^{\text{MFPT}(\gamma)}$. These averages are taken over random intervention problems with fixed control cost. Starting from $\gamma = 0$, we increase the value of $\gamma$. For each control cost, the desired value of $\gamma$ is the minimal one for which, on average, $\Gamma_g^{\text{opt}} > \Gamma_g^{\text{MFPT}(\gamma)}$. This condition guarantees that on average the MFPT control policy applies no more control than the optimal control policy. Since the values of the parameter $\gamma$ are found from random intervention problems, in practice one can have a conservative approach and choose the parameter $\gamma$ to be greater than the proposed value. The conservative approach can assure a high probability that $\Gamma_g^{\text{opt}} > \Gamma_g^{\text{MFPT}(\gamma)}$. On the other hand, the deviation of $\Delta P_g^{\text{MFPT}(\gamma)}$ from $\Delta P_g^{\text{opt}}$ becomes larger.

## B. Applications of the MFPT Algorithm

We devise solutions according to the MFPT algorithm for four intervention applications.

### 1. Identification of the Best Control Gene

Recalling the example of p53 and telomerase in the introduction, it is important to select the most effective control gene in a therapeutical intervention. The best control gene $g^*$ can be found by directly solving a dynamic programming algorithm and computing $\{\Delta P_g^{\text{opt}}\}_{g=1}^n$ for all the genes $g$ in the network. In short, $g^*$ is given by

$$g^* = \arg \max_{g=1\ldots n} \Delta P_g^{\text{opt}}; \tag{6.4}$$

however, this optimal method to find the best control gene is computationally prohibitive. On the other hand, the MFPT algorithm enables the computation of the MFPT control policies $\{\hat{\mu}_g^\gamma\}_{g=1}^n$ for all the genes in the network with an acceptable complexity. Taking this approach, the MFPT algorithm predicts the best control gene to be

$$\hat{g} = \arg \max_{g=1\ldots n} \Delta P_g^{\text{MFPT}(\gamma)}. \tag{6.5}$$

We will show that $\hat{g} = g^*$ with high probability, and that $\Delta P_{g^*}^{\text{opt}} - \Delta P_{\hat{g}}^{\text{opt}}$ is small whenever $\hat{g} \neq g^*$. In this context, we are using the MFPT algorithm to find the control gene. Once the best gene candidate is identified, an optimal control policy can be obtained using dynamic programming algorithms. We will also show that the MFPT-based prediction of the best control gene significantly outperforms the influence method.

### 2. Approximation of the Optimal Control Policy

The MFPT algorithm can devise an intervention strategy as an approximation of the optimal intervention strategy. To this end, we numerically find the value of the parameter $\gamma$ for each control cost so that, on average, $\Gamma_{g^*}^{\text{opt}} > \Gamma_{g^*}^{\text{MFPT}}$. To assess the accuracy of the approximation, we show that the average of $\Delta P_{g^*}^{\text{opt}} - \Delta P_{g^*}^{\text{MFPT}(\gamma)}$ over random intervention problems with fixed control cost is small. Note that, so as not to confound approximation accuracy with the MFPT algorithm's ability to find a control gene, we apply both the optimal and MFPT methods using the optimal control gene $g^*$.

### 3. Controllability

An important aspect of prognosis is quantifying the possibility of recovery. In our framework, this amounts to quantifying the *controllability* of a gene regulatory network, a concept borrowed from traditional control theory. Can the network be sufficiently controlled to provide meaningful recovery? We desire a controllability measure where the objective of the control is to reduce the likelihood of observing the undesirable states in the long-run. An optimal control strategy takes into account the cost of control, but here we want only to focus on the possibility of sufficient control, absent concerns with costs, either medical or financial. To this end, we choose the cost of control to be zero. The zero control-cost strategy admits any number of states with active control. Our point (one that is certainly debatable) is that we desire a measure of controllability with no restrictions on the number of times the control might be applied. Thus, a possible approach is to set the cost of control to zero and compute $\Delta P_{g^*}^{\text{opt}}$. To reduce the computational burden, we propose $\Delta P_{g^*}^{\text{MFPT}(0)}$ ($\gamma = 0$) as a controllability measure. Our simulations show that the $\Delta P_{g^*}^{\text{MFPT}(0)}$ is a highly accurate approximation of $\Delta P_{g^*}^{\text{opt}}$ when the cost of control is zero. Therefore, the MFPT algorithm can be employed to determine the controllability of a network. For

example if $\Delta P_{g^*}^{\text{MFPT}(0)}$ is very small, we conclude that the network is not controllable. If $\Delta P_{g^*}^{\text{MFPT}(0)} = 0.5$, then we conclude that it is possible to shift $50\%$ of the probability mass of the undesirable states to desirable ones in the long-run, given the application of the control has zero cost.

### 4. Model-free Intervention

To date, the proposed intervention methods for PBNs are model dependent, requiring knowledge of the transition probability matrix. This can be derived from the PBN if the latter is known. Since in practice PBNs are not known except via system identification from observed data, one is faced with a difficult inference problem [50]. This problem can be avoided by directly inferring the transition probability matrix; however, this is still a formidable task because the complexity of estimating the transition probabilities of a Markov chain increases exponentially with the number of genes in the model. When time-course data are available, the MFPT algorithm can be implemented by directly estimating the mean first-passage times. The estimated mean first-passage times are used to construct the matrices of the mean first-passage times, $K_{\mathcal{U}}$ and $K_{\mathcal{D}}$. The MFPT algorithm can then be applied to the estimated matrices $K_{\mathcal{U}}$ and $K_{\mathcal{D}}$ to devise a *model-free* MFPT control policy.

In the following, we propose a procedure for estimating the mean first-passage times from time-course measurements. Assume $x$ is a desirable state and it is observed at time $t_0$. Further assume that, starting from time $t_0$, the first undesirable state occurs at time $t_0 + k_0$. In other words, it takes $k_0$ time points for the desirable state $x$ to transition (reach) to an undesirable state. Similarly, assume the next observation of state $x$ is at time $t_1$ and since time $t_1$ the first undesirable state occurs at time $t_1 + k_1$. In this example, the average first passage time from state $x$ to the subset of undesirable states is $(k_0 + k_1)/2$. Likewise, one can define an example for an undesirable state $y$ reaching the subset of desirable states. It is evident that for larger numbers of observations, this estimation becomes more accurate.

The above procedure needs to be implemented with low complexity. At each time point, we update the number of steps for each state to reach the opposite subset of states and store the frequency of the occurrence of each state. One needs to update the average first passage times for a new observation. The complexity of estimating the mean first-passage times following our procedure is constant with respect to the number of genes for each iteration.

An advantage of the model-free approach is that the estimated matrices $K_{\mathcal{U}}$ and $K_{\mathcal{D}}$ can be updated whenever new time-course data become available. The possibility of updating the estimated mean first-passage times enables the MFPT algorithm to adapt its control policy to the status of gene interactions. In other words, the model-free MFPT control method is adaptive to changes in the network model. In contrary, the control policy devised by the existing intervention methods cannot adapt to the changes in the status of gene interactions. Once the PBN is inferred form data, the model-dependent control policy is fixed.

Through numerical studies, we will exhibit the effectiveness of the model-free MFPT control policy obtained by estimating the mean first-passage times. On one hand, we will estimate the matrices $K_{\mathcal{U}}$ and $K_{\mathcal{D}}$ based on synthetic time-course data and use the MFPT algorithm to find the control policy; on the other hand, we will use the same time-course data to build a Markov chain representing the dynamics of the model and then find the control policy based on the estimated transition probability matrix using dynamic programming. We will observe that the MFPT control policy based on the estimated mean first-passage times outperforms the control policy devised from the estimated transition probabilities of the Markov chain, given the same set of time-course data, for feasible size data sets.

## C. Complexity Analysis of the MFPT Algorithm

The main objective of an effective intervention strategy is to reduce the likelihood of visiting undesirable states compared to a network without intervention by modifying the long-run behavior of the network. Given a time-course data set, there are two possible approaches for designing a strategy for any model such that its dynamic behavior can be represented as a Markov chain (such as PBN or Dynamic Bayesian Network).

In the first approach, one can estimate the transition probabilities of the states from time-course measurements. Let us call this approach model-dependent. We require all the details about the model, i.e. the transition probabilities of the Markov chain. Various methods can be employed to design an effective intervention strategy based on the estimated model. The optimal control policy can be designed via dynamic programming techniques [11]. In favor of lower computational complexity, an approximation of the optimal control policy can be achieved using the MFPT algorithm.

In the second approach, an effective intervention strategy can be designed directly from time-course measurements. We call this approach model-free. In contrary to the model-dependent approach where the transition probabilities of the Markov chain are needed, we do not require the details of the model. To this end, a model-free algorithm based on reinforcement learning has recently been introduced [39]. This method bypasses the impediment of model estimation and an effective control policy can be designed with a low complexity. We propose that the MFPT algorithm can also be considered as a model-free method. In this section, we analyze the complexity of the model-based and the model-free MFPT algorithms.

### 1. Model-dependent Approach

In the previous section, we introduced the four major applications of the MFPT algorithm: identification of the best control gene, approximation of an optimal control policy, controllability, and model-free intervention. Employment of the MFPT algorithm in the first three applications is considered as a model-dependent approach since it is assumed that the transition probability matrix of the Markov chain is known. Given the model is known, let us compare the computational complexities of the dynamic programming and the MFPT algorithms.

To find an optimal control policy using value or policy iteration, one should iteratively find the value (cost) function until the algorithm reaches the fixed point of the Bellman optimality equation. Once the optimal cost functions are computed, one must check which control value attains the minimum in the right-hand side of the Bellman optimality equation and this procedure should be iterated for all the states. To the best of our knowledge, there does not exist a tight upper bound on the number of iterations required to find an optimal policy using either value or policy iteration, despite recent research initiatives [51]. Given the control gene, the policy iteration algorithm has complexity $\mathbf{O}(2^{3n})$ per iteration, whereas the complete complexity of the MFPT algorithm, which consists of two matrix inversions, is $\mathbf{O}(2^{3n})$. In general, it is known that the policy iteration algorithm converges, but it is not known whether "the number of iterations in policy iteration can be bounded by a polynomial in the instance size" [51]. Even assuming that the number of iterations can be bounded by a polynomial in the number of states, the complexity of the MFPT algorithm is lower than the policy iteration algorithm because it is computed once and does not require iteration. Regarding the value iteration algorithm, the asymptotic upper bound on the number of iterations required to find an optimal policy using the value iteration algorithm is polynomial in the number of states [51]. The degree of the polynomial is determined

to be greater than two in special cases [52, 53]. Given the complexity of each iteration in the value iteration algorithm is $\mathbf{O}(2^{2n})$, the complexity of the value iteration algorithm to find an optimal control policy is $\mathbf{O}(2^{(2+\alpha)n})$, where $\alpha > 1$. Hence, the complexity of the MFPT algorithm is also lower than the complexity of the value iteration algorithm. To find the optimal cost functions for $n$ control genes, the complexity of a dynamic programming algorithm is $n$ times the complexity of this algorithm for one control gene. In contrary, once the mean first-passage time vectors are computed, they can be used to devise MFPT control policies for all control genes.

It is important to point out that for any control gene, in addition to the above complexities, the dynamic programming and the MFPT algorithms must loop over all the states to find their corresponding control policies. In dynamic programming algorithms, to obtain the optimal control policy, one must check which control value attains the minimum in the right-hand side of the Bellman optimality equation and this procedure must be iterated for all the states. In the MFPT algorithm, one must investigate which control value leads to a more favorable mean first-passage time and this procedure must be repeated for all the states.

It is evident from the above analysis that the application of our proposed method is restricted to small number of genes since the complexity of the MFPT algorithm increases exponentially with the number of genes. We should point out that in our application of interest, intervention in gene regulatory networks, the goal is not to model fine-grained molecular interactions among a host of genes, but rather to model a limited number of genes, typically with very coarse quantization, whose regulatory activities are significantly related to a particular aspect of a specific disease, such as metastasis in melanoma [8]. Hence, while the asymptotic results on the complexities of optimal algorithms are encouraging, they are not our main interest; rather our problem deals with networks with small numbers of states. Fig. 19 shows the average execution time of the value and policy iter-

ation algorithms over $1000$ randomly generated intervention problems as a function of the number of genes $n$, along with the execution times of the MFPT algorithm. Per this figure, the execution time of the MFPT algorithm is much smaller than the execution time of the two optimal algorithms. The direct comparison has been limited to 10-gene networks on account of the high complexity of the modeling and optimal intervention algorithms. The maximum size of the intervention problem which can be solved by our MFPT method is hardware-dependent. For instance, our current hardware configuration (single Xeon processor and 1-GB memory) can obtain MFPT intervention policy for a synthetic 15-gene regulatory network, which, given the data limits of current expression measuring technology, is sufficient for the applications in which we are now engaged. Given more memory and processing power, intervention strategies can be designed for larger networks. Should the need arise for larger networks, we will consider implementation on our Beowulf cluster at the Translational Genomics Research Institute.

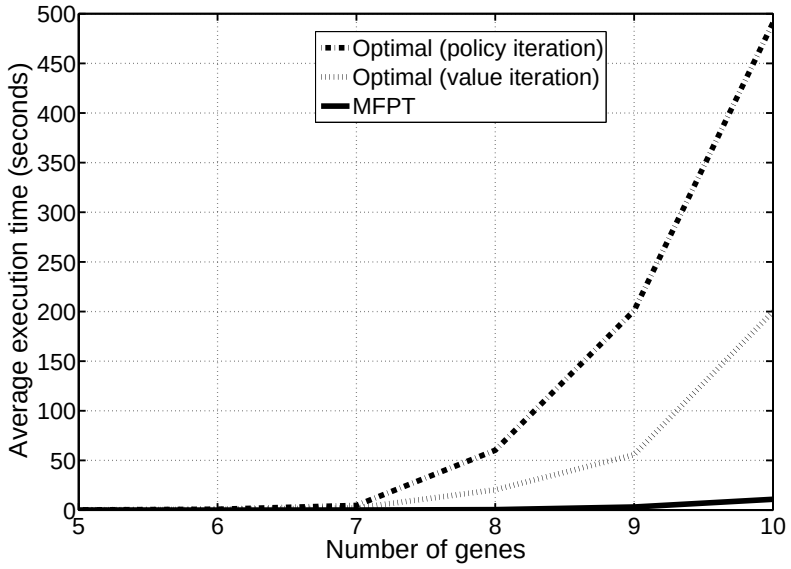



Fig. 19. The average execution time of the value and policy iteration algorithms over $1000$ randomly generated intervention problems as functions of the number of genes, along with the execution times of the MFPT algorithm.

## 2. Model-free Approach

The model-dependent approaches yield effective solutions for large numbers of observations. However, these approaches have major drawbacks in practice. For lower numbers of observations, which correspond better to feasible experimental conditions, estimating the Markov chain yields poor results. Estimation errors may have a huge impact on finding an effective intervention strategy, which is often quite sensitive to changes in the transition probabilities [54]. Furthermore, the complexity of estimating the transition probabilities of a Markov chain increases exponentially with the number of genes in the model, $\mathbf{O}(2^{2n})$. This is in addition to the complexity of designing an effective intervention strategy. Hence, a procedure that can find an effective intervention strategy without having to know the transition probabilities is very attractive.

The model-free based MFPT algorithm (fourth application) estimates the mean first-passage times from time-course measurements. The complexity of estimating these vectors following the proposed procedure in the previous section is constant with respect to $n$ for each iteration, where $n$ denotes the number of genes. In other words, we devise an effective intervention strategy by learning about the mean first-passage times directly from the data.

The highlight of this paper is the possibility of employing the MFPT algorithm in a model-free approach. To this end, we summarize the two main benefits of our proposed model-free method: 1) The complexity of the modeling and intervention is significantly less than that of the model-dependent methods; 2) In contrary to the optimal control problem approach, which is sensitive to changes in the system, the MFPT algorithm needs the average behavior of the system and is expected to be more appealing for smaller numbers of observations. We corroborate this claim in the result section by comparing the model-free MFPT method with the model-dependent optimal control method.

## D. Results and Discussion

In this section, we first study the performance of the MFPT algorithm for each of the aforementioned applications through extensive simulations of random PBNs. We then compare the performance of the MFPT algorithm and the influence method for the network obtained from a melanoma gene-expression data set.

### 1. Synthetic Networks

We postulate the following cost-per-stage:

$$g(u,j) = \begin{cases} 0 & \text{if } u = 0 \text{ and } j \in \mathcal{D}, \\ 10 & \text{if } u = 0 \text{ and } j \in \mathcal{U}, \\ \text{c} & \text{if } u = 1 \text{ and } j \in \mathcal{D}, \\ 10 + \text{c} & \text{if } u = 1 \text{ and } j \in \mathcal{U}, \end{cases}$$

where $c$ is the cost of control. The target gene is chosen to be the most significant gene in the GAP. We assume the up-regulation of the target gene is undesirable. Consequently, the state-space is partitioned into desirable states, $\mathcal{D} = \{0, \ldots, 2^{n-1} - 1\}$, and undesirable states, $\mathcal{U} = \{2^{n-1}, \ldots, 2^n - 1\}$, where $n$ is the number of genes. The cost values have been chosen in accord with an earlier study [11]. Since our objective is to down-regulate the target gene, a higher cost is assigned to destination states having an up-regulated target gene. Moreover, for a given status of the target gene for a destination state, a higher cost is assigned when the control is applied, versus when it is not. In order to investigate the effect of the cost of control in our algorithm, we vary its value from $0$ to $10$, which is the cost of the undesirable states.

We generate random PBNs in the following manner. Each PBN consists of $10$ constituent BNs. Each BN is randomly generated with a specific bias $b$, the bias being the probability that a randomly generated Boolean function takes on the value 1. Since the bias

affects the dynamical properties of randomly generated BNs [4], we take it as a parameter in our simulations. We randomly select the bias $b$ of a BN from a beta distribution. We vary the mean of the beta distribution from $0.3$ to $0.7$ with step-size $0.1$. The variance $\sigma^2$ of the beta distribution is set to a constant value $0.0001$. This provides random biases from low ($0.3$) to high ($0.7$). We generate $1000$ random PBNs for each bias mean. For each PBN, the transition probabilities of the corresponding Markov chain are estimated. The above procedure is repeated for networks of $5$ to $10$ genes. Due to the computational complexity of the optimal stochastic control problem and the estimation of the transition probabilities of the corresponding Markov chain, the study of a large number of networks beyond $10$ genes is outside our current computational capacity.

a. Identification of the Best Control Gene

We first show the performance of the MFPT algorithm and the influence method when they are employed to predict the best control gene. It is assumed that the cost of control $c$ is equal to $1$. In Tables VIII, X, IX, and XI, we compare the performances of the MFPT algorithm and the influence method for predicting the best control gene. First the optimal control policy for each control gene is obtained by a dynamic programming algorithm. The best control gene $g^*$ is found based on (6.4). Similarly, the MFPT control policy for each control gene is computed and the predicted control gene $\hat{g}$ is found based on (6.5). The influence method is also employed to predict the best control gene. The predicted best control gene by the influence method is denoted $\breve{g}$. We define the probability of the correct prediction of each method to be the number of PBNs for which the method correctly predicts the best control gene divided by the total number of PBNs in the experiment. The probabilities of correctly predicting the best control gene by the MFPT algorithm and the influence method are shown in Tables VIII and X. The average differences between proportions of reduction in the total probability of undesirable states corresponding to the gene predicted by each

method and the best control gene, i.e. $(\Delta P_{g^*}^{\text{opt}} - \Delta P_{\hat{g}}^{\text{opt}})$ and $(\Delta P_{g^*}^{\text{opt}} - \Delta P_{\breve{g}}^{\text{opt}})$, are shown in Tables IX and XI. In our experiments, the probability of the correct prediction by the MFPT algorithm is always greater than $0.94$. Table. IX shows that $\Delta P_{g^*}^{\text{opt}} - \Delta P_{\hat{g}}^{\text{opt}}$ on average is less than $0.0002$.

Table VIII. The probability of finding the best control gene with the MFPT algorithm when c = 1 for networks with different number of genes.

| Bias | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|
| (A) 5 genes | 0.9850 | 0.9640 | 0.9570 | 0.9600 | 0.9720 |
| (B) 6 genes | 0.9430 | 0.9700 | 0.9760 | 0.9580 | 0.9870 |
| (C) 7 genes | 0.9440 | 0.9680 | 0.9670 | 0.9700 | 0.9570 |
| (D) 8 genes | 0.9660 | 0.9740 | 0.9860 | 0.9790 | 0.9710 |
| (E) 9 genes | 0.9132 | 0.9233 | 0.9741 | 0.9812 | 0.9812 |
| (F) 10 genes | 0.9470 | 0.9570 | 0.9860 | 0.9690 | 0.9610 |

Table IX. The average difference between the proportions of reduction in the total probability of undesirable states obtained by the best control gene $g^*$ and the predicted control gene obtained by the MFPT algorithm $\hat{g}$ for networks with various number of genes.

| Bias | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|
| (A) 5 genes | 0.0000 | 0.0000 | 0.0001 | 0.0001 | 0.0001 |
| (B) 6 genes | 0.00016 | 0.00010 | 0.00003 | 0.00006 | 0.00006 |
| (C) 7 genes | 0.00013 | 0.00013 | 0.00006 | 0.00005 | 0.00005 |
| (D) 8 genes | 0.0001 | 0.00008 | 0.00005 | 0.00002 | 0.00003 |
| (E) 9 genes | 0.0002 | 0.00001 | 0.00001 | 0.00004 | 0.00001 |
| (F) 10 genes | 0.0001 | 0.00008 | 0.00003 | 0.00002 | 0.00005 |

Table X. The probability of finding the best control gene with the influence method when c = 1 for networks with different number of genes.

| Bias | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|
| (A) 5 genes | 0.6660 | 0.6240 | 0.5480 | 0.5670 | 0.5740 |
| (B) 6 genes | 0.5630 | 0.5320 | 0.4790 | 0.5070 | 0.5340 |
| (C) 7 genes | 0.5470 | 0.5550 | 0.5320 | 0.5460 | 0.5060 |
| (D) 8 genes | 0.5190 | 0.5290 | 0.5290 | 0.5780 | 0.5600 |
| (E) 9 genes | 0.5086 | 0.5186 | 0.5186 | 0.5676 | 0.5496 |
| (F) 10 genes | 0.5480 | 0.5230 | 0.5030 | 0.4010 | 0.4610 |

The performance of the influence method is also shown in Tables X and XI. These tables suggest that approximately $0.60$ of the time the influence method's prediction is correct. In general, $\Delta P_{g^*}^{\text{opt}} - \Delta P_{\breve{g}}^{\text{opt}}$ is greater than $0.001$. Tables XII, XIII, XIV, and XV

Table XI. The average difference between the proportions of reduction in the total probability of undesirable states obtained by the best control gene $g^*$ and the predicted control gene obtained by the influence method $\breve{g}$ for networks with various number of genes.

| Bias | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|
| (A) 5 genes | 0.0079 | 0.0109 | 0.0102 | 0.0133 | 0.0134 |
| (B) 6 genes | 0.0081 | 0.0107 | 0.0140 | 0.0207 | 0.0158 |
| (C) 7 genes | 0.0086 | 0.0108 | 0.0104 | 0.0115 | 0.0130 |
| (D) 8 genes | 0.0100 | 0.0137 | 0.0151 | 0.0180 | 0.0131 |
| (E) 9 genes | 0.0016 | 0.0228 | 0.0134 | 0.0415 | 0.0130 |
| (F) 10 genes | 0.0104 | 0.0097 | 0.0152 | 0.0178 | 0.0211 |

Table XII. The probability of finding the best control gene with the MFPT algorithm.

| Bias | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|
| (A) c=2 | 0.9034 | 0.9121 | 0.8983 | 0.8848 | 0.9085 |
| (B) c=4 | 0.8614 | 0.8897 | 0.8839 | 0.8701 | 0.8035 |

Table XIII. The average difference between the proportions of reduction in the total probability of undesirable states obtained by the best control gene $g^*$ and the predicted control gene obtained by the MFPT algorithm $\hat{g}$ with various cost values.

| Bias | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|
| (A) c=2 | 0.0004 | 0.0005 | 0.0006 | 0.0008 | 0.0005 |
| (B) c=4 | 0.0020 | 0.0013 | 0.0014 | 0.0020 | 0.0022 |

Table XIV. The probability of finding the best control gene with the influence method.

| Bias | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|
| (A) c=2 | 0.6432 | 0.6670 | 0.5950 | 0.5755 | 0.6050 |
| (B) c=4 | 0.6151 | 0.6247 | 0.6616 | 0.6321 | 0.6533 |

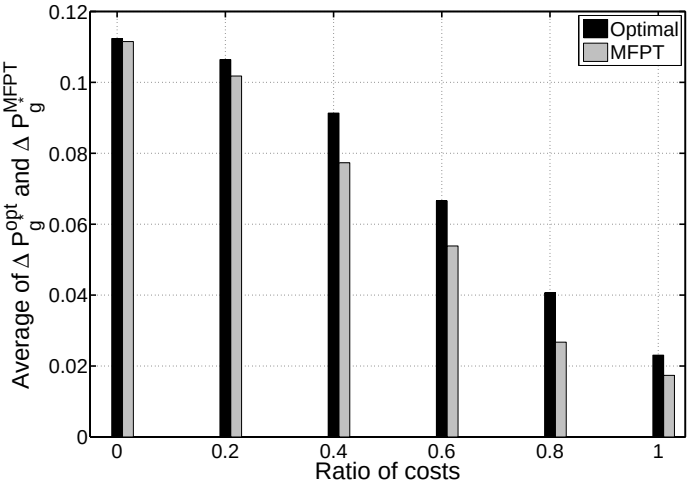
Table XV. The average difference between the proportions of reduction in the total probability of undesirable states obtained by the best control gene $g^*$ and the predicted control gene obtained by the influence method $\breve{g}$ with various cost values.

| Bias | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 |
|---|---|---|---|---|---|
| (A) c=2 | 0.0098 | 0.0102 | 0.0120 | 0.0133 | 0.0144 |
| (B) c=4 | 0.0103 | 0.0190 | 0.0151 | 0.0190 | 0.0115 |

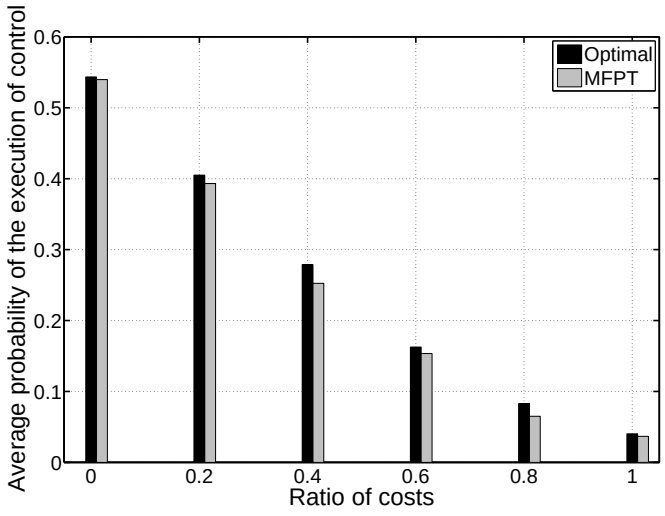
show the performance of the MFPT algorithm for higher values of $c$. Although the correct prediction of the MFPT algorithm slightly degrades for higher values of the control cost $c$, it still outperforms the influence method.

b. Approximation of the Optimal Control Policy

Once the best control gene $g^*$ is known, the corresponding MFPT control policy $\hat{\mu}_{g^*}^{\gamma}$ can be used as an approximate solution to the optimal stochastic control problem. As previously explained, the parameter $\gamma$ depends on the ratio of the cost of control to the cost of undesirable states. We numerically find the minimal value of the parameter $\gamma$ for each control cost so that on average $\Gamma_{g^*}^{\text{opt}} > \Gamma_{g^*}^{\text{MFPT}(\gamma)}$. It is shown that the average of $\Delta P_{g^*}^{\text{opt}} - \Delta P_{g^*}^{\text{MFPT}(\gamma)}$ over random intervention problems with fixed control cost is small. We generate random PBNs following the procedure explained earlier. The cost of undesirable states is fixed. For the PBNs with identical bias mean, we formulate the intervention problems with various costs of control, which are varied such that the ratio of the cost of control to the cost of undesirable states changes from $0$ to $1$. For PBNs with each bias mean and cost of control, we compute the averages of $\Delta P_{g^*}^{\text{opt}}$ and $\Gamma_{g^*}^{\text{opt}}$. The averages are taken over $1000$ intervention problems with PBNs whose bias means are fixed. Similarly, the averages of $\Delta P_{g^*}^{\text{MFPT}(\gamma)}$ and $\Gamma_{g^*}^{\text{MFPT}(\gamma)}$ are found. Furthermore, we compute the average of these averages over all bias means. The parameter $\gamma$ is determined such that $\Gamma_{g^*}^{\text{MFPT}(\gamma)} < \Gamma_{g^*}^{\text{opt}}$. For each given control cost, we show the behavior of $\Delta P_{g^*}^{\text{opt}}$ and $\Gamma_{g^*}^{\text{opt}}$ ($\Delta P_{g^*}^{\text{MFPT}}$ and $\Gamma_{g^*}^{\text{MFPT}(\gamma)}$). As seen in Fig. 20a, both $\Delta P_{g^*}^{\text{MFPT}(\gamma)}$ and $\Delta P_{g^*}^{\text{opt}}$ decrease when the ratio of the cost of control to the cost of undesirable states increases. We observe that on average the difference between $\Delta P_{g^*}^{\text{opt}}$ and $\Delta P_{g^*}^{\text{MFPT}(\gamma)}$ is less than $0.02$. As Fig. 20b shows, the probability of the execution of control for both policies decreases as the cost of control increases. Table XVI shows the relation of the parameter $\gamma$ with the ratio of the cost of control to the cost of undesirable states found in the above experiment.




**(a)**




**(b)**

Fig. 20. a) Average of $\Delta P_{g^*}^{\mathrm{MFPT}(\gamma)}$ and $\Delta P_{g^*}^{\mathrm{opt}}$ b) Average of $\Gamma_{g^*}^{\mathrm{MFPT}(\gamma)}$ and $\Gamma_{g^*}^{\mathrm{opt}}$. Horizontal axis shows the ratio of the cost of control to the cost of undesirable states. Values of $\gamma$ are chosen from Table XVI.

Table XVI. Value of the parameter $\gamma$ as a function of the ratio of the cost of control to the cost of undesirable states

| Ratio of costs | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 1 |
|---|---|---|---|---|---|---|
| $\gamma$ | 0 | 0.29 | 0.61 | 0.91 | 1.5 | 1.94 |

Table XVII. Conservative value of the parameter $\gamma$ as a function of the ratio of the cost of control to the cost of undesirable states

| Ratio of costs | 0 | 0.2 | 0.4 | 0.6 | 0.8 | 1 |
|---|---|---|---|---|---|---|
| $\gamma$ | 0.05 | 0.5 | 0.9 | 1.1 | 1.9 | 2.3 |

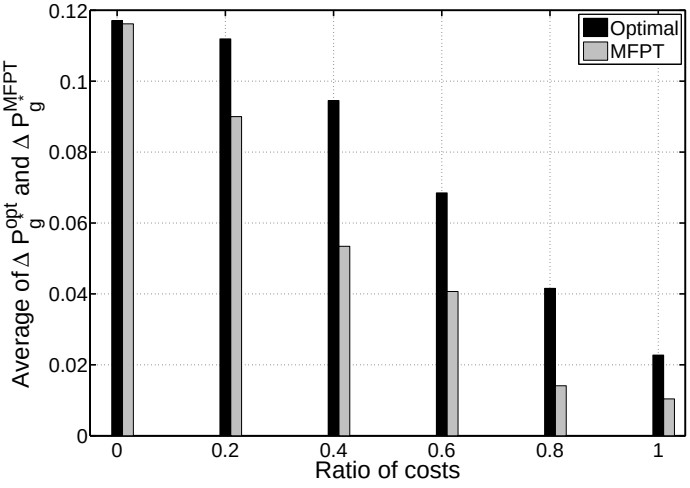
Since the values in Table XVI are found from random PBNs, one can have a conservative approach and choose the parameter $\gamma$ to be greater than the proposed values. To this end, $\Gamma_{g^*}^{\text{MFPT}(\gamma)}$ is smaller than $\Gamma_{g^*}^{\text{opt}}$ in each intervention problem. Fig. 21 and Table XVII show the outcomes of the same experiment explained earlier when the parameter $\gamma$ is chosen conservatively. In all the intervention problems of this experiment, $\Gamma_{g^*}^{\text{MFPT}(\gamma)} < \Gamma_{g^*}^{\text{opt}}$ and the deviation of $\Delta P_{g^*}^{\text{MFPT}(\gamma)}$ from $\Delta P_{g^*}^{\text{opt}}$ is smaller than $0.04$.

c. Controllability

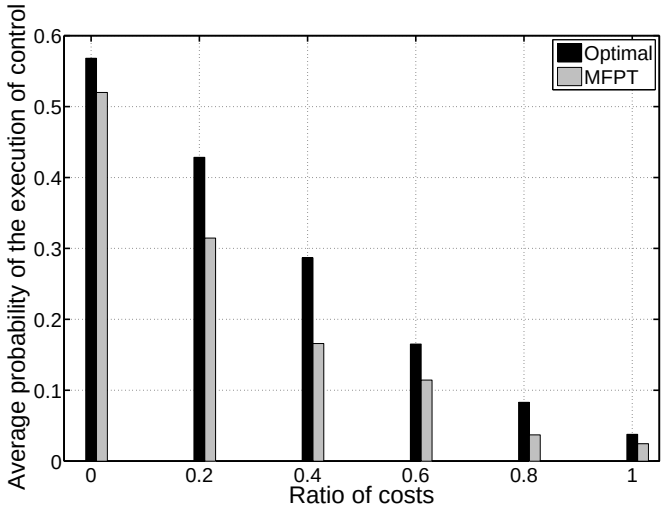
To corroborate that the MFPT algorithm can be employed to determine the controllability of a network, we consider the results in Fig. 20. In this figure, when the cost of control is zero ($\gamma = 0$), $\Delta P_{g^*}^{\text{MFPT}(0)}$ is an accurate approximation of the $\Delta P_{g^*}^{\text{opt}}$. The average of the difference $\Delta P_{g^*}^{\text{MFPT}(0)} - \Delta P_{g^*}^{\text{opt}}$ has a negligible value equal to $0.0007$.

d. Model-free Intervention

To compare the performance of the model-free MFPT control algorithm with an optimal control algorithm, where the latter includes estimation of the transition probability matrix, we generate synthetic time-course data for $100,000$ time-steps from an existing model. Using the synthetic time-course data, we estimate the mean first-passage times after each $10^k$




**(a)**




**(b)**

Fig. 21. a) Average of $\Delta P_{g^*}^{\mathrm{MFPT}(\gamma)}$ and $\Delta P_{g^*}^{\mathrm{opt}}$ b) Average of $\Gamma_{g^*}^{\mathrm{MFPT}(\gamma)}$ and $\Gamma_{g^*}^{\mathrm{opt}}$. Horizontal axis shows the ratio of the cost of control to the cost of undesirable states. Values of $\gamma$ are chosen conservatively from Table XVII.

time-steps, for $k = 2, \ldots, 5$, and fix the cost of control to have the value 1. As the duration of estimating the mean first-passage times increases, $\Delta P_{g^*}^{\mathrm{MFPT}(\gamma)}$ approaches $\Delta P_{g^*}^{\mathrm{opt}}$. Fig. 22 shows the average of $|\Delta P_{g^*}^{\mathrm{opt}} - \Delta P_{g^*}^{\mathrm{MFPT}(\gamma)}|$, where $\Delta P_{g^*}^{\mathrm{opt}}$ is obtained from the original transition probabilities, with various estimating durations over $1000$ trials. For an optimal control policy based on the Markov chain estimated from the data, we denote the shift in the steady-state distribution by $\widehat{\Delta P_{g^*}^{\mathrm{opt}}}$. Fig. 22 shows the average of $|\Delta P_{g^*}^{\mathrm{opt}} - \widehat{\Delta P_{g^*}^{\mathrm{opt}}}|$ with various estimating durations over $1000$ trials. The graphs clearly demonstrate the superior performance of the model-free approach using the MFPT algorithm. In particular, for lower numbers of observations, which correspond better to feasible experimental conditions, estimating the Markov chain yields poor results, whereas the MFPT approximation performs quite well.

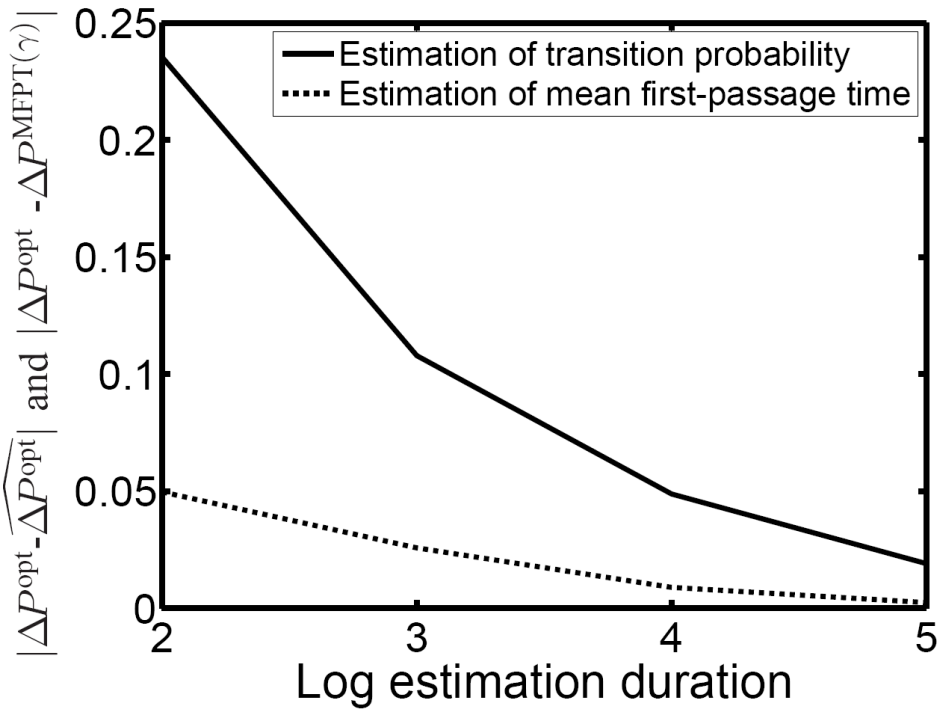



Fig. 22. Average of $|\Delta P^{\mathrm{opt}}\text{-}\widehat{\Delta P^{\mathrm{opt}}}|$ (solid) and $|\Delta P^{\mathrm{opt}}\text{-}\Delta P^{\mathrm{MFPT}(\gamma)}|$ (dash) over 1000 trials as a function of the logarithm of estimation duration.

## 2. Melanoma Gene Expression

In this section, we compare the performances of optimal and MFPT control polices in the context of a gene network developed from steady-state data. Based on our objective, the cost of control is assumed to be 1 and the states are assigned penalties according to the following scheme:

$$r(u,j) = \begin{cases} 0 & \text{if } u = 0 \text{ and } j \in \mathcal{D}, \\ 5 & \text{if } u = 0 \text{ and } j \in \mathcal{U}, \\ 1 & \text{if } u = 1 \text{ and } j \in \mathcal{D}, \\ 6 & \text{if } u = 1 \text{ and } j \in \mathcal{U}, \end{cases}$$

which is the same cost structure as assumed in [11]. Since our objective is to down-regulate the WNT5A gene, a higher penalty is assigned for destination states having WNT5A up-regulated. Also, for a given WNT5A status for the destination state, a higher penalty is assigned when the control is active versus when it is not. Note that the cost scheme reflects our objective; in practice, the actual values would have to be assigned by a physician according to his or her understanding of the disease. Optimal and MFPT control policies are found for the melanoma-related PBN. Table XVIII summarizes the amount of the shift in the total probability mass of the undesirable states obtained by each of these two methods. We apply the influence method to predict the best control gene. We then compare the prediction of the influence method with the prediction of the MFPT algorithm and the optimal gene determined directly by the solution of a dynamic programming algorithm. Table XIX shows the ranking of the genes based on: direct solution of the optimal control policy, the MFPT algorithm, and the influence method. The MFPT method not only predicts the best control gene, but it also exactly predicts the ranking of the control genes. As Table XIX shows, the influence method does a poor job on predicting the best control gene.

Table XVIII. $\Delta P_g^{\text{opt}}$ and $\Delta P_g^{\text{MFPT}(\gamma)}$ for all control genes $g$ in the melanoma case-study

| Gene ($g$) | STC2 | Synuclein | HADHB | MART1 | PHOC | MMP3 | RET1 | S100P | pirin |
|---|---|---|---|---|---|---|---|---|---|
| $\Delta P_g^{\text{opt}}$ | 0.0733 | 0.0892 | 0.1453 | 0.1104 | 0.2325 | 0.1121 | 0.0529 | 0.1032 | 0.1305 |
| $\Delta P_g^{\text{MFPT}(\gamma)}$ | 0.0721 | 0.0824 | 0.1437 | 0.1071 | 0.2312 | 0.1120 | 0.0507 | 0.1021 | 0.1272 |

Table XIX. Comparison of the control gene ranking based on $\Delta P_{g^*}^{\text{opt}}$, $\Delta P_{\hat{g}}^{\text{opt}}$, and $\Delta P_{\breve{g}}^{\text{opt}}$

| Rank | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| Optimal | PHOC | HADHB | pirin | MMP3 | MART1 | S100P | Synuclein | STC2 | RET1 |
| MFPT | PHOC | HADHB | pirin | MMP3 | MART1 | S100P | Synuclein | STC2 | RET1 |
| Influence | MMP3 | HADHB | MART1 | S100P | STC2 | pirin | PHOC | RET1 | Synuclein |

# CHAPTER VII

# CONCLUSION

In this work, we discussed several approaches that have been recently developed for addressing the issue of inference and intervention in gene regulatory networks. The results reported indicate that significant progress has been made in this area. Our current collaborations with the Translational Genomics Research Institute (TGen) aims at validating the efficacy of mathematically derived intervention strategies for controlling the pathological behavior of cancerous cells. This research has the potential to dramatically impact future medical practice. Engineering-based interventions that achieve cellular behavior alteration will enhance current cancer treatment and lead to the development of personalized cancer therapies. In the following, I outline some promising research directions for the future. The feasibility of the proposed project critically depends on a genuine collaboration between biologists, physicians, and engineers

## Effective Intervention in Heterogeneous Metastatic Cells

Metastasis, the spread of cancerous cells from the primary tumor to distant organs, and their relentless growth, is the most fearsome aspect of cancer. Despite significant improvements in diagnosis, surgical techniques, and general patient care, most deaths from cancer are due to metastases that are resistant to conventional therapies. The main barrier to the treatment of metastases is the biological heterogeneity of cancer cells in the primary tumor and in metastases. Continual empiricism in the treatment of cancer metastasis is unlikely to produce significant improvements in cancer therapy. Therefore, understanding the pathogenesis of metastasis at the systemic level is an important goal of cancer research.

By the time of initial diagnosis, malignant tumors already contain multiple cell subpopulations with diverse biological heterogeneity. The heterogeneous nature of the re-

sponse of malignant tumor cell sub-populations to cytotoxic drugs makes it unlikely that a single treatment regimen will be able to kill all the cells in a tumor. The goal of this research project is to devise therapeutic methods to maximally eradicate heterogenous metastatic cells. To achieve this goal, we envision a number of objectives for which both models and experiments must be advanced.

- Detect major cell subpopulations in a tumor.
- Estimate the growth rate of each cell subpopulation.
- Devise effective therapeutic methods to halt uncontrolled cell-growth in cell subpopulations.
- Devise adaptive strategy to eradicate various cell subpopulations.

The current approach to cancer therapy is to experiment with one drug after the other until one drug works for a particular patient or all available options get exhausted. If this proposed research is successful, it should be possible to study a cancer patient's tumor in vitro and predict apriori which treatment or set of treatments is most likely to work for that patient. This should enhance the current trial and error approach to cancer therapy and thereby considerably improve the quality of life and therapy outcome for cancer patients.

# REFERENCES

[1] T. Schlitt and A. Brazma, "Current approaches to gene regulatory network modelling," *BMC Bioinformatics*, vol. 8, pp. 1–22, 2007.

[2] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using bayesian networks to analyze expression data," *Journal of Computational Biology*, vol. 7, no. 3, pp. 601–620, 2000.

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

[4] I. Shmulevich and E. R. Dougherty, *Genomic Signal Processing*, Princeton: Princeton University Press, 2007.

[5] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang, "Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks," *Bioinformatics*, vol. 18, no. 2, pp. 261–274, 2002.

[6] I. Shmulevich, E. R. Dougherty, and W. Zhang, "Gene perturbation and intervention in probabilistic Boolean networks," *Bioinformatics*, vol. 18, no. 10, pp. 1319–1331, 2002.

[7] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, 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, V. Sondak, N. Hayward, and J. Trent, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–450, 2000.

[8] A. Datta, R. Pal, A. Choudhary, and E. R. Dougherty, "Control approaches for probabilistic gene regulatory networks," *IEEE Signal Processing Mag.*, vol. 24, no. 1, pp. 54–63, 2007.

[9] A. Datta and E. R. Dougherty, *Introduction to Genomic Signal Processing with Control*, Roca Baton, FL: CRC Press, 2006.

[10] A. Datta, A. Choudhary, M. L. Bittner, and E. R. Dougherty, "External control in Markovian genetic regulatory networks," *Machine Learning*, vol. 52, no. 1/2, pp. 169–191, 2003.

[11] R. Pal, A. Datta, and E. R. Dougherty, "Optimal infinite-horizon control for probabilistic Boolean networks," *IEEE Transactions on Signal Processing*, vol. 54, no. 6, pp. 2375–2387, 2006.

[12] G. Vahedi, I. Ivanov, and E.R. Dougherty, "Inference of Boolean networks under constraint on bidirectional gene relationships," *IET Systems Biology*, vol. 3, no. 3, pp. 191–202, 2009.

[13] G. Vahedi, B. Faryabi, J. F. Chamberland, A. Datta, and E. R. Dougherty, "Sampling-rate-dependent probabilistic Boolean networks," *Submitted to the Journal of Theoretical Biology*, 2009.

[14] R. Simon and L. Norton, "The Norton-Simon hypothesis: Designing more effective and less toxic chemotherapeutic regimens," *Nature Clinical Practice Oncology*, vol. 3, no. 8, pp. 406–407, 2006.

[15] G. Vahedi, B. Faryabi, J. F. Chamberland, A. Datta, and E. R. Dougherty, "Optimal intervention strategies for cyclic therapeutic methods," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 2, pp. 281–291, 2009.

[16] G. Vahedi, B. Faryabi, J. F. Chamberland, A. Datta, and E. R. Dougherty, "Intervention in gene regulatory networks via a stationary mean-first-passage-time control policy," *IEEE Transactions on Biomedical Engineering*, vol. 55, no. 10, pp. 2319–2331, 2008.

[17] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *Journal of Theoretical Biology*, vol. 22, no. 3, pp. 437–467, 1969.

[18] D. P. Bertsekas, *Dynamic Programming and Optimal Control*, Belmont, MA: Athena Scientific, 2001.

[19] J. R. Norris, *Markov Chains*, Cambridge, U.K.: Cambridge University Press, 1998.

[20] 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.

[21] S. Kim, H. Li, E. R. Dougherty, N. W. Cao, Y. D. Chen, M. L. Bittner, and E. B. Suh, "Can Markov chain models mimic biological regulation?," *Journal of Biological Systems*, vol. 10, no. 4, pp. 337–357, 2002.

[22] R. Pal, I. Ivanov, A. Datta, M. Bittner, and E. R. Dougherty, "Generating Boolean networks with a prescribed attractor structure," *Bioinformatics*, vol. 21, no. 21, pp. 4021–4025, 2005.

[23] A. Faure, A. Naldi, C. Chaouiya, and D. Theiffry, "Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle," *Bioinformatics*, vol. 22, no. 14, pp. e124–e131, 2006.

[24] H. De Jong, "Modeling and simulation of genetic regulatory systems: A literature review," *Journal of Computational Biology*, vol. 9, no. 1, pp. 67–103, 2002.

[25] I. Ivanov and E. R. Dougherty, "Modeling genetic regulatory networks: Continuous or discrete?," *Journal of Biological Systems*, vol. 14, no. 2, pp. 219–229, 2006.

[26] E. P. van Someren, L. F. A. Wessels, E. Backer, and M. J. T. Reinders, "Multi-criterion optimization for genetic network modeling," *Signal Processing*, vol. 83, no. 4, pp. 763–775, 2003.

[27] S. Liang, S. Fuhrman, and R. Somogyi, "REVEAL, a general reverse engineering algorithm for inference of genetic network architectures," in *Pac. Symp. Biocomput.*, Hawaii, 1999, vol. 3, pp. 18–29.

[28] T. Akutsu, "Identification of genetic networks from a small number of gene expression patterns under the Boolean network model," in *Pac. Symp. Biocomput.*, Hawaii, 1999, vol. 4, pp. 17–28.

[29] H. Lahdesmaki, I. Shmulevich, and O. Yli-Harja, "On learning gene regulatory networks under the Boolean network model," *Machine Learning*, vol. 52, no. 1-2, pp. 147–167, 2003.

[30] L. Glass and S. A. Kauffman, "Logical analysis of continuous, nonlinear biochemical control networks," *Journal of Theoretical Biology*, vol. 39, no. 1, pp. 103–129, 1973.

[31] X. B. Zhou, X. D. Wang, R. D. Pal, I. Ivanov, M. Bittner, and E. R. Dougherty, "A Bayesian connectivity-based approach to constructing probabilistic gene regulatory networks," *Bioinformatics*, vol. 20, no. 17, pp. 2918–2927, 2004.

[32] S. Huang, "Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery," *Journal of Molecular Medicine*, vol. 77, no. 6, pp. 469–480, 1999.

[33] E. R. Dougherty and Y. Xiao, "Design of probabilistic Boolean networks under the requirement of contextual data consistency," *IEEE Transactions on Circuits and Systems I*, vol. 53, no. 11, pp. 2431–2437, 2006.

[34] E. R. Dougherty, S. Kim, and Y. D. Chen, "Coefficient of determination in nonlinear signal processing," *Signal Processing*, vol. 80, no. 10, pp. 2219–2235, 2000.

[35] V. Kaufman and B. Drossel, "On the properties of cycles of simple Boolean networks," *European Physical Journal B*, vol. 43, no. 1, pp. 115–124, 2005.

[36] S. C. Kim, E. R. Dougherty, Y. D. Chen, K. Sivakumar, P. Meltzer, J. M. Trent, and M. Bittner, "Multivariate measurement of gene expression relationships," *Genomics*, vol. 67, no. 2, pp. 201–209, 2000.

[37] B. Faryabi, J. F. Chamberland, G. Vahedi, A. Datta, and E. R. Dougherty, "Optimal intervention in asynchronous genetic regulatory networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 2, no. 3, pp. 412–423, 2008.

[38] E. R. Dougherty, A. Datta, and C. Sima, "Research issues in genomic signal processing," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 46–68, 2005.

[39] B. Faryabi, A. Datta, and E. Dougherty, "On approximate stochastic control in genetic regulatory networks," *IET Systems Biology*, vol. 1, no. 6, pp. 361–368, 2007.

[40] Tatsuya Akutsu, Morihiro Hayashida, Wai-Ki Ching, and Michael K. Ng, "Control of Boolean networks: Hardness results and algorithms for tree structured networks," *Journal of Theoretical Biology*, vol. 244, no. 4, pp. 670–679, 2007.

[41] T. Miyashita and J. C. Reed, "Tumor-suppressor p53 is a direct transcriptional activator of the human bax gene," *Cell*, vol. 80, no. 2, pp. 293–299, 1995.

[42] L. B. Owenschaub, W. Zhang, J. C. Cusack, L. S. Angelo, S. M. Santee, T. Fujiwara, J. A. Roth, A. B. Deisseroth, W. W. Zhang, E. Kruzel, and R. Radinsky, "Wild-type human p53 and a temperature-sensitive mutant induce Fas/Apo-1 expression," *Molecular and Cellular Biology*, vol. 15, no. 6, pp. 3032–3040, 1995.

[43] W. S. Eldeiry, T. Tokino, V. E. Velculescu, D. B. Levy, R. Parsons, J. M. Trent, D. Lin, W. E. Mercer, K. W. Kinzler, and B. Vogelstein, "Waf1, a potential mediator of p53 tumor suppression," *Cell*, vol. 75, no. 4, pp. 817–825, 1993.

[44] C. B. Harley, "Telomere loss - mitotic clock or genetic time bomb," *Mutation Research*, vol. 256, no. 2-6, pp. 271–282, 1991.

[45] N. W. Kim, M. A. Piatyszek, K. R. Prowse, C. B. Harley, M. D. West, P. L. C. Ho, G. M. Coviello, W. E. Wright, S. L. Weinrich, and J. W. Shay, "Specific association of human telomerase activity with immortal cells and cancer," *Science*, vol. 266, no. 5193, pp. 2011–2015, 1994.

[46] E. Hiyama, K. Hiyama, T. Yokoyama, Y. Matsuura, M. A. Piatyszek, and J. W. Shay, "Correlating telomerase activity levels with human neuroblastoma outcomes," *Nature Medicine*, vol. 1, no. 3, pp. 249–255, 1995.

[47] W. Zhang, M. A. Piatyszek, T. Kobayashi, E. Estey, M. Andreeff, A. B. Deisseroth, W. E. Wright, and J. W. Shay, "Telomerase activity in human acute myelogenous leukemia: Inhibition of telomerase activity by differentiation-inducing agents," *Clinical Cancer Research*, vol. 2, no. 5, pp. 799–803, 1996.

[48] S. W. Lowe, E. M. Schmitt, S. W. Smith, B. A. Osborne, and T. Jacks, "p53 is required for radiation-induced apoptosis in mouse thymocytes," *Nature*, vol. 362, no. 6423, pp. 847–849, 1993.

[49] A. Datta, R. Pal, and E. R. Dougherty, "Intervention in probabilistic gene regulatory networks," *Current Bioinformatics*, vol. 1, no. 2, pp. 167–184, 2006.

[50] S. Marshall, L. Yu, Y. Xiao, and E. R. Dougherty, "Inference of a probabilistic Boolean network from a single observed temporal sequence," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2007, pp. 32454–32569, 2007.

[51] V. D. Blondel and J. N. Tsitsiklis, "A survey of computational complexity results in systems and control," *Automatica*, vol. 36, no. 9, pp. 1249–1274, 2000.

[52] O. Madani, "Complexity results for infinite-horizon markov decision processes," Ph.D. dissertation, Dept. Comp. Sci. Eng., Univ. of Washington, Seattle, 2000.

[53] P. Tseng, "Solving h-horizon, stationary Markov decision-problems in time proportional to log(h)," *Operations Research Letters*, vol. 9, no. 5, pp. 287–297, 1990.

[54] E. Feinberg and A. Shwartz., *Handbook of Markov Decision Processes*, Boston: Kluwers Academic Publishers, 2002.

# VITA

Golnaz Vahedi received her B.S. degree in electrical and computer engineering from Sharif University of Technology, Tehran, Iran, in 2001, and her M.S. degree also in electrical and computer engineering from the University of Alberta, Edmonton, Canada, in 2004. She received her Ph.D. degree in electrical & computer engineering at Texas A&M University, College Station, in August 2009. Her research interests lie in the areas of computational biology, genomic signal processing, and bioinformatics. She has been the winner of the 2004 Innovation-Research Contest of the Association of Health Technologies Industry, Montreal, Canada. She is a member of the IEEE Engineering in Medicine and Biology Society (EMBS), International Society for Computational Biology (ISCB), and Women in Science and Engineering (WISE). She can be reached at golnaz.vahedi@ gmail.com or Department of Electrical Engineering, c/o Dr. Edward Dougherty, Texas A&M University, College Station, TX, 77843-3128.