Control of Gene Regulatory Networks using Bayesian Inverse Reinforcement Learning

Mahdi Imani, *Student Member, IEEE* and Ulisses M. Braga-Neto, *Senior Member, IEEE*

***Abstract*—Control of gene regulatory networks (GRNs) to shift gene expression from undesirable states to desirable ones has received much attention in recent years. Most of the existing methods assume that the cost of intervention at each state and time point, referred to as the *immediate cost function*, is fully known. In this paper, we employ the Partially-Observed Boolean Dynamical System (POBDS) signal model for a time sequence of noisy expression measurement from a Boolean GRN and develop a Bayesian Inverse Reinforcement Learning (BIRL) approach to address the realistic case in which the only available knowledge regarding the immediate cost function is provided by the sequence of measurements and interventions recorded in an experimental setting by an expert. The Boolean Kalman Smoother (BKS) algorithm is used for optimally mapping the available gene-expression data into a sequence of Boolean states, and then the BIRL method is efficiently combined with the Q-learning algorithm for quantification of the immediate cost function. The performance of the proposed methodology is investigated by applying a state-feedback controller to two GRN models: a melanoma WNT5A Boolean network and a p53-MDM2 negative feedback loop Boolean network, when the cost of the undesirable states, and thus the identity of the undesirable genes, is learned using the proposed methodology.**

***Index Terms*—Gene Regulatory Networks, Partially-Observed Boolean Dynamical System, Boolean Kalman Smoother, Bayesian Inverse Reinforcement Learning, Q-Learning, Melanoma, p53-MDM2.**

I. Introduction

Developing therapeutic intervention strategies for control of gene regulatory networks (GRNs) is a problem of great current interest in system biology. The interventions are usually taken to avoid undesirable states associated with cancer, such as cell proliferation and metastasis-implicated states. Several mathematical models have been developed for modeling GRNs, such as Boolean networks [1]–[3], ordinary differential equations (OED) [4], [5], S-systems [6], [7], and Bayesian networks [8], [9]. Methods for inference of gene regulatory networks for discovery of cellular identity and their functionalities have been proposed in [10]–[17].

Boolean networks, in particular, have been widely used for studying GRNs [18]–[21]. In the Boolean network model, the transcriptional state of each gene is represented by 0 (OFF) or 1 (ON), and the relationship among genes is described by logical gates updated at discrete time intervals [22]. Boolean networks were first introduced as a completely-observable, deterministic model by Kauffman and collaborators [1]. Several variations of the original Boolean network model have been introduced in the literature to account for the stochasticity in the behavior of gene regulatory networks. These models include Random Boolean Networks [1], Boolean Networks with perturbation (BNp) [23], Probabilistic Boolean Networks (PBN) [2], and Boolean Control Networks (BCN) [24]. All aforementioned methods assume that the transcriptional states of genes are directly observable. However, in practice, the Boolean gene states are never observed directly, but only indirectly and incompletely through noisy measurements based on expression technologies such as cDNA microarrays [25], RNA-Seq [26], and cell imaging-based assays [27].

The partially-observed Boolean dynamical systems (POBDS) signal model [3], [28] generalizes and unifies the existing Boolean network models by allowing stochasticity in both state and measurement processes. Several tools have been developed in recent years for this signal model, including the optimal minimum mean-square error (MMSE) state estimators called the Boolean Kalman filter (BKF) [3], [28] and Boolean Kalman smoother (BKS) [29], particle filters for state and parameter estimation [30], adaptive filters for simultaneous estimation of state and parameters of POBDS [3], network inference [31], sensor selection [32] and optimal filter with correlated observation noise [33]. Most of these tools are freely available through an open-source R package called "BoolFilter" [34], [35].

Several strategies have been developed for control of gene regulatory networks [14], [23], [24], [36]–[38]. Infinite-horizon controllers for the POBDS model under various constraints have been introduced in [39]–[42]. All aforementioned methods assume that the undesirable conditions (e.g., identity of genes associated with tumor growth) and the cost of intervention (e.g., the severity of side effects and amount of financial cost) are fully known. For instance, in the case of the melanoma regulatory network discussed in Section IV, this information includes the undesirability of activation of the WNT5A gene, which has been known to be associated with metastasis. Given this information, which is referred to as *immediate cost function*, one seeks to obtain an intervention strategy to avoid states where WNT5A is upregulated. However, in practice, the immediate cost function might be unknown or partially-known, and one needs to estimate this cost function before deriving an intervention strategy. In this paper, we consider a realistic scenario in which a sequence of interventions performed by an expert (e.g., a physician or biologist) and associated gene-expression data is available. This sequence is assumed to convey the near-optimal behavior of the expert in an experimental setting. Given this information, the objective is estimating the immediate cost function

M. Imani and U.M. Braga-Neto are with the Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: m.imani88@tamu.edu,ulisses@ece.tamu.edu)

reflecting the undesirability of transcriptional states vis-à-vis the cost of the intervention itself.

In this paper, we employ *inverse reinforcement learning* (IRL) [44] to achieve that objective. There exist many variations of basic IRL [45]–[48]. Our proposed methodology is based on Bayesian inverse reinforcement learning (BIRL), due to its flexibility and accuracy [45]. However, there are two main issues that need to be overcome in order to apply BIRL for finding the immediate cost function of partially-observed GRNs:

- The transcriptional states of the genes are not observed directly, but only indirectly through noisy gene-expression data.
- BIRL requires multiple applications of computationally-expensive dynamical programming algorithms, such as value iteration or policy iteration [49], to large state spaces common in GRNs.

To overcome the first issue, the Boolean Kalman smoother (BKS) [29] is used for optimally mapping the available gene-expression data into a Boolean state trajectory to be used by the BIRL. To address the computational issue, we proposed the use of the Q-learning algorithm [50] as an approximation of the optimal dynamic programming solution. We demonstrate the application of the proposed method with numerical experiments using synthetic expert interventions given a sequence of transcriptional data generated from two GRN models: a melanoma WNT5A Boolean network [37] and a p53-MDM2 negative feedback loop Boolean network [51]. The immediate cost function is learned under two different partial knowledge cases, and the effectiveness of the proposed methodology is demonstrated by using it in the design of a simple state-feedback controller to shift the dynamics of the network away from states associated with metastasis.

The article is organized as follows. In Section II, the infinite-horizon control problem and the POBDS state and observational models are briefly introduced, after which the Boolean Kalman smoother (BKS) algorithm is reviewed. In Section III, the proposed Bayesian inverse reinforcement learning for quantification of the immediate cost function is presented in detail. Numerical experiments using the melanoma gene regulatory network are reported and discussed in Section IV. Finally, Section V contains concluding remarks.

II. Mathematical Preliminaries

A. Infinite-Horizon Control

A gene regulatory network containing d genes is described by a *state process* $\{\mathbf{X}_k; k = 0, 1, \ldots\}$, where $\mathbf{X}_k \in \{0,1\}^d$ represents the activation/inactivation state of the genes at time k. The state of the genes is affected by a sequence of *control inputs* $\{\mathbf{u}_k; k = 0, 1, \ldots\}$, where $\mathbf{u}_k$ takes values in a finite set $\mathbb{U}$. Let $c(\mathbf{X}_k, \mathbf{u}_k)$ be a bounded *immediate cost* of applying control input $\mathbf{u}_k$ when the state of the system is $\mathbf{X}_k$. The infinite-horizon cost associated with an initial state $\mathbf{x} \in \{0,1\}^d$ is defined as:

$$J^{\pi}(\mathbf{x}) \;=\; E\left[\sum_{k=0}^{\infty} \gamma^k \, c\left(\mathbf{X}_k, \pi(\mathbf{X}_k)\right) \;\middle|\; \mathbf{X}_0 = \mathbf{x}\right], \qquad (1)$$

where $\pi : \{0,1\}^d \to \mathbb{U}$ is a (stationary) *control policy* that prescribes a control input for each Boolean state, Π is the space of all possible control policies, and the discount factor γ places a premium on minimizing the cost of early interventions as opposed to later ones, which is sensible from a medical perspective [36]. Given an initial state $\mathbf{x} \in \{0,1\}^d$, the goal of infinite-horizon control is to find an optimal policy π^* such that $J^*(\mathbf{x}) \stackrel{\text{def}}{=} J^{\pi^*}(\mathbf{x}) \leq J^{\pi}(\mathbf{x})$, for all $\pi \in \Pi$.

According to the theory of dynamic programming [49], the optimal value function satisfies the following Bellman equation:

$$J^*(\mathbf{x}) \;=\; \min_{\mathbf{u}\in\mathbb{U}} \left[c(\mathbf{x},\mathbf{u}) + \gamma \, E_{\mathbf{x}'|\mathbf{x},\mathbf{u}}\left[J^*(\mathbf{x}')\right]\right], \qquad (2)$$

where the expectation $E_{\mathbf{x}'|\mathbf{x},\mathbf{u}}$ is taken over all successor Boolean states $\mathbf{x}' \in \{0,1\}^d$ if the current state is $\mathbf{x} \in \{0,1\}^d$ and control input $\mathbf{u} \in \mathbb{U}$ is taken.

An equivalent convenient way of representing the cost function under policy π is to use the joint Boolean state and intervention spaces:

$$Q^{\pi}(\mathbf{x},\mathbf{u}) \;=\; E\left[c(\mathbf{X}_0,\mathbf{u}_0) + \sum_{r=1}^{\infty} \gamma^r \, c\left(\mathbf{X}_r, \pi(\mathbf{X}_r)\right) \;\middle|\; \mathbf{X}_0 = \mathbf{x}, \mathbf{u}_0 = \mathbf{u}\right], \qquad (3)$$

for $\mathbf{x} \in \{0,1\}^d$ and $\mathbf{u} \in \mathbb{U}$. The Q-function $Q^{\pi}(\mathbf{x},\mathbf{u})$ is the expected return when starting from state $\mathbf{x}$, applying $\mathbf{u}$, and following π thereafter. This cost function satisfies the following Bellman equation:

$$Q^{\pi}(\mathbf{x},\mathbf{u}) \;=\; c(\mathbf{x},\mathbf{u}) + \gamma \, E_{\mathbf{x}'|\mathbf{x},\mathbf{u}}\left[Q^{\pi}(\mathbf{x}', \pi(\mathbf{x}'))\right], \qquad (4)$$

where $E_{\mathbf{x}'|\mathbf{x},\mathbf{u}}$ and $\mathbf{x}'$ are as before.

The optimal Q-function can be computed by searching over the set of all possible policies Π as:

$$Q^*(\mathbf{x},\mathbf{u}) \;=\; \min_{\pi\in\Pi} Q^{\pi}(\mathbf{x},\mathbf{u}), \qquad (5)$$

which will lead to the following optimal policy:

$$\pi^*(\mathbf{x}) \;=\; \operatorname*{argmin}_{\mathbf{u}\in\mathbb{U}} Q^*(\mathbf{x},\mathbf{u}), \qquad (6)$$

for $\mathbf{x} \in \{0,1\}^d$.

B. POBDS State Model

The state process $\{\mathbf{X}_k; k = 0, 1, \ldots\}$ is assumed to satisfy the following nonlinear first-order Markov signal model:

$$\mathbf{X}_k \;=\; \mathbf{f}\left(\mathbf{X}_{k-1}\right) \oplus \mathbf{u}_{k-1} \oplus \mathbf{n}_k, \qquad (7)$$

for $k = 1, 2, \ldots$, where $\mathbf{f} : \{0,1\}^d \times \mathbb{U} \to \{0,1\}^d$ is a Boolean function, called the *network function*, $\mathbf{n}_k \in \{0,1\}^d$ is Boolean transition noise, and "$\oplus$" indicates componentwise modulo-2 addition, which acts as a componentwise XOR operator. For example,

$$\begin{bmatrix}0\\0\\1\end{bmatrix} \oplus \begin{bmatrix}0\\1\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\end{bmatrix}. \qquad (8)$$

Hence, the manner in which the control input influences state evolution is that if $\mathbf{u}_{k-1}(i)$ is one, it flips the value

of the ith bit of the Boolean state $\mathbf{X}_k$. In practice, control would be accomplished by means of drugs targeted at those genes. The noise process $\{\mathbf{n}_k; k = 1, 2, ...\}$ is assumed to be "white" in the sense that the noises at distinct time points are independent random variables. We also assume that the noise process is independent of the initial state $\mathbf{X}_0$. We assume that the components of the vector $\mathbf{n}_k$ are i.i.d. (the general non i.i.d. case can be similarly handled, at the expense of introducing more parameters), with $P(\mathbf{n}_k(i) = 1) = p$, for $i = 1, ..., d$. Parameter $0 \leq p \leq 1/2$ corresponds to the amount of "perturbation" to the Boolean state process; the case $p = 1/2$ corresponds to maximum uncertainty.

C. POBDS Observation Model

The states are observed indirectly through noisy gene-expression data. The latter constitute the observational layer of the POBDS model. In this paper, we assume a Gaussian observational model, which is an appropriate model for many important gene-expression measurement technologies, such as cDNA microarrays [25] and live cell imaging-based assays [27], in which transcript measurements are continuous and unimodal (within a single population of interest).

Let $\mathbf{Y}_k = (\mathbf{Y}_k(1), \ldots, \mathbf{Y}_k(d))$, where $\mathbf{Y}_k(j)$ is the abundance measurement corresponding to transcript j, for $j = 1, \ldots, d$, at time k, for $k = 1, 2, \ldots$ We assume conditional independence of the transcript measurements given the state:

$$p(\mathbf{Y}_k = \mathbf{y} \mid \mathbf{X}_k = \mathbf{x}) = \prod_{j=1}^{d} p\left(\mathbf{Y}_k(j) = \mathbf{y}(j) \mid \mathbf{X}_k(j) = \mathbf{x}(j)\right), \tag{9}$$

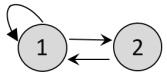
and adopt the Gaussian model:

$$p(\mathbf{Y}_k(j) = \mathbf{y}(j) \mid \mathbf{X}_k(j) = \mathbf{x}(j)) = \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp\left(-\frac{(\mathbf{y}(j) - m_j)^2}{2\,\sigma_j^2}\right), \tag{10}$$

where m_j and $\sigma_j > 0$ are the mean and standard deviation of abundance of transcript j, respectively, for $j = 1, \ldots, d$, such that

$$\begin{aligned} m_j &= m_{j,0}\,(1 - \mathbf{x}(j)) + m_{j,1}\,\mathbf{x}(j), \\ \sigma_j &= \sigma_{j,0}\,(1 - \mathbf{x}(j)) + \sigma_{j,1}\,\mathbf{x}(j), \end{aligned} \tag{11}$$

where the parameters $(m_{j,0}, \sigma_{j,0} > 0)$ and $(m_{j,1}, \sigma_{j,1} > 0)$ are the means and standard deviations of the abundance of transcript j in the inactivated and activated states, respectively.



$$\mathbf{f}(\mathbf{X}_{k-1}) = \begin{bmatrix} \mathbf{X}_{k-1}(1) \text{ AND } \mathbf{X}_{k-1}(2) \\ \mathbf{X}_{k-1}(2) \end{bmatrix}$$

Fig. 1: Simple system with 2 genes.

D. POBDS Example

In this section we present a simple example of a POBDS model and cost structure, which illustrates the concepts introduced in the previous sections. Fig 1 depicts the state model for a simple GRN containing 2 genes. In this case, there are 4 possible Boolean states: $\mathbf{x}^1 = (0,0)^T, \mathbf{x}^2 = (0,1)^T, \mathbf{x}^3 = (1,0)^T, \mathbf{x}^4 = (1,1)^T$. The Boolean state $\mathbf{X}_k$ at time k can assume any of these 4 values. It is obtained from the previous state $\mathbf{X}_{k-1}$ (in the absence of an external input $\mathbf{u}_k$) as follows: $\mathbf{X}_k(1)$ is equal to $\mathbf{X}_{k-1}(1)$ AND $\mathbf{X}_{k-1}(2)$ with probability $1 - p$, or its complement, with probability p, while $\mathbf{X}_k(2)$ is equal to $\mathbf{X}_{k-1}(2)$ with probability $1 - p$, or its complement, with probability p. The observation on the state are given by $\mathbf{Y}_k(j) \sim \mathcal{N}(m_{j,0}\,(1 - \mathbf{X}_k(j)) + m_{j,1}\,\mathbf{X}_k(j), \sigma_{j,0}^2\,(1 - \mathbf{X}_k(j)) + \sigma_{j,1}^2\,\mathbf{X}_k(j))$, for $j = 1, 2, 3, 4$; where $\mathcal{N}(\mu, \sigma^2)$ represents a Gaussian distribution with mean μ and variance σ^2. The control input flips the state value of a gene in the next time point. For instance, to flip the value of first gene at time step k, one needs to apply the control input $\mathbf{u}_{k-1} = (1,0)^T$. If we assume that at most one gene can have its value flipped at each time point, the space of control is $\mathbb{U} = \{\mathbf{u}^1 = (1,0)^T, \mathbf{u}^2 = (0,1)^T, \mathbf{u}^3 = (0,0)^T\}$. Assume that the activation of the first gene is undesirable, with a unit cost incurred whenever it is active. Assuming further that the cost of applying any control input is zero, the immediate cost function in this case is given by $c(\mathbf{x}^1, \mathbf{u}) = 0, c(\mathbf{x}^2, \mathbf{u}) = 0, c(\mathbf{x}^3, \mathbf{u}) = 1, c(\mathbf{x}^4, \mathbf{u}) = 1$, for any $\mathbf{u} \in \mathbb{U}$. It is easy to show that, for $p < 0.5$, the optimal stationary policy for the various states is as follows: $\pi^*(\mathbf{x}^1) = \mathbf{u}^3, \pi^*(\mathbf{x}^2) = \mathbf{u}^3, \pi^*(\mathbf{x}^3) = \mathbf{u}^3, \pi^*(\mathbf{x}^4) = \mathbf{u}^2$. In other words, the optimal control input when the state of the system is $\mathbf{x}^4$ is $\mathbf{u}^2$, and for all other cases, it is $\mathbf{u}^3$.

E. Boolean Kalman Smoother

Given a sequence of control inputs $\mathbf{u}_{0:T}$ and measurements $\mathbf{Y}_{1:T}$, the optimal (fixed-point) smoother is an estimator $\hat{\mathbf{X}}_{k|T}$ of the state $\mathbf{X}_k$ that minimizes the mean-square error (MSE):

$$\text{MSE}(\hat{\mathbf{X}}_{k|T} \,|\mathbf{u}_{0:T}, \mathbf{Y}_{1:T}) = E\left[||\hat{\mathbf{X}}_{k|T} - \mathbf{X}_k||_2^2 \mid \mathbf{u}_{0:T}, \mathbf{Y}_{1:T}\right], \tag{12}$$

where $||.||_2$ is the usual L_2 vector norm. It has been shown that the optimal smoother is given by [3]:

$$\hat{\mathbf{X}}_{k|T}^{\text{MS}} = \overline{E\left[\mathbf{X}_k \mid \mathbf{u}_{0:T}, \mathbf{Y}_{1:T}\right]}, \tag{13}$$

where $\overline{\mathbf{v}}(i) = I_{\mathbf{v}(i) > 1/2}$ for $i = 1, \ldots, d$. This estimator is called the Boolean Kalman smoother (BKS) [3], [29]. We briefly outline an iterative algorithm for exact computation of the BKS below.

Let $(\mathbf{x}^1, \ldots, \mathbf{x}^{2^d})$ be an arbitrary enumeration of the possible state vectors, and define the state conditional probability distribution vectors $\mathbf{\Pi}_{k|r}$ and $\mathbf{\Delta}_{k|r}$ of length 2^d via:

$$\begin{aligned} \mathbf{\Pi}_{k|r}(i) &= P\left(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{Y}_{1:r}, \mathbf{u}_{0:r-1}\right), \\ \mathbf{\Delta}_{k|r}(i) &= p\left(\mathbf{Y}_{r+1:T} \mid \mathbf{X}_k = \mathbf{x}^i, \mathbf{u}_{r+1:T-1}\right), \end{aligned} \tag{14}$$

for $i = 1, ..., 2^d$, $r = 1, 2, ..., T$, $r \leq k$.

Let the *controlled transition matrix* be a matrix of size $2^d \times 2^d$ given by:

$$\begin{aligned}(M_k(\mathbf{u}))_{ij} &= P(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{X}_{k-1} = \mathbf{x}^j, \mathbf{u}_{k-1} = \mathbf{u}) \\ &= P\left(\mathbf{n}_k = \mathbf{f}(\mathbf{x}^j) \oplus \mathbf{u} \oplus \mathbf{x}^i\right) \\ &= p^{||\mathbf{f}(\mathbf{x}^j) \oplus \mathbf{u} \oplus \mathbf{x}^i||_1} (1-p)^{d-||\mathbf{f}(\mathbf{x}^j) \oplus \mathbf{u} \oplus \mathbf{x}^i||_1},\end{aligned} \tag{15}$$

for $i, j = 1, \ldots, 2^d$ and a control input vector $\mathbf{u} \in \mathbb{U}$.

Additionally, let the *update matrix* be a diagonal matrix of size $2^d \times 2^d$ with diagonal elements given by:

$$\begin{aligned}(T_k(\mathbf{Y}_k))_{ii} &= p\left(\mathbf{Y}_k \mid \mathbf{X}_k = \mathbf{x}^i\right) \\ &= \prod_{j=1}^{d} \left(\frac{1}{\sqrt{2\pi \left(\sigma_{j,0}\left(1 - \mathbf{x}^i(j)\right) + \sigma_{j,1}\,\mathbf{x}^i(j)\right)^2}} \right) \\ &\times \exp\left(-\sum_{j=1}^{d} \frac{\left(\mathbf{Y}_k(j) - m_{j,0}\left(1 - \mathbf{x}^i(j)\right) - m_{j,1}\,\mathbf{x}^i(j)\right)^2}{2\left(\sigma_{j,0}\left(1 - \mathbf{x}^i(j)\right) + \sigma_{j,1}\,\mathbf{x}^i(j)\right)^2} \right),\end{aligned} \tag{16}$$

for $i = 1, \ldots, 2^d$ and an observation vector $\mathbf{y} \in \mathbb{R}^d$.

Finally, let A be a matrix of size $d \times 2^d$ containing all Boolean states of the system as:

$$A = \left[\mathbf{x}^1 \cdots \mathbf{x}^{2^d}\right]. \tag{17}$$

It is easy to verify that $E\left[\mathbf{X}_k \mid \mathbf{u}_{0:T}, \mathbf{Y}_{1:T}\right] = A\mathbf{\Pi}_{k|T}$, so it follows from (13) that:

$$\hat{\mathbf{X}}_{k|T}^{\mathrm{MS}} = \overline{A\mathbf{\Pi}_{k|T}}\,. \tag{18}$$

The complete procedure for computation of the BKS is given in Algorithm 1. Notice that "$\circ$" denotes the "Hadamard" product.

Algorithm 1 BKS $(\mathbf{u}_{0:T}, \mathbf{Y}_{1:T})$

Forward Probabilities:

Initialization: $(\mathbf{\Pi}_{0|0})_i = P\left(\mathbf{X}_0 = \mathbf{x}^i\right)$, for $i = 1, \ldots, 2^d$.
for $k = 1, \ldots, T$ **do**
  Prediction: $\mathbf{\Pi}_{k|k-1} = M_k(\mathbf{u}_{k-1})\,\mathbf{\Pi}_{k-1|k-1}$.
  Update: $\boldsymbol{\beta}_k = T_k(\mathbf{Y}_k)\,\mathbf{\Pi}_{k|k-1}$.
  Normalization: $\mathbf{\Pi}_{k|k} = \boldsymbol{\beta}_k / ||\boldsymbol{\beta}_k||_1$.
end for

Backward Probabilities:

$\mathbf{\Delta}_{T|T} = \mathbf{1}_{2^d}$.
for $k = T, T-1, \ldots, 1$ **do**
  Update: $\mathbf{\Delta}_{k|k-1} = T_k(\mathbf{Y}_k)\,\mathbf{\Delta}_{k|k}$.
  Prediction: $\mathbf{\Delta}_{k-1|k-1} = M_k(\mathbf{u}_{k-1})^T\,\mathbf{\Delta}_{k|k-1}$.
end for

Optimal State Estimation:

Initial Smoothed Posterior Distribution:

$$\mathbf{\Pi}_{0|T} = (\mathbf{\Pi}_{0|0} \circ \mathbf{\Delta}_{0|0}) / ||\mathbf{\Pi}_{0|0} \circ \mathbf{\Delta}_{0|0}||_1\,.$$

Initial Smoothed Estimate of State: $\hat{\mathbf{X}}_{0|T}^{\mathrm{MS}} = \overline{A\mathbf{\Pi}_{0|T}}$.
for $k = 1, \ldots, T$ **do**
  Smoothed Posterior Distribution:

$$\mathbf{\Pi}_{k|T} = (\mathbf{\Pi}_{k|k-1} \circ \mathbf{\Delta}_{k|k-1}) / ||\mathbf{\Pi}_{k|k-1} \circ \mathbf{\Delta}_{k|k-1}||_1\,.$$

  Smoothed Estimate of State: $\hat{\mathbf{X}}_{k|T}^{\mathrm{MS}} = \overline{A\mathbf{\Pi}_{k|T}}$.
end for

Return $(\hat{\mathbf{X}}_{0:T|T}^{\mathrm{MS}})$

III. Bayesian Inverse Reinforcement Learning for Quantification of the Immediate Cost Function

In this paper, we consider the realistic case where the immediate cost function $c(\mathbf{x}, \mathbf{u})$ is unknown or partially-known, and must be learned from a sequence of noisy gene-expression measurements and control inputs taken by an expert:

$$D = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{Y}}_{1:T}\}\,. \tag{19}$$

We assume the expert to be imperfect, which means we are provided with a noisy sample of the ideal expert's policy; therefore, both the sample inputs and observations are random variables (we do not capitalize $\tilde{\mathbf{u}}_{0:T}$ for conformity with the notation for ordinary control inputs).

The main components of the proposed methodology are described next.

A. Bayesian Inverse Reinforcement Learning

The Inverse Reinforcement Learning (IRL) method was introduced in [44], followed by several variations of it [45]–[48]. Due to the availability of prior biological knowledge, we employ the Bayesian version of IRL proposed in [45].

Let the uncertainty in the immediate cost function be represented in a parametric form as $c_\theta(\mathbf{x}, \mathbf{u})$, where θ is a vector of parameters in an arbitrary space Θ. It should be noted that this parametric representation does not impose any limitation on the form of the immediate cost function. For instance, if no prior information on the immediate cost function exists, θ will be a vector of size $2^d \times |\mathbb{U}|$, the elements of which are the costs for all possible states and control inputs.

The optimal Q-function in (5) under the immediate cost function specified by θ will be denoted by $Q_\theta^*(\mathbf{x}, \mathbf{u})$. Assuming the Boltzmann softmax policy [52], we have

$$P(\mathbf{u} \mid \mathbf{x}, \theta) \propto \exp\left(-\eta\, Q_\theta^*(\mathbf{x}, \mathbf{u})\right), \tag{20}$$

for $\mathbf{x} \in \{0,1\}^d$ and $\mathbf{u} \in \mathbb{U}$; where $\eta > 0$ represents our confidence on the expert's decision. The smaller the value of η, the more "imperfect" the expert is expected to be.

Now, applying the Boolean Kalman smoother (BKS) in Algorithm 1 to the expert's sequence in (19) produces a sequence of estimated Boolean state vectors:

$$D = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{Y}}_{1:T}\} \xrightarrow{\mathrm{BKS}} \tilde{D} = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{X}}_{0:T}\}\,. \tag{21}$$

Assuming 1) a first-order Markovian structure for $\tilde{\mathbf{X}}_{0:T}$ (as in the POBDS model) and 2) that the input control $\tilde{\mathbf{u}}_k$ depends

only on $\tilde{\mathbf{X}}_k$ and not on previous states, for $k = 0, ..., T$; then it is easy to verify that the data joint likelihood is given by:

$$\begin{aligned} P(\tilde{D} \mid \theta) \;&=\; P(\tilde{\mathbf{X}}_{0:T}) \prod_{r=0}^{T} P(\tilde{\mathbf{u}}_r \mid \tilde{\mathbf{X}}_r, \theta) \\ &=\; P(\tilde{\mathbf{X}}_{0:T}) \prod_{r=0}^{T} \frac{\exp\Big(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_r, \tilde{\mathbf{u}}_r)\Big)}{\sum_{\mathbf{u}' \in \mathbb{U}} \exp\Big(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_r, \mathbf{u}')\Big)}\,. \end{aligned} \tag{22}$$

We adopt the posterior mean $\hat{\theta} = E[\theta \mid \tilde{D}]$ as the estimator of the parameter vector θ. It is shown in [45] that this provides the best estimator in the sense of closeness of the estimated immediate cost function and policy to the expert's immediate cost function and policy. Due to the Bayesian nature of the estimation process and the fact that the closed-form expression for computation of the likelihood in (22) does not exist, we approximate $E[\theta \mid \tilde{D}]$ using a simple Markov chain Monte Carlo (MCMC) technique [45].

Given the current parameter vector θ_r in the rth iteration of the MCMC method, a candidate $\tilde{\theta}$ for the next iteration is chosen from the neighbors of θ_r, in such a way that $\tilde{\theta}$ differs from θ_r only in a randomly selected element i:

$$\tilde{\theta}(i) \;=\; \begin{cases} \theta_r(i) \,+\, \delta_i & \text{with prob. } 0.5\,, \\ \theta_r(i) \,-\, \delta_i & \text{with prob. } 0.5\,, \end{cases} \tag{23}$$

where $\delta_i > 0$ is the search step-length for the ith element of θ, and $\tilde{\theta}(j) = \theta(j)$ for $j \neq i$.

The next MCMC sample is found using the Metropolis-Hastings step:

$$\theta_{r+1} \;=\; \begin{cases} \tilde{\theta} & \text{w.p. } \min\Big\{1, \frac{P(\tilde{D}|\tilde{\theta})\, P(\tilde{\theta})}{P(\tilde{D}|\theta_r)\, P(\theta_r)}\Big\}\,, \\ \theta_r & \text{o.w.} \end{cases} \tag{24}$$

After discarding the first M samples, the posterior mean is approximated as:

$$\hat{\theta} \;\approx\; \frac{1}{N} \sum_{i=M+1}^{M+N} \theta_i\,. \tag{25}$$

For a good approximation, both M and N need to be sufficiently large, and computation of each MCMC sample requires dynamic programming for evaluation of the optimal Q-function in (22). This makes the MCMC process intractable. In the next section, a procedure to address this using the Q-learning algorithm is described.

B. Q-learning

Q-learning [50] is a model-free reinforcement learning technique, which is used here to speed up the computation of the MCMC step in the previous section by obtaining an approximation of the optimal Q-function. The latter is learned based on the simulated data generated from the state model: assume that in the nth step of the Q-learning algorithm, the system is in state $\mathbf{x}$, and by applying a control input $\mathbf{u}$, the system moves to state $\mathbf{x}'$ in step $n+1$. All elements of Q-function at time $n+1$ will be the same as time n, except the element corresponding to state $\mathbf{x}$ and control input $\mathbf{u}$ which is updated as:

$$\begin{aligned} Q^{n+1}_\theta(\mathbf{x}, \mathbf{u}) \;&=\; (1 - \alpha_n(\mathbf{x}, \mathbf{u}))\; Q^n_\theta(\mathbf{x}, \mathbf{u}) \\ &\quad +\; \alpha_n(j, \mathbf{u}) \left(c_\theta(\mathbf{x}, \mathbf{u}) + \gamma \min_{\mathbf{u}' \in \mathbb{U}} Q^n_\theta(\mathbf{x}', \mathbf{u}') \right), \end{aligned} \tag{26}$$

where $0 \leq \alpha_n(\mathbf{x}, \mathbf{u}) < 1$ is the learning rate for state $\mathbf{x}$ at episode n when control input $\mathbf{u}$ is applied to the system. It has been shown that for proper choices of the learning rate, the solution of the Q-learning method converges to the optimal solution Q^*_θ as $n \to \infty$ [50].

Several methods to select the learning rate for Q-learning have been proposed in literature. Here, we follow [38], in which the learning rate is assumed to be a function of a parameter $0 < C < 1$ and the number of visits to each state and control input during the learning process. The complete procedure is presented in Algorithm 2. The only parameters which need to be set are the maximum number of iterations $n^{\max}$ and the parameter C.

Algorithm 2 Q-learning (θ)

Initialization Step: Set $0 < C < 1$, $n^{\max}$, and initialize $Q_\theta(\mathbf{x}^j, \mathbf{u}) = 0$, $v(\mathbf{x}^j, \mathbf{u}) = 0$, for $j = 1, ..., 2^d$ and $\mathbf{u} \in \mathbb{U}$.
Select an arbitrary initial state $\mathbf{x}^j$.
for $n = 0, ..., n^{\max}$ **do**
  Select randomly $\mathbf{u}$ from $\mathbb{U}$.
  Find successor state: $\mathbf{x}^i = \mathbf{f}(\mathbf{x}^j) \oplus \mathbf{u} \oplus \mathbf{n}_n$.
  Update visiting counter: $v(\mathbf{x}^j, \mathbf{u}) = v(\mathbf{x}^j, \mathbf{u}) + 1$.
  Update learning rate: $\alpha = \frac{C}{v(\mathbf{x}^j, \mathbf{u})}$.
  Update $Q_\theta(\mathbf{x}^j, \mathbf{u})$:

$$\begin{aligned} Q_\theta(\mathbf{x}^j, \mathbf{u}) \;&=\; (1 - \alpha)\; Q_\theta(\mathbf{x}^j, \mathbf{u}) \\ &\quad +\; \alpha \left(c_\theta(\mathbf{x}^j, \mathbf{u}) + \gamma \min_{\mathbf{u}' \in \mathbb{U}} Q_\theta(\mathbf{x}^i, \mathbf{u}') \right) \end{aligned}$$

  $j \leftarrow i$.
end for
Return $(Q_\theta(\mathbf{x}^i, \mathbf{u}), i = 1, ..., 2^d, \mathbf{u} \in \mathbb{U})$

C. The Proposed Algorithm

A schematic diagram of the proposed Bayesian inverse reinforcement learning (BIRL) method is presented in Fig. 2. First, the noisy expert's sequence $D = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{Y}}_{1:T}\}$ is mapped to a sequence $\tilde{D} = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{X}}_{0:T}\}$ using the Boolean Kalman smoother (BKS). Then, the MCMC iteration starts by first drawing an initial sample from the prior distribution, $\theta_0 \sim p(\theta)$, the Q-learning algorithm is run for approximation of the optimal Q-function $Q^*_{\theta_0}$, and the unnormalized posterior $p_0 = P(\tilde{D} \mid \theta_0)\, p(\theta_0)$ is computed. The iteration proceeds as follows: at the $(r+1)$th step, the next candidate parameter vector $\tilde{\theta}$ is generated randomly around the previous sample θ_r using the process presented in (23). The unnormalized posterior probability $p^{\tilde{\theta}}$ is computed using the results of the Q-learning algorithm tuned to $\tilde{\theta}$. Finally, the unnormalized posterior probability for $\tilde{\theta}$ and the previous unnormalized

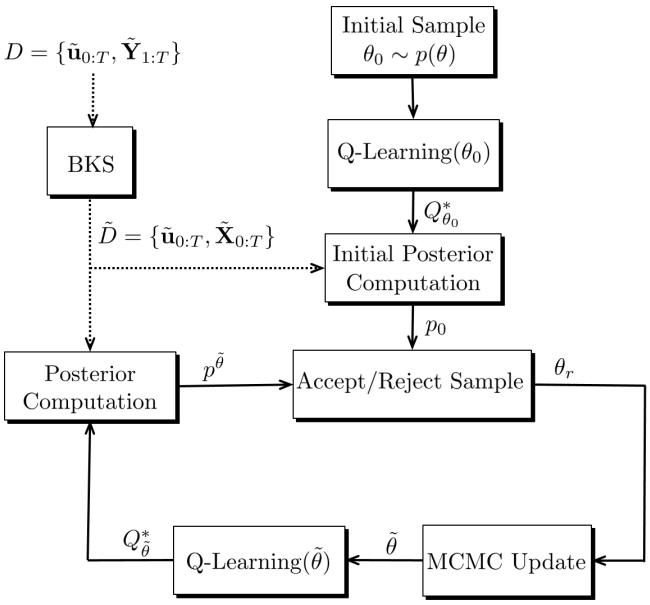


Fig. 2: The schematic diagram of the proposed Bayesian inverse reinforcement learning (BIRL) for quantification of the immediate cost function.

posterior probability specify the next MCMC sample based on (24). The process continues for $N + M$ steps, where the first M steps yield the discarded burn-in samples, and the parameter estimator $\hat{\theta}$ is computed by the sample mean in (25). The complete procedure is displayed in Algorithm 3.

IV. Numerical Experiments

In this section, the performance of the proposed methodology is investigated by applying a state-feedback controller to two important Boolean models of GRNs: a Melanoma WNT5A Boolean network [37] and a p53-MDM2 negative feedback loop Boolean network [51]. To the best of authors' knowledge, this paper is the first paper discussing estimation of the immediate cost function for GRNs and as a result no comparison have been made in the numerical experiments.

A. Performance Evaluation Based on a Melanoma Network

In this section, we demonstrate the performance of the proposed Bayesian inverse reinforcement learning methodology by simulating the expert's sequence $D = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{Y}}_{1:T}\}$ from a Boolean model for a gene regulatory network implicated in metastatic melanoma [37] and learning the immediate cost function under two different partial knowledge cases. We also demonstrate the effectiveness of the proposed methodology by using it in the design of a simple state-feedback controller to shift the dynamics of the network away from states associated with metastasis.

The network contains 7 genes: WNT5A, pirin, S100P, RET1, MART1, HADHB and STC2. The regulatory relationship for this network is presented in Table I. The ith output binary string specifies the output value for the ith input gene(s) in binary representation. For example, the last

Algorithm 3 BIRL: Bayesian Inverse Reinforcement Learning for Quantification of the Immediate Cost Function

1: Set step-length parameters δ_i, for $i = 1, ..., |\theta|$.

Sequence Mapping:

2: Map the noisy expert's sequence $D = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{Y}}_{1:T}\}$ to a sequence $\tilde{D} = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{X}}_{0:T}\}$ using the Boolean Kalman smoother (Algorithm 1):

$$D = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{Y}}_{1:T}\} \xrightarrow{\text{BKS}} \tilde{D} = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{X}}_{0:T}\}.$$

Initial Sampling:

3: Draw a sample from prior $\theta_0 \sim p(\theta)$.

Q-Learning:

4: Run Q-learning tuned to the initial parameter vector θ_0:

$$Q^*_{\theta_0} \leftarrow \text{Q-Learning}(\theta_0)$$

Posterior Computation:

5: Find the initial unnormalized posterior probability:

$$p_0 = p(\theta_0) \times \prod_{r=0}^{T} \frac{\exp\left(-\eta\, Q^*_{\theta_0}(\tilde{\mathbf{X}}_r, \tilde{\mathbf{u}}_r)\right)}{\sum_{\mathbf{u}' \in \mathbb{U}} \exp\left(-\eta\, Q^*_{\theta_0}(\tilde{\mathbf{X}}_r, \mathbf{u}')\right)}.$$

MCMC Process:

6: **for** $i = 1, ..., N + M$ **do**

7: Select a random index $l \in \{1, ..., |\theta|\}$.

8: Set $\tilde{\theta}(l) = \begin{cases} \theta_{i-1}(l) + \delta_l & \text{w.p. } 0.5 \\ \theta_{i-1}(l) - \delta_l & \text{o.w.} \end{cases}$, $\tilde{\theta}(j) = \theta_{i-1}(j),\ j \neq l.$

9: Run Q-learning tuned to the candidate of parameter $\tilde{\theta}$:

$$Q^*_{\tilde{\theta}} \leftarrow \text{Q-Learning}(\tilde{\theta})$$

10: Find the unnormalized posterior probability of parameter $\tilde{\theta}$:

$$p^{\tilde{\theta}} = p(\tilde{\theta}) \times \prod_{r=0}^{T} \frac{\exp\left(-\eta\, Q^*_{\tilde{\theta}}(\tilde{\mathbf{X}}_r, \tilde{\mathbf{u}}_r)\right)}{\sum_{\mathbf{u}' \in \mathbb{U}} \exp\left(-\eta\, Q^*_{\tilde{\theta}}(\tilde{\mathbf{X}}_r, \mathbf{u}')\right)}.$$

11: Set $(\theta_i, p_i) = \begin{cases} (\tilde{\theta}, p^{\tilde{\theta}}) & \text{w.p. } \min\left\{1, \frac{p^{\tilde{\theta}}}{p_{i-1}}\right\} \\ (\theta_{i-1}, p_{i-1}) & \text{o.w.} \end{cases}$

12: **end for**

Immediate Cost Function Estimation:

13: The BIRL estimate of parameter:

$$\hat{\theta} = \frac{1}{N} \sum_{i=M+1}^{M+N} \theta_i.$$

row of Table I specifies the value of STC2 at current time step k from different pairs of (pirin, STC2) values at previous time step $k - 1$:

(pirin=0, STC2=0)$_{k-1}$ $\rightarrow$ STC2$_k$=1
(pirin=0, STC2=1)$_{k-1}$ $\rightarrow$ STC2$_k$=1
(pirin=1, STC2=0)$_{k-1}$ $\rightarrow$ STC2$_k$=0
(pirin=1, STC2=1)$_{k-1}$ $\rightarrow$ STC2$_k$=1

In the study conducted in [53], the expression of WNT5A was found to be highly discriminatory between cells with properties typically associated with high metastatic competence

TABLE I: Boolean functions for the melanoma WNT5A gene regulatory network.

Genes	Input Gene(s)	Output
WNT5A	HADHB	10
pirin	pirin, RET1, HADHB	00010111
S100P	S100P, RET1, STC2	10101010
RET1	RET1, HADHB, STC2	00001111
MART1	pirin, MART1, STC2	10101111
HADHB	pirin, S100P, RET1	01110111
STC2	pirin, STC2	1101

versus those with low metastatic competence. Furthermore, [54] suggests that WNT5A activation be reduced indirectly through control of other genes' activities. The reason is that an intervention that blocked the WNT5A protein from activating its receptor, could substantially reduce WNT5A's ability to induce a metastatic phenotype. For more information about the biological rationale for this, the reader is referred to [37].

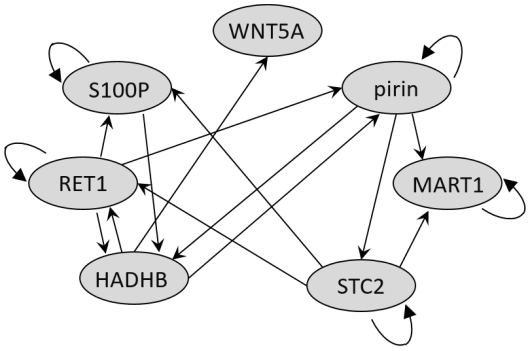


Fig. 3: Melanoma WNT5A gene regulatory network

In our numerical experiments, the intervention is applied to either RET1 or HADHB. As the goal of control is to prevent WNT5A to be upregulated, we assume the following reference immediate cost function:

$$c(\mathbf{x},\mathbf{u}) = \begin{cases} 5+||\mathbf{u}||_1 & \text{if WNT5A is 1 in state } \mathbf{x}, \\ ||\mathbf{u}||_1 & \text{if WNT5A is 0 in state } \mathbf{x}. \end{cases} \quad (27)$$

The observed data $D = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{Y}}_{1:T}\}$ is simulated by 1) obtaining a realization $\{\mathbf{X}_{0:T}, \tilde{\mathbf{u}}_{0:T}\}$ according to the state process (7), corresponding to the melanoma gene regulatory network, with the following expert's intervention policy:

$$\tilde{\mathbf{u}}_k \;=\; \underset{\mathbf{u}\in\mathbb{U}}{\operatorname{argmin}} \; \exp\left(-\eta^* \, Q^*(\mathbf{X}_k,\mathbf{u})\right), \quad (28)$$

for $k = 0,...,T$, where $\eta^* = 15$ and Q^* is the optimal Q-function computed based on the immediate cost function in (27), and the control input space is either

$$\mathbb{U}^{\mathrm{RET1}} = \{(0,0,0,1,0,0,0),(0,0,0,0,0,0,0)\}, \quad (29)$$

or

$$\mathbb{U}^{\mathrm{HADHB}} = \{(0,0,0,0,0,1,0),(0,0,0,0,0,0,0)\}, \quad (30)$$

TABLE II: Parameter values used in all experiments with the melanoma WNT5A network.

Parameter	**Value**
Number of genes d	7
Transition noise intensity p	0.01, 0.05
Initial belief $\mathbf{\Pi}_{0\|0}(i)$, $i = 1,\ldots,128$	1/128
Expression mean $m_j^0, m_j^1, j = 1,\ldots,7$	40, 60
Expression standard deviation $\sigma_j^0 = \sigma_j^1$, $j = 1,\ldots,7$	10, 20
Discount factor γ	0.95
MCMC iteration N	100,000
MCMC burn-in sample size M	1,000
Expert confidence η	0.1, 1, 10
Q-Learning parameters	$n^{\max} = 10^6$ $C = 0.8$

depending on whether the control gene is RET1 or HADHB, respectively; and 2) generating $\tilde{\mathbf{Y}}_{1:T}$ from $\mathbf{X}_{0:T}$ using the POBDS observation model in (9)–(10). It should be emphasized that (28) is only a simulation of the expert's intervention, since in practice the expert would not have access to the true state process. In this simulation, the accuracy of the expert's policy can be controlled by varying the parameter η^*, with larger values rendering more accurate policies.

In all the numerical experiments, we assume the same fixed set of values for the system parameters, summarized in Table II. All average results presented in the numerical experiments are computed over 1000 independent runs.

In the sequel, we consider different scenarios for the availability of prior knowledge about the immediate cost function, and apply the proposed cost function estimation method using the expert's sequence.

1) Case 1: Known Undesirable Gene with Unknown Severity: Here, we assume that the undesirable gene WNT5A is known. However, the cost (severity) of activation of WNT5A versus the cost of intervention is not known. Therefore, the immediate cost function is modeled as:

$$c_\theta(\mathbf{x},\mathbf{u}) = \begin{cases} \theta+||\mathbf{u}||_1 & \text{if WNT5A is 1 in state } \mathbf{x}, \\ ||\mathbf{u}||_1 & \text{if WNT5A is 0 in state } \mathbf{x}, \end{cases} \quad (31)$$

where the parameter θ denotes the unknown cost of activation of WNT5A; contrast to (27).

The RET1 gene is used as the control gene and the process and measurement noises are assumed to be $p = 0.01$ and $\sigma_j^0 = \sigma_j^1 = 5$, $j = 1,...,d$, respectively. The average absolute difference between the estimated parameter $\hat{\theta}$ computed by the proposed IBRL method (Algorithm 3) and the reference parameter $\theta^* = 5$ for various choices of η and different lengths of expert's sequence T is displayed in Fig. 4. The prior probability for θ is: 1) uniformly distributed between 0 and 7 in Fig. 4(a), 2) normally distributed with mean 4 and standard deviation 1 in Fig. 4(b). One can see that the average

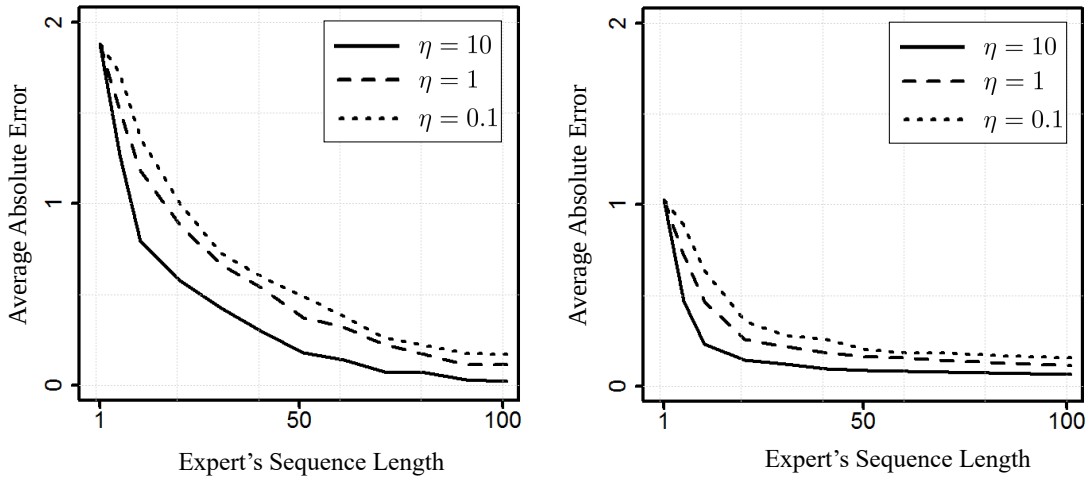


Fig. 4: Average absolute difference between the estimated parameter $\hat{\theta}$ and the reference value $\theta^* = 5$ for (a) $\theta \sim$ uniform$[0, 7]$, and (b) $\theta \sim \mathcal{N}(4, 1)$, for the melanoma WNT5A network.

difference is converging to zero as the length of the expert's sequence increases. Since the expert policy is modeled using $\eta^* = 15$ in (28), the average difference is smallest for the closest value of η to that, namely, $\eta = 10$. One can also see that performance is better with a Gaussian prior in comparison to the uniform one. The reason is that the thin Gaussian prior makes the MCMC iteration more efficient than the uniform prior.

2) Case 2: Unknown Undesirable Gene and Severity: In this case, we assume that both the undesirable gene (WNT5A) and cost of activation are unknown. The immediate cost function is thus modeled as:

$$c_\theta(\mathbf{x}^j, \mathbf{u}) = \theta_1 \mathbf{x}^j(1) + \theta_2 \mathbf{x}^j(2) + \theta_3 \mathbf{x}^j(3) + \theta_4 \mathbf{x}^j(4) + ||\mathbf{u}||_1 \,, \tag{32}$$

where $\theta = (\theta_1, \theta_2, \theta_3, \theta_4)$. From (27), the reference value is $\theta^* = (5, 0, 0, 0)$.

The noise intensities are set to $p = 0.01$ and $\sigma_j^0 = \sigma_j^1 = 5$, and the prior for all components of θ is uniform between 0 and 6, independent of each other. The average absolute difference between the estimated parameter vector and the reference vector is displayed in Fig. 5. As expected, the performance of the proposed BIRL method improves steadily as the length of expert's sequence and the parameter η increase.

3) State-Feedback Controller with Unknown Immediate Cost: In this section, we demonstrate the application of the proposed method in the design of a simple state-feedback controller, namely the V_BKF controller [39], [42], to shift the dynamics of the melanoma network away from states associated with metastasis.

In fact, after quantification of the parameters of the immediate cost function using the proposed method, any controller can be designed for either finite or infinite horizon control of the partially-observed GRN. We select the V_BKF controller here for its simplicity.

The V_BKF is based on the Boolean Kalman filter (BKF) and the stationary policy of the underlying Boolean dynamical system. During the execution process, the BKF computes the optimal MMSE estimation of the Boolean state using all available control inputs and measurements. The estimated state is then treated as the true Boolean state for choosing the control input based on the computed control policy for the underlying Boolean dynamical system. For more information about this method, the reader is referred to [39], [42].

The cost parametrization in (31) is used in this experiment. The performance of the state feedback controller (V_BKF) with unknown and known immediate cost function as well as the system without control are compared. Table III shows the average cost per step over 10,000 time steps achieved by the various methods. Three different expert's sequence lengths are considered here: $T = 10, T = 50$, and $T = 100$, and RET1 and HADHB are both considered as the control genes.

We can observe that the performance of the V_BKF with unknown immediate cost function is better for larger expert's sequence lengths, since the parameters of the immediate cost function can be estimated more accurately, leading to better controller performance. For large process and measurement noise intensities, the performance decreases in all cases. This reduction is more obvious for a system with unknown cost function and particularly small expert's sequence length. This can be justified based on two main facts: 1) the error of the Boolean Kalman smoother (BKS) for mapping the noisy gene-expression data to Boolean state trajectory for estimating the immediate cost function decreases under large process or measurement noise; 2) the V_BKF is a state-feedback controller, the performance of which strongly depends on the accuracy of estimation by the Boolean Kalman filter. The underlying Boolean dynamical system is less identifiable in the presence of large measurement noise, which makes the estimation task more challenging as well. Therefore, the

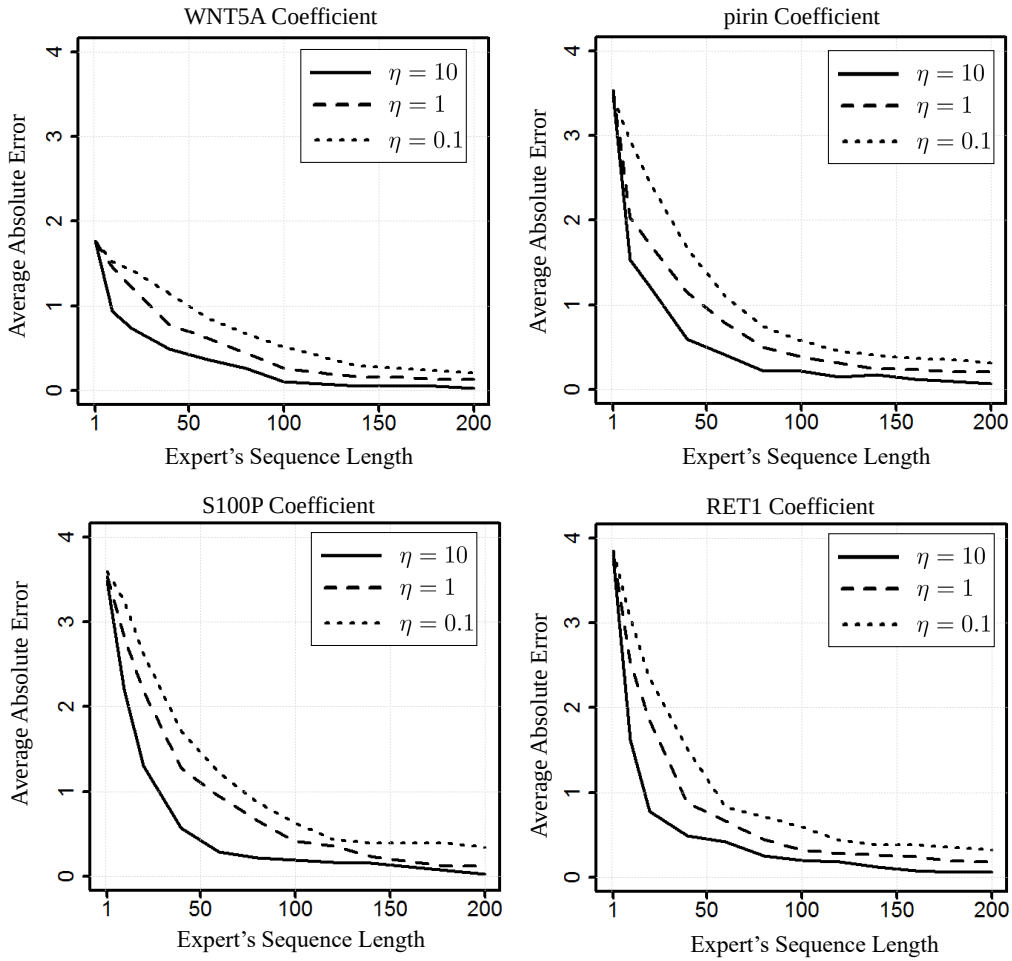


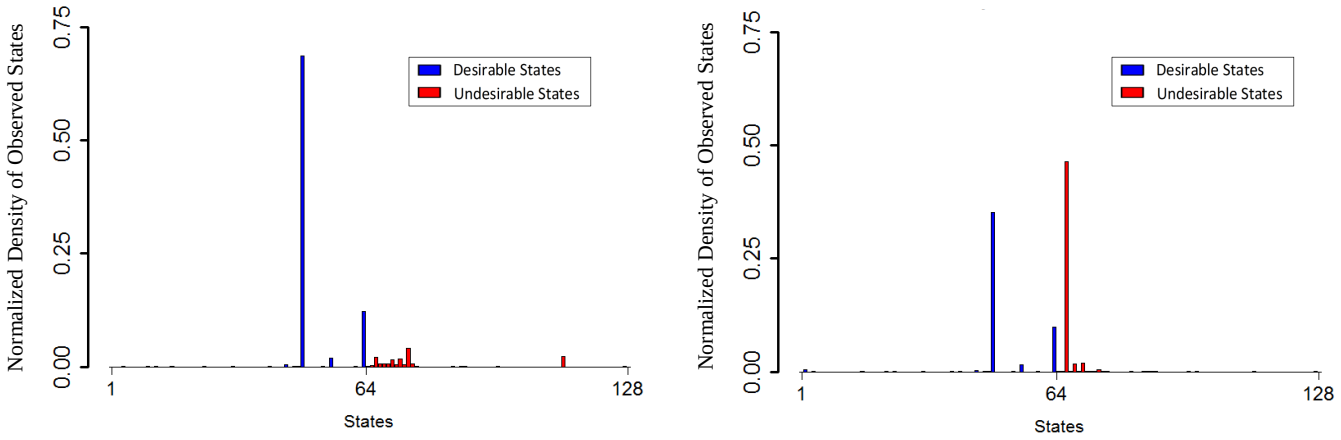
Fig. 5: Average absolute difference between the estimated parameter vector $\hat{\theta}$ and the reference value $\theta^* = (5, 0, 0, 0)$, for the melanoma WNT5A network.

TABLE III: Average results for state-feedback controller with both known and unknown immediate cost functions, for the melanoma WNT5A network.

			V_BKF with Unknown Cost				
Control	p	$\sigma_j^0 = \sigma_j^1$	$T = 10$	$T = 50$	$T = 100$	V_BKF	No-Control
RET1	0.01	10	0.57	0.21	0.18	0.15	2.28
		20	0.98	0.32	0.29	0.25	
	0.05	10	1.12	0.83	0.79	0.73	2.33
		20	1.71	1.17	1.10	0.99	
HADHB	0.01	10	0.97	0.56	0.49	0.45	2.28
		20	1.33	0.99	0.91	0.83	
	0.05	10	1.82	1.38	1.29	1.20	2.33
		20	2.09	1.75	1.62	1.48	

policies obtained by V_BKF become less accurate under large noise conditions.

Moreover, from the results of Table III, one can see that the RET1 gene seems to be a better control input than HDHAB for reducing the activation of WNT5A, in terms of lower achieved cost.



(a) System Under Control of RET1

(b) System Without Control

Fig. 6: Normalized histogram plots of observed states for the V_BKF method with unknown cost function and for the system without control, for the melanoma WNT5A GRN.

Finally, Fig. 6 displays normalized histograms of observed states using 10,000 time series with 1000 time steps each, comparing the V_BKF and the system without control. For this comparison, RET1 is used as a control gene and the parameters are set to $p = 0.01$, $\sigma_j^0 = \sigma_j^1 = 5$, $j = 1, ..., 10$, $T = 100$. The histograms over desirable and undesirable states are shown by blue and red colors respectively (recalling that the undesirable states are those where WNT5A is activated). It is clear that the system under control of the proposed method visited undesirable states less often than the system without control.

B. Performance Evaluation Based on a p53-MDM2 Network

In this section, we investigate the performance of the proposed method on a Boolean model of the p53-MDM2 negative feedback loop gene regulatory network [51]. The p53 gene codes for the tumor suppressor protein p53 in humans, and its activation plays a critical role in cellular responses to various stress signals that might cause genome instability [55]. The gene regulatory network consists of four genes: ATM, p53, Wip1, and MDM2, and the input "dna_dsb", which indicates the presence of DNA double strand breaks.

The pathway diagram for this network is presented in the left plot of Fig. 7. Normal arrows represent activating regulations and blunt arrows represent suppressive regulations. The Boolean function is represented by the following logic functions:

$$\begin{aligned}
\text{ATM}_k &= \overline{\text{WIP1}_{k-1}} \ \textit{AND} \ \text{dna_dsb} \\
\text{p53}_k &= \text{ATM}_{k-1} \ \textit{AND} \ \overline{\text{WIP1}_{k-1}} \ \textit{AND} \ \overline{\text{MDM2}_{k-1}} \\
\text{WIP1}_k &= \text{p53}_{k-1} \\
\text{MDM2}_k &= \left(\overline{\text{ATM}_{k-1}} \ \textit{AND} \ (\text{p53}_{k-1} \ \textit{OR} \ \text{WIP1}_{k-1})\right) \\
&\qquad \textit{OR} \ \left(\text{p53}_{k-1} \ \textit{AND} \ \text{WIP1}_{k-1}\right).
\end{aligned}$$

We can see that ATM is the transductor gene for the DNA damage signal, which eventually activates p53, through inactivation of MDM2. However, there is also a negative feedback loop between p53 and ATM, through Wip1, so that p53 is expected to display an oscillatory behavior under DNA damage; on the other hand, under no stress, it is known that all four proteins are inactivated in the steady state [55].

The state transition diagram of the system under DNA damage is shown in the right plot of Fig. 7. Five states are part of the cyclic attractor of the system. When the system is in the attractor, the process of DNA repair is in effect. However, when the system gets out of attractor, the repairing process is disturbed. Thus, in the DNA damage condition, the *transient states*, which are the states not in the attractor, are all undesirable. Since the desirable states (the attractor states) are $\mathbb{U}^{\text{des}} = \{(1,0,0,0),(1,1,0,0),(1,1,1,0),(0,0,1,1),(0,0,0,1)\}$, we assume the following reference immediate cost function:

$$c(\mathbf{x},\mathbf{u}) = \begin{cases} 5 + ||\mathbf{u}||_1 & \text{if } \mathbf{x} \notin \mathbb{U}^{\text{des}} \\ ||\mathbf{u}||_1 & \text{o.w.} \end{cases} \tag{33}$$

for $\mathbf{u} \in \mathbb{U}$ and $\mathbf{x} \in \{0,1\}^d$; where the control space $\mathbb{U}$ is set to be $\mathbb{U} = \{(0,0,0,0),(1,0,0,0),(0,1,0,0)\}$. The expert sequence is created using equation (28) with $\gamma^* = 15$.

In our experiment, we assume that the undesirable condition (i.e. the identity of transient states) are known, but the intensity of their undesirable outcomes is unknown. The immediate cost function is thus modeled as:

$$c_\theta(\mathbf{x}^j,\mathbf{u}) = \begin{cases} \theta + ||\mathbf{u}||_1 \text{ if } \mathbf{x} \notin \mathbb{U}^{\text{des}}, \\ ||\mathbf{u}||_1 \text{ if } \mathbf{x} \in \mathbb{U}^{\text{des}}, \end{cases} \tag{34}$$

where the reference value of θ is $\theta^* = 5$. The parameters are set to $\eta = 10$, $\gamma = 0.95$, $M = 1,000$, $N = 100,000$, $\sigma_j^0 = \sigma_j^1 = 10$, and the prior of θ is assumed to be uniform between 1 and 8.

The average percentage of time spent in desirable states for the system under control of a state-feedback controller with known and unknown immediate cost function, as well as for the system without control, is presented in Table IV. For the system without control, the shortest average time spent in desirable states is obtained in the presence of large process noise, since in that case the system will transition out of its

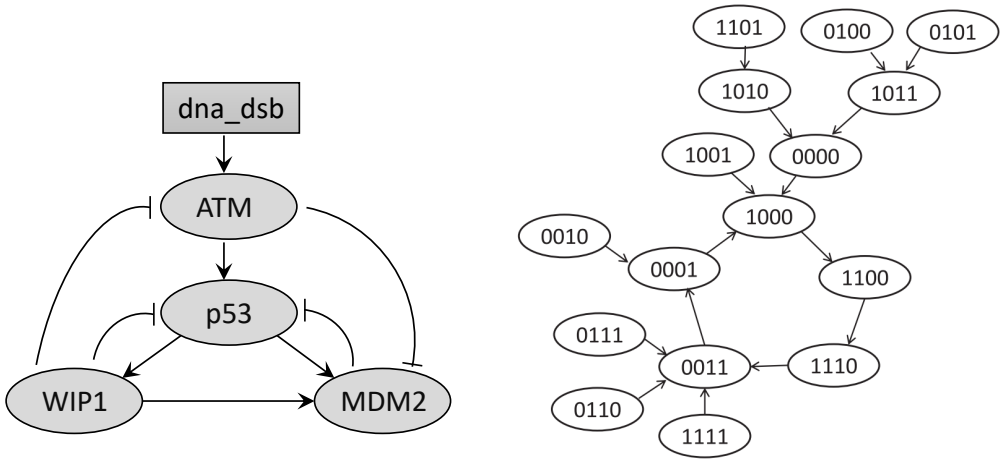


Fig. 7: Activation/repression pathway diagram and state transition diagram corresponding to a constant input dna_dsb $= 1$ (DNA-damage) for the p53-MDM2 negative feedback loop Boolean network model.

attractor more often, leading to more undesirable transient states being visited. As expected, the system spends more time on average in desirable states under control with known or partially-known immediate cost function, in comparison to the system without control. In addition, as the length of the expert's sequence increases, the performance of the system with unknown immediate cost function becomes closer to that of the system with known immediate cost function.

TABLE IV: Average percentage of time spent in desirable states (i.e. attractor states) for the p53-MDM2 GRN under a state-feedback controller with both known and unknown immediate cost functions.

	V_BKF with Unknown Cost				
p	$T = 20$	$T = 60$	$T = 100$	V_BKF	No-Control
0.10	68	72	74	74	67
0.15	55	62	64	65	53
0.20	48	51	53	53	46

V. Conclusion

In this paper, a method for quantification of the expert behavior required for control of partially-observed gene regulatory networks (GRNs) was proposed. The partially-observed Boolean dynamical system (POBDS) signal model was used for modeling the transcriptional data derived from the GRN. Given a single timeline of interventions made by an expert and the accompanying transcriptional data, the Boolean Kalman smoother (BKS) was first employed to optimally map the expert sequence to the smoothed Boolean trajectory. Then, the Bayesian inverse reinforcement learning technique was proposed in combination with the Q-learning method to efficiently estimate parameters of immediate cost function. The ability of the proposed methodology to obtain a good control policy was demonstrated by numerical experiments involving models of important GRNs: a Melanoma WNT5A Boolean network and a p53-MDM2 negative feedback loop Boolean network, observed through noisy gene expression data.

Acknowledgment

The authors acknowledge the support of the National Science Foundation, through NSF award CCF-1718924.

References

[1] Stuart A Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.

[2] Ilya Shmulevich, Edward R Dougherty, and Wei Zhang. From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proceedings of the IEEE*, 90(11):1778–1792, 2002.

[3] Mahdi Imani and Ulisses Braga-Neto. Maximum-likelihood adaptive filter for partially-observed Boolean dynamical systems. *IEEE transaction on Signal Processing*, 65:359–371, 2017.

[4] Ting Chen, Hongyu L He, George M Church, et al. Modeling gene expression with differential equations. *Pacific symposium on biocomputing*, 4(29):40, 1999.

[5] MK Stephen Yeung, Jesper Tegnér, and James J Collins. Reverse engineering gene networks using singular value decomposition and robust regression. *Proceedings of the National Academy of Sciences*, 99(9):6163–6168, 2002.

[6] Shinichi Kikuchi, Daisuke Tominaga, Masanori Arita, Katsutoshi Takahashi, and Masaru Tomita. Dynamic modeling of genetic networks using genetic algorithm and S-system. *Bioinformatics*, 19(5):643–650, 2003.

[7] Shuhei Kimura, Kaori Ide, Aiko Kashihara, Makoto Kano, Mariko Hatakeyama, Ryoji Masui, Noriko Nakagawa, Shigeyuki Yokoyama, Seiki Kuramitsu, and Akihiko Konagaya. Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm. *Bioinformatics*, 21(7):1154–1163, 2004.

[8] Nir Friedman, Michal Linial, Iftach Nachman, and Dana Pe'er. Using Bayesian networks to analyze expression data. *Journal of computational biology*, 7(3-4):601–620, 2000.

[9] Kevin Murphy, Saira Mian, et al. Modelling gene expression data using dynamic Bayesian networks. Technical report, Computer Science Division, University of California, Berkeley, CA, 1999.

[10] Annelien Verfaillie, Hana Imrichová, Bram Van de Sande, Laura Standaert, Valerie Christiaens, Gert Hulselmans, Koen Herten, Marina Naval Sanchez, Delphine Potier, Dmitry Svetlichnyy, et al. iRegulon: from a gene list to a gene regulatory network using large motif and track collections. *PLoS computational biology*, 10(7):e1003731, 2014.

[11] Shane Neph, Andrew B Stergachis, Alex Reynolds, Richard Sandstrom, Elhanan Borenstein, and John A Stamatoyannopoulos. Circuitry and dynamics of human transcription factor regulatory networks. *Cell*, 150(6):1274–1286, 2012.

[12] Terri A Long, Siobhan M Brady, and Philip N Benfey. Systems approaches to identifying gene regulatory networks in plants. *Annual review of cell and developmental biology*, 24:81–103, 2008.

[13] Shahin Boluki, Mohammad Shahrokh Esfahani, Xiaoning Qian, and Edward Russell Dougherty. Constructing pathway-based priors within a Gaussian mixture model for Bayesian regression and classification. *IEEE/ACM transactions on computational biology and bioinformatics*, 2017.

[14] Lesley T MacNeil and Albertha JM Walhout. Gene regulatory networks and the role of robustness and stochasticity in the control of gene expression. *Genome research*, 21(5):645–657, 2011.

[15] Alireza Karbalayghareh, Ulisses Braga-Neto, and Edward Russell Dougherty. Classification of single-cell gene expression trajectories from incomplete and noisy data. *IEEE/ACM transactions on computational biology and bioinformatics*, 2017.

[16] Zhi-Ping Liu, Canglin Wu, Hongyu Miao, and Hulin Wu. RegNetwork: an integrated database of transcriptional and post-transcriptional regulatory networks in human and mouse. *Database*, 2015:bav095, 2015.

[17] Zhi-Ping Liu, Hulin Wu, Jian Zhu, and Hongyu Miao. Systematic identification of transcriptional and post-transcriptional regulations in human respiratory epithelial cells during influenza a virus infection. *BMC bioinformatics*, 15(1):336, 2014.

[18] Réka Albert and Hans G Othmer. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster. *Journal of theoretical biology*, 223(1):1–18, 2003.

[19] Stuart Kauffman, Carsten Peterson, Björn Samuelsson, and Carl Troein. Random Boolean network models and the yeast transcriptional network. *Proceedings of the National Academy of Sciences*, 100(25):14796–14799, 2003.

[20] Maria I Davidich and Stefan Bornholdt. Boolean network model predicts cell cycle sequence of fission yeast. *PloS one*, 3(2):e1672, 2008.

[21] Adrien Fauré, Aurélien Naldi, Claudine Chaouiya, and Denis Thieffry. Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle. *Bioinformatics*, 22(14):e124–e131, 2006.

[22] Ilya Shmulevich and Edward R Dougherty. *Genomic signal processing*. Princeton University Press, 2014.

[23] I Shmulevich and ER Dougherty. *Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2009.

[24] Daizhan Cheng, Hongsheng Qi, and Zhiqiang Li. *Analysis and control of Boolean networks: a semi-tensor product approach*. Springer Science & Business Media, 2010.

[25] Yidong Chen, Edward R Dougherty, and Michael L Bittner. Ratio-based decisions and the quantitative analysis of cDNA microarray images. *Journal of Biomedical optics*, 2(4):364–374, 1997.

[26] Ali Mortazavi, Brian A Williams, Kenneth McCue, Lorian Schaeffer, and Barbara Wold. Mapping and quantifying mammalian transcriptomes by RNA-seq. *Nature methods*, 5(7):621–628, 2008.

[27] Jianping Hua, Chao Sima, Milana Cypert, Gerald C Gooden, Sonsoles Shack, Lalitamba Alla, Edward A Smith, Jeffrey M Trent, Edward R Dougherty, and Michael L Bittner. Dynamical analysis of drug efficacy and mechanism of action using GFP reporters. *Journal of Biological Systems*, 20(04):403–422, 2012.

[28] Ulisses Braga-Neto. Optimal state estimation for Boolean dynamical systems. In *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, pages 1050–1054. IEEE, 2011.

[29] Mahdi Imani and Ulisses Braga-Neto. Optimal state estimation for Boolean dynamical systems using a Boolean Kalman smoother. In *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 972–976. IEEE, 2015.

[30] Mahdi Imani and Ulisses Braga-Neto. Particle filters for partially-observed Boolean dynamical systems. *Automatica*, 87:238–250, 2018.

[31] Mahdi Imani and Ulisses Braga-Neto. Optimal gene regulatory network inference using the Boolean Kalman filter and multiple model adaptive estimation. In *2015 49th Asilomar Conference on Signals, Systems and Computers*, pages 423–427. IEEE, 2015.

[32] Mahdi Imani and Ulisses Braga-Neto. Optimal finite-horizon sensor selection for Boolean Kalman filter. In *2017 51th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2017.

[33] Levi Daniel McClenny, Mahdi Imani, and Ulisses Braga-Neto. Boolean Kalman filter with correlated observation noise. In *the 42nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*. IEEE, 2017.

[34] Levi D McClenny, Mahdi Imani, and Ulisses M Braga-Neto. BoolFilter: an R package for estimation and identification of partially-observed Boolean dynamical systems. *BMC bioinformatics*, 2017.

[35] Levi D McClenny, Mahdi Imani, and Ulisses Braga-Neto. BoolFilter package vignette. *The Comprehensive R Archive Network (CRAN)*, 2017.

[36] Ranadip Pal, Aniruddha Datta, and Edward R Dougherty. Optimal infinite-horizon control for probabilistic Boolean networks. *Signal Processing, IEEE Transactions on*, 54(6):2375–2387, 2006.

[37] Edward R Dougherty, Ranadip Pal, Xiaoning Qian, Michael L Bittner, and Aniruddha Datta. Stationary and structural control in gene regulatory networks: basic concepts. *International Journal of Systems Science*, 41(1):5–16, 2010.

[38] B Faryabi, A Datta, and ER Dougherty. On approximate stochastic control in genetic regulatory networks. *IET Systems Biology*, 1(6):361–368, 2007.

[39] Mahdi Imani and Ulisses Braga-Neto. State-feedback control of partially-observed Boolean dynamical systems using RNA-seq time series data. In *American Control Conference (ACC), 2016*, pages 227–232. IEEE, 2016.

[40] Mahdi Imani and Ulisses Braga-Neto. Point-based value iteration for partially-observed Boolean dynamical systems with finite observation space. In *Decision and Control (CDC), 2016 IEEE 55th Conference on*, pages 4208–4213. IEEE, 2016.

[41] Mahdi Imani and Ulisses Braga-Neto. Control of gene regulatory networks with noisy measurements and uncertain inputs. *IEEE Transactions on Control of Network Systems*, 2018. doi: 10.1109/TCNS.2017.2746341.

[42] Mahdi Imani and Ulisses Braga-Neto. Multiple model adaptive controller for partially-observed Boolean dynamical systems. In *Proceedings of the 2017 American Control Conference (ACC 2017), Seattle, WA*, pages 1103–1108. IEEE, 2017.

[43] Shuilian Xie, Mahdi Imani, Ulisses Braga-Neto, and Edward Russell Dougherty. Nonstationary Linear Discriminant Analysis. In *2017 51th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2017.

[44] Andrew Y Ng, Stuart J Russell, et al. Algorithms for inverse reinforcement learning. In *Icml*, pages 663–670, 2000.

[45] Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. *Urbana*, 51(61801):1–4, 2007.

[46] Brian D Ziebart, Andrew L Maas, J Andrew Bagnell, and Anind K Dey. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438. Chicago, IL, USA, 2008.

[47] Manuel Lopes, Francisco Melo, and Luis Montesano. Active learning for reward estimation in inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 31–46. Springer, 2009.

[48] Gergely Neu and Csaba Szepesvári. Apprenticeship learning using inverse reinforcement learning and gradient methods. *arXiv preprint arXiv:1206.5264*, 2012.

[49] Dimitri P Bertsekas. *Dynamic programming and optimal control*. Athena Scientific Belmont, MA, 1995.

[50] Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

[51] R. Layek and A. Datta. From biological pathways to regulatory networks. *Molecular Biosystems*, 7:843–851, 2011.

[52] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[53] Meltzer Bittner, P Meltzer, Y Chen, Y Jiang, E Seftor, M Hendrix, M Radmacher, Rm Simon, Z Yakhini, A Ben-Dor, et al. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, 406(6795):536–540, 2000.

[54] Ashani T Weeraratna, Yuan Jiang, Galen Hostetter, Kevin Rosenblatt, Paul Duray, Michael Bittner, and Jeffrey M Trent. Wnt5a signaling directly affects cell motility and invasion of metastatic melanoma. *Cancer cell*, 1(3):279–288, 2002.

[55] Eric Batchelor, Alexander Loewer, and Galit Lahav. The ups and downs of p53: understanding protein dynamics in single cells. *Nature Reviews Cancer*, 9(5):371–377, 2009.

Mahdi Imani received his B.Sc. degree in Mechanical Engineering and his M.Sc. degree in Electrical Engineering, both from University of Tehran in 2012 and 2014. He is currently a Ph.D. student at the Department of Electrical and Computer Engineering of Texas A&M University, College Station, TX. His research interests include machine learning, control theory and signal processing.

Ulisses M. Braga-Neto is an Associate Professor at the Department of Electrical and Computer Engineering and a member of the Center for Bioinformatics and Genomic Systems Engineering at Texas A&M University, College Station, TX. He holds a Ph.D. degree in Electrical and Computer Engineering from The Johns Hopkins University, Baltimore, MD and has held post-doctoral positions at the University of Texas M.D. Anderson Cancer Center, Houston, TX and at the Oswaldo Cruz Foundation, Recife, Brazil. His research interests include Pattern Recognition and Statistical Signal Processing. Dr. Braga-Neto is a Senior Member of the IEEE. He is author of the textbook Error Estimation for Pattern Recognition (IEEE-Wiley, 2015) and has received the NSF CAREER Award for his work in this area.