Optimal Control of Gene Regulatory Networks with Unknown Cost Function

Mahdi Imani and Ulisses Braga-Neto

***Abstract*— Most of the existing methodologies for control of Gene Regulatory Networks (GRNs) assume that the immediate cost function at each state and time point is fully known. In this paper, we introduce an optimal control strategy for control of GRNs with unknown or partially-known immediate cost function. Toward this, we adopt a partially-observed Boolean dynamical system (POBDS) model for the GRN and propose a *Inverse Reinforcement Learning* (IRL) methodology for quantifying the imperfect behavior of experts, obtained via prior biological knowledge or experimental data. The constructed cost function then is used in finding the optimal infinite-horizon control strategy for the POBDS. The application of the proposed method using a single sequence of experimental data is investigated through numerical experiments using a melanoma gene regulatory network.**

I. Introduction

A fundamental problem in genomic signal processing is the design of intervention strategies for gene regulatory networks (GRNs), in order to beneficially alter network dynamics. Several mathematical models have been developed for modeling GRNs [1]–[4]. Several control strategies have also been developed for various GRN models to reduce the steady-state probability mass over undesirable states, such as cell proliferation states, which may be associated with cancer [5]–[8]. Boolean networks have been shown to be effective in capturing much of the complex dynamics of gene regulatory networks [9]–[12]. In Boolean networks, 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 [13], i.e., through a Boolean dynamical system. These models were first introduced as a completely-observable, deterministic model by Kauffman and collaborators [1]. Several variations of original Boolean network models 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) [14], Probabilistic Boolean Networks (PBN) [15], and Boolean Control Networks (BCN) [16], [17]. The Partially-Observed Boolean dynamical system (POBDS) model unifies and generalizes all of the aforementioned Boolean network models. It is also more realistic, in that it allows the GRN states to be hidden and only partially observable through noisy measurements from gene-expression technologies such as cDNA microarrays, live cell imaging-based assays, and RNA-Seq data [7], [18]–[21].

Most of the existing intervention techniques developed for GRNs (e.g. [5]–[7]) are based on the restrictive assumption that the immediate effects of taking interventions on the gene transcriptional states are fully known. However, this immediate cost function might not be available in practice. This paper provides a methodology that uses a sequence of intervention actions performed by an expert in an experimental setting to construct the immediate cost function. More specifically, given a pair of sequences of observations and interventions, we develop a methodology based on *Inverse Reinforcement Learning* [22] for quantifying the expert sequence to obtain the immediate cost function. We demonstrate the application of the proposed method with experiments based on a Boolean model of a melanoma gene regulatory network, where the control objective is to drive the system evolution away from the states associated with metastasis.

II. POBDS Model

The system is described by a *state process* $\{\mathbf{X}_k; k = 0, 1, \ldots\}$, where $\mathbf{X}_k \in \{0,1\}^d$ represents the gene activation/inactivation state at time k. The system state is affected by a sequence of *control inputs* $\{\mathbf{u}_k; k = 0, 1, \ldots\}$, where $\mathbf{u}_k$ is a vector of size d, which represents a purposeful control input and takes its value from a finite set $\mathbb{U}$. The states are assumed to be updated at each discrete time through the following nonlinear signal model:

$$\mathbf{X}_k = \mathbf{f}(\mathbf{X}_{k-1}) \oplus \mathbf{u}_{k-1} \oplus \mathbf{n}_k, \tag{1}$$

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. The way that the control input influences state evolution is that if $\mathbf{u}_{k-1}(i)$ is one, it flips the value of ith bit of the Boolean state $\mathbf{X}_k$. In practice, control would be accomplished by means of drug interventions targeted at those genes. We assume that the bits in $\mathbf{n}_k$ 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 $i = 1, \ldots, d$. 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 the maximum uncertainty.

The states of the system are observed indirectly through noisy gene-expression data sequence $\mathbf{Y}_{1:T} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_T)$. The relationship between the observation data and the state process is specified by a conditional distribution

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

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

$p\big(\mathbf{Y}_k \mid \mathbf{X}_k\big)$, for $k = 1, \ldots, T$. For example, this conditional distribution may be Gaussian in the case of cDNA microarray data [23] or a Poisson distribution in the case of RNA-seq count data [24].

III. Problem Formulation

In this section, the infinite-horizon control policy for GRNs with directly observed states is outlined. In the infinite-horizon control problem, the goal is to select the appropriate control inputs $\mathbf{u}_k \in \mathbb{U}$ at time point k, in such a way that the system spends the least amount of time, on average, in undesirable states at a small cost of control. In formal terms, let $c(\mathbf{X}_k, \mathbf{u}_k)$ be the bounded immediate cost of applying control input $\mathbf{u}_k$ when the state of the system is $\mathbf{X}_k$. Let also *control policy* $\pi : \{0,1\}^d \to \mathbb{U}$ be a function which associates a control input to each Boolean state, and Π be the space of all possible policies. The infinite-horizon control cost function for a given policy $\pi \in \Pi$ is defined as:

$$J^{\pi}(\mathbf{x}) = E\left[\sum_{k=0}^{\infty} \gamma^k \, c\big(\mathbf{X}_k, \pi(\mathbf{X}_k)\big) \,\middle|\, \mathbf{X}_0 = \mathbf{x}\right], \tag{2}$$

for $\mathbf{x} \in \{0,1\}^d$; where the discount factor γ places a premium on minimizing the costs of early interventions as opposed to later ones, which is sensible from a medical perspective [6], and the expectation is taken over the stochasticity of the Boolean state transition under policy π. Let π^* be the optimal policy. The optimal cost function is denoted by $J^*(\mathbf{x}) = J^{\pi^*}(\mathbf{x})$, for $\mathbf{x} \in \{0,1\}^d$, which satisfies $J^*(\mathbf{x}) \le J^{\pi}(\mathbf{x})$, for any $\pi \in \Pi$ and $\mathbf{x} \in \{0,1\}^d$.

According to the theory of dynamic programming [25], the optimal value function satisfies the following Bellman equation:

$$J^*(\mathbf{x}) = \min_{\mathbf{u} \in \mathbb{U}} \Big[c(\mathbf{x}, \mathbf{u}) + \gamma \, E_{\mathbf{x}'|\mathbf{x},\mathbf{u}}\big[J^*(\mathbf{x}')\big] \Big], \tag{3}$$

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

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

$$Q^{\pi}(\mathbf{x}, \mathbf{u}) = E\left[c(\mathbf{X}_0, \mathbf{u}_0) + \sum_{r=1}^{\infty} \gamma^r \, c\big(\mathbf{X}_r, \pi(\mathbf{X}_r)\big) \,\middle|\, \mathbf{X}_0 = \mathbf{x}, \mathbf{u}_0 = \mathbf{u} \right], \tag{4}$$

for $\mathbf{x} \in \{0,1\}^d$ and $\mathbf{u} \in \mathbb{U}$; where the Q-function $Q^{\pi} : \{0,1\}^d \times \mathbb{U} \to \mathbb{R}$ of the policy π for every pair $(\mathbf{x}, \mathbf{u})$, gives the expected return when starting from state $\mathbf{x}$, applying $\mathbf{u}$, and following π thereafter. This cost function satisfies the following Bellman equation:

$$Q^{\pi}(\mathbf{x}, \mathbf{u}) = c(\mathbf{x}, \mathbf{u}) + \gamma \, E_{\mathbf{x}'|\mathbf{x},\mathbf{u}}\big[Q^{\pi}(\mathbf{x}', \pi(\mathbf{x}'))\big], \tag{5}$$

for $\mathbf{x} \in \{0,1\}^d$ and $\mathbf{u} \in \mathbb{U}$.

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

$$Q^*(\mathbf{x}, \mathbf{u}) = \min_{\pi \in \Pi} Q^{\pi}(\mathbf{x}, \mathbf{u}). \tag{6}$$

which leads to the following optimal policy:

$$\pi^*(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{u} \in \mathbb{U}} Q^*(\mathbf{x}, \mathbf{u}), \tag{7}$$

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

The immediate cost function, $c(\mathbf{x}, \mathbf{u})$, is not usually known and one needs to approximate it using the available experimental data recorded by an expert. In the next section, an efficient methodology for quantification of the immediate cost function will be discussed.

IV. Quantification of the Immediate Cost Function using Inverse Reinforcement Learning

Let us assume the realistic case where the immediate cost function $c(\mathbf{x}, \mathbf{u})$ is unknown or partially-known. 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 of the immediate cost function does not impose any limitation on the form of the immediate cost function. For instance, if no information regarding the immediate cost function exists, θ will be a vector of size $2^d \times |\mathbb{U}|$, the element of which are the costs for all possible states and control inputs.

Inverse reinforcement learning (IRL) is a technique for recovering the unobserved underlying immediate cost (or reward) function from the behavior of an expert [22]. The original idea of IRL method is introduced in [22] followed by several variations of it [27]–[30]. In this paper, we assume no prior information regarding the unknown parameters of the immediate cost function exists.

Let θ be a realization of the parameter vector. The Q-function corresponding to the optimal immediate cost function associated to θ is denoted by Q^*_θ. This Q-function can be computed by performing a dynamic programming technique, such as Value Iteration or Policy Iteration method [25], using the immediate cost function corresponding to the parameter vector θ. Assuming the Boltzmann softmax policy [31], we have:

$$P(\mathbf{u} \mid \mathbf{x}, \theta) \propto \exp\big(\eta \, Q^*_\theta(\mathbf{x}, \mathbf{u})\big), \tag{8}$$

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

All that is available for quantification of the immediate cost function is a sequence of observed states and control inputs taken by the expert:

$$D = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{Y}}_{1:T}\}. \tag{9}$$

One has only the expert sequence in (9) to quantify the unobserved immediate cost function required for deriving a proper intervention process.

The Boolean Kalman Smoother (BKS) provides the optimal minimum-square state estimator given the entire data sequence [18], [26]. Given D, we apply the BKS as state observer:

$$D = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{Y}}_{1:T}\} \xrightarrow{\text{BKS}} \tilde{D} = \{\tilde{\mathbf{u}}_{0:T}, \tilde{\mathbf{X}}_{0:T}\}. \tag{10}$$

Now, assuming the input control $\tilde{\mathbf{u}}_k$ depends only on $\tilde{\mathbf{X}}_k$ and not on previous states, for $r = 0, \ldots, T$; then it is easy to verify that the data joint likelihood is given by:

$$\begin{aligned} L(\theta) &= P(\tilde{D} \mid \theta) \\ &= P(\tilde{\mathbf{X}}_{0:T}) \prod_{k=0}^{T} P(\tilde{\mathbf{u}}_k \mid \tilde{\mathbf{X}}_k, \theta), \end{aligned} \tag{11}$$

with $P(\tilde{\mathbf{u}}_k \mid \tilde{\mathbf{X}}_k, \theta)$ given by (8). The log of the likelihood in (11) after removing terms independent of θ can be expressed as:

$$\log L(\theta) \propto \Lambda(\theta) = \sum_{k=0}^{T} \log l_k(\theta), \tag{12}$$

where

$$l_k(\theta) = \frac{\exp\left(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_k, \tilde{\mathbf{u}}_k)\right)}{\sum_{\mathbf{u}' \in \mathbb{U}} \exp\left(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_k, \mathbf{u}')\right)}. \tag{13}$$

The maximum likelihood estimate of the parameter is

$$\hat{\theta}^* = \underset{\theta \in \Theta}{\operatorname{argmax}}\, \Lambda(\theta). \tag{14}$$

Unfortunately, the maximization in (14) cannot be solved analytically. Here, we propose a gradient-based optimization procedure for finding a local maximum of the log-likelihood. The gradient of the log-likelihood in (12) is given by

$$(\nabla_\theta \Lambda(\theta))_t = \sum_{k=0}^{T} \frac{1}{l_k(\theta)} \frac{\partial l_k(\theta)}{\partial \theta_t}, \tag{15}$$

for $t = 1, \ldots, |\theta|$, where

$$\begin{aligned} \frac{\partial l_k(\theta)}{\partial \theta_t} &= \frac{\partial \frac{\exp\left(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_k, \tilde{\mathbf{u}}_k)\right)}{\sum_{\mathbf{u}' \in \mathbb{U}} \exp\left(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_k, \mathbf{u}')\right)}}{\partial \theta_t} \\ &= -\eta \frac{\exp\left(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_k, \tilde{\mathbf{u}}_k)\right)}{[\sum_{\mathbf{u}' \in \mathbb{U}} \exp(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_k, \mathbf{u}'))]^2} \\ &\quad \left[\sum_{\mathbf{u}' \in \mathbb{U}} \exp\left(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_k, \mathbf{u}')\right) \right. \\ &\quad \left. \times \left(\frac{\partial Q^*_\theta(\tilde{\mathbf{X}}_k, \tilde{\mathbf{u}}_k)}{\partial \theta_t} - \frac{\partial Q^*_\theta(\tilde{\mathbf{X}}_k, \mathbf{u}')}{\partial \theta_t} \right) \right]. \end{aligned} \tag{16}$$

Thus, one needs to compute the gradient of $Q^*_\theta(\tilde{\mathbf{X}}_k, \mathbf{u})$, for $k = 0, \ldots, T$ and $\mathbf{u} \in \mathbb{U}$, with respect to parameter θ.

Let the optimal cost function corresponding to θ be represented by J^*_θ, and let $(\mathbf{x}^1, \ldots, \mathbf{x}^{2^d})$ be an arbitrary enumeration of the possible state vectors. It is easy to show that:

$$J^*_\theta = \mathbf{c}_{\pi^*_\theta} + \gamma\, P_{\pi^*_\theta}\, J^*_\theta, \tag{17}$$

where $\mathbf{c}_{\pi^*_\theta}$ is a vector of size 2^d with ith element given by

$$\mathbf{c}_{\pi^*_\theta}(i) = c_\theta\left(\mathbf{x}^i, \pi^*_\theta(\mathbf{x}^i)\right), \tag{18}$$

and P_{π^*} is a matrix of size $2^d \times 2^d$ given by:

$$P_{\pi^*_\theta} = M_k\left(\pi^*_\theta(\mathbf{x}^j)\right), \tag{19}$$

where M_k is the *controlled transition matrix* of the underlying Markov state process, given by

$$\begin{aligned} (M_k(\mathbf{u}))_{ij} &= P(\mathbf{X}_k = \mathbf{x}^i \mid \mathbf{X}_{k-1} = \mathbf{x}^j, \mathbf{u}_{k-1} = \mathbf{u}) \\ &= P\left(\mathbf{n}_k = \mathbf{f}(\mathbf{x}^j) \oplus \mathbf{u} \oplus \mathbf{x}^i\right) \\ &= p^{\|\mathbf{f}(\mathbf{x}^j) \oplus \mathbf{u} \oplus \mathbf{x}^i\|_1} (1-p)^{d - \|\mathbf{f}(\mathbf{x}^j) \oplus \mathbf{u} \oplus \mathbf{x}^i\|_1}, \end{aligned} \tag{20}$$

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

Solving equation (17) for J^*_θ leads to

$$J^*_\theta = \mathbf{T}\, \mathbf{c}_{\pi^*_\theta}, \tag{21}$$

where

$$\mathbf{T} = (\mathbf{I}_{2^d} - \gamma\, P_{\pi^*_\theta})^{-1}, \tag{22}$$

with $\mathbf{I}_{2^d}$ denoting the identity matrix of size 2^d. On the other hand, the Q-function associated with θ for a pair of $(\mathbf{x}^j, \mathbf{u}')$ can be written as:

$$Q^*_\theta(\mathbf{x}^j, \mathbf{u}') = c_\theta(\mathbf{x}^j, \mathbf{u}') + \sum_{i=1}^{2^d} (M_k(\mathbf{u}'))_{ij} (J^*_\theta)_i. \tag{23}$$

Replacing (21) into (23) leads to the following expression:

$$Q^*_\theta(\mathbf{x}^j, \mathbf{u}') = c_\theta(\mathbf{x}^j, \mathbf{u}') + \gamma \sum_{i=1}^{2^d} (M_k(\mathbf{u}'))_{ij} \sum_{l=1}^{2^d} \mathbf{T}_{il}\, \mathbf{c}_{\pi^*_\theta}(l). \tag{24}$$

Now, assuming that a small change in a particular component of the parameter does not induce a change in a component of the policy, we can use the following approximation

$$\begin{aligned} \frac{\partial Q^*_\theta(\mathbf{x}^j, \mathbf{u}')}{\partial \theta_t} &\approx \frac{\partial c_\theta(\mathbf{x}^j, \mathbf{u}')}{\partial \theta_t} \\ &\quad + \gamma \sum_{i=1}^{2^d} (M_k(\mathbf{u}'))_{ij} \sum_{l=1}^{2^d} \mathbf{T}_{il}\, \frac{\partial c_\theta(\mathbf{x}^l, \pi^*_\theta(\mathbf{x}^l))}{\partial \theta_t}. \end{aligned} \tag{25}$$

The previous calculations specify the gradient $\nabla_\theta \Lambda(\theta)$ in (15), which is then used in the parameter update:

$$\theta_{k+1} = \theta_k + \alpha_k\, \nabla_\theta \Lambda(\theta), \tag{26}$$

where α_k is a step length at time step k.

A schematic diagram and pseudocode of the proposed method are presented in Algorithm 1 and Fig. 1, respectively.

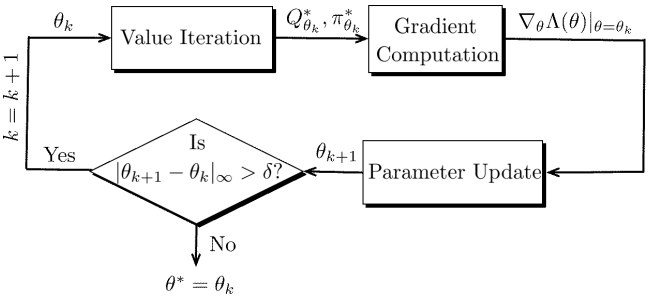

Fig. 1: Schematic diagram of the proposed method for quantification of the immediate cost of GRN control.

Algorithm 1 Proposed method for quantification of the immediate cost of GRN control.

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 [26]:

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

Initial guess θ'.

do

$\theta \leftarrow \theta'$.

Value Iteration

$J_\theta(\mathbf{x}^j) = 0$, for $j = 1, \ldots, 2^d$.

do

$\tilde{J}_\theta(\mathbf{x}^j) = J_\theta(\mathbf{x}^j)$, for $j = 1, \ldots, 2^d$.

For $j = 1, \ldots, 2^d$, do

$$J_\theta(\mathbf{x}^j) = \min_{\mathbf{u} \in \mathbb{U}} \left[c_\theta(\mathbf{x}^j, \mathbf{u}) + \gamma \sum_{i=1}^{2^d} (M(\mathbf{u}))_{ij}\, \tilde{J}_\theta(\mathbf{x}^i) \right]$$

while $\max_{j=1,\ldots,2^d}(|\tilde{J}_\theta(\mathbf{x}^j) - J_\theta(\mathbf{x}^j)|) > \beta$

For $j = 1, \ldots, 2^d$, do:

$$Q^*_\theta(\mathbf{x}^j, \mathbf{u}) = c_\theta(\mathbf{x}^j, \mathbf{u}) + \gamma \sum_{i=1}^{2^d} (M(\mathbf{u}))_{ij}\, J_\theta(\mathbf{x}^i), \text{for } \mathbf{u} \in \mathbb{U},$$

$$\pi^*_\theta(\mathbf{x}^j) = \operatorname*{argmin}_{\mathbf{u} \in \mathbb{U}} \left[c_\theta(\mathbf{x}^j, \mathbf{u}) + \gamma \sum_{i=1}^{2^d} (M(\mathbf{u}))_{ij}\, J_\theta(\mathbf{x}^i) \right].$$

Gradient Computation

$\mathbf{c}_{\pi^*_\theta}(i) = c_\theta\left(\mathbf{x}^i, \pi^*_\theta(\mathbf{x}^i)\right)$, for $i = 1, \ldots, 2^d$.

$(P_{\pi^*_\theta})_{ij} = \left(M_k\left(\pi^*_\theta(\mathbf{x}^j)\right)\right)_{ij}$, $i, j = 1, \ldots, 2^d$.

$\mathbf{T} = (\mathbf{I}_{2^d} - \gamma\, P_{\pi^*_\theta})^{-1}$.

for $t = 1, \ldots, |\theta|$ **do**

for $\mathbf{u} \in \mathbb{U}$ **do**

for $i = 1, \ldots, 2^d$ **do**

$\mathbf{dQ}^t_{\mathbf{u}}(i) = \frac{\partial c_\theta(\mathbf{x}^j, \mathbf{u}')}{\partial \theta_t} + \gamma \sum_{i=1}^{2^d} (M_k(\mathbf{u}'))_{ij} \sum_{l=1}^{2^d} \mathbf{T}_{il}\, \frac{\partial c_\theta(\mathbf{x}^l, \pi^*_\theta(\mathbf{x}^l))}{\partial \theta_t}$

end for

end for

end for

$\mathbf{dLogL}_\theta(t) = 0$, for $t = 1, \ldots, |\theta|$.

for $r = 0, \ldots, T$ **do**

$l_r(\theta) = \frac{\exp\left(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_k, \tilde{\mathbf{u}}_k)\right)}{\sum_{\mathbf{u}' \in \mathbb{U}} \exp\left(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_k, \mathbf{u}')\right)}$.

for $t = 1, \ldots, |\theta|$ **do**

$$\delta_\theta(t) = \delta_\theta(t) - \eta \frac{\exp\left(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_k, \tilde{\mathbf{u}}_k)\right)}{l_r(\theta)\left[\sum_{\mathbf{u}' \in \mathbb{U}} \exp(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_k, \mathbf{u}'))\right]^2} \left[\sum_{\mathbf{u}' \in \mathbb{U}} \exp\left(-\eta\, Q^*_\theta(\tilde{\mathbf{X}}_k, \mathbf{u}')\right)\left(\mathbf{dQ}^t_{\tilde{\mathbf{u}}_k}(\tilde{\mathbf{X}}_k) - \mathbf{dQ}^t_{\mathbf{u}'}(\tilde{\mathbf{X}}_k)\right)\right]$$

end for

end for

Parameter Update

$\theta' \leftarrow \theta + \alpha\, \delta_\theta$.

while $|\theta' - \theta|_\infty > \epsilon$

$\hat{\theta} = \theta$.

Estimatied Cost function $c_{\hat{\theta}}(\mathbf{x}^i, \mathbf{u})$, for $i = 1, \ldots, 2^d$ and $\mathbf{u} \in \mathbb{U}$.

V. Numerical Experiments

In this section, we report the results of numerical experiment using a Boolean model for a gene regulatory network implicated in metastatic melanoma [32]. The network contains 7 genes: WNT5A, pirin, S100P, RET1, MART1, HADHB and STC2. The Boolean regulatory relationships for this network are displayed 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 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)$_{k-1}$ → STC2$_k$=1
(pirin=0, STC2=1)$_{k-1}$ → STC2$_k$=1
(pirin=1, STC2=0)$_{k-1}$ → STC2$_k$=0
(pirin=1, STC2=1)$_{k-1}$ → STC2$_k$=1

TABLE I: Boolean regulatory relationships for the melanoma 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

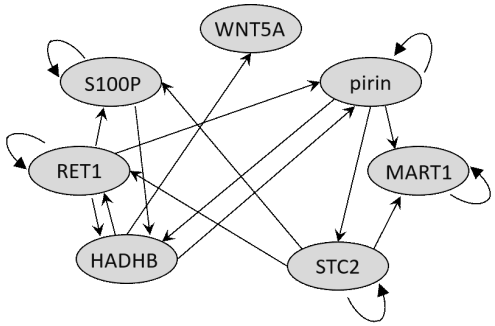

Fig. 2: Melanoma gene regulatory network

The goal is preventing WNT5A gene to be upregulated. For more information about the biological rationale for this, the reader is referred to [32]. In our experiments, the intervention is applied to either RET1 or HADHB. The control space associated to control inputs RET1 and HADHB are as follows:

$$\begin{aligned} \mathbb{U}^{\text{RET1}} &= \{(0,0,0,1,0,0,0), (0,0,0,0,0,0,0)\}, \\ \mathbb{U}^{\text{HADHB}} &= \{(0,0,0,0,0,1,0), (0,0,0,0,0,0,0)\}. \end{aligned} \tag{27}$$

Since the goal of control is preventing WNT5A gene to be upregulated, we assume the following reference immediate

cost function:

$$c(\mathbf{x}^j, \mathbf{u}) = \begin{cases} 5 + \|\mathbf{u}\|_1 & \text{if WNT5A is 1 for state } j, \\ \|\mathbf{u}\|_1 & \text{if WNT5A is 0 for state } j, \end{cases} \quad (28)$$

where the cost of control is assumed to be 1 for any taken intervention and 0 when there is no intervention, and the cost of activation of WNT5A gene is assumed to be 5.

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 this paper, a new version of the "Augmented Lagrange Method" [33] is used for the optimization process.

TABLE II: Parameter values used in all experiments.

Parameter	**Value**
Number of genes d	7
Transition noise intensity p	0.01
Initial probability $P(\mathbf{X}_0 = \mathbf{x}^i)$, $i = 1, \ldots, 128$	1/128
Expression mean $m_j^0, m_j^1, j = 1, \ldots, 7$	30, 60
Expression standard deviation $\sigma_j^0 = \sigma_j^1$, $j = 1, \ldots, 7$	10
Discount factor γ	0.95
Stopping threshold ϵ	0.001
Expert confidence η	0.1, 1, 10
Value iteration threshold β	10^{-6}

We assume the cost function presented in (28) is only partially known,

$$c_\theta(\mathbf{x}^j, \mathbf{u}) = \begin{cases} \theta + \|\mathbf{u}\|_1 & \text{if WNT5A is 1 for state } j, \\ \|\mathbf{u}\|_1 & \text{if WNT5A is 0 for state } j, \end{cases} \quad (29)$$

where the parameter θ denotes the (unknown) cost of observing activation of WNT5A (i.e. undesirable states). On the other hand, we assume that the expert confidence parameter η in (8) is known.

RET1 gene is considered as a control gene and the process noise is assumed to be $p = 0.01$. The average absolute difference between the estimated parameter $\hat{\theta}^*$ computed by the proposed method (Algorithm1) and the reference parameter $\theta^* = 5$ for various choices of η and different lengths of expert's sequence T is displayed in Fig.3. One can see that the average difference is converging to zero as the length of expert's sequence increases. The effect of parameter η is also presented in Fig. 3. As mentioned previously, the larger η is, the more accurate the expert's policy will be, and therefore the average difference is smaller for larger η.

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 [20], [34], to shift the dynamics of the melanoma network away from states associated with metastasis.

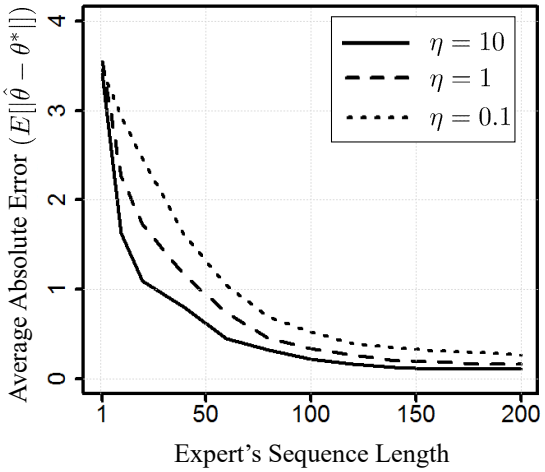

Fig. 3: The average absolute difference between the estimated parameter $\hat{\theta}$ and true value $\theta^* = 5$.

TABLE III: Average results for V_BKF method with both known and unknown immediate cost functions.

		V_BKF with Unk. Cost			
Control	p	$T = 50$	$T = 100$	V_BKF	No-Control
RET1	0.01	0.19	0.16	0.12	2.28
	0.05	0.80	0.75	0.67	2.33
HADHB	0.01	0.50	0.44	0.38	2.28
	0.05	1.33	1.20	1.11	2.33

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.

In this part of experiment, the performance of the V_BKF with unknown and known immediate cost functions as well as the system without control are compared. Two different expert's sequence lengths are considered here: $T = 50$, and $T = 100$. RET1 and HADHB are both considered as control genes.

We can observe that the performance of the V_BKF with unknown immediate cost function is better for larger expert sequence lengths. The reason is that the parameters of the immediate cost function under larger expert sequences can be estimated better which can lead to better performance of control process. For large process noise intensities, the performance of all cases decreases. This reduction is more obvious for system with unknown cost function and specifically small expert sequence. Moreover, the RET1 gene seems to be a better control input for reducing the activation of WNT5A, as lower cost can be seen under the RET1 control gene in comparison to the HADHB gene in all cases.

VI. Conclusion

In this paper, we proposed a methodology for quantification of the cost function of the gene regulatory networks (GRNs) through an expert's behavior. The well-known Boolean network model was employed for modeling GRNs. Given a single sequence of Boolean states and interventions by an expert, we proposed a maximum likelihood approach for quantification of the cost function based on the inverse reinforcement learning technique. The ability of the proposed methodology to obtain a good control policy was demonstrated by numerical experiments involving a Boolean model of a melanoma gene regulatory network.

References

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

[2] T. Chen, H. L. He, G. M. Church, *et al.*, "Modeling gene expression with differential equations.," *Pacific symposium on biocomputing*, vol. 4, no. 29, p. 40, 1999.

[3] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita, "Dynamic modeling of genetic networks using genetic algorithm and S-system," *Bioinformatics*, vol. 19, no. 5, pp. 643–650, 2003.

[4] N. Friedman, M. Linial, I. Nachman, and D. Pe'er, "Using Bayesian networks to analyze expression data," *Journal of computational biology*, vol. 7, no. 3-4, pp. 601–620, 2000.

[5] 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.

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

[7] M. Imani and U. Braga-Neto, "Control of gene regulatory networks with noisy measurements and uncertain inputs," *IEEE Transactions on Control of Network Systems*, 2018.

[8] D. Laschov and M. Margaliot, "A maximum principle for single-input Boolean control networks," *IEEE Transactions on Automatic Control*, vol. 56, no. 4, pp. 913–917, 2011.

[9] R. Albert and H. G. Othmer, "The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster," *Journal of theoretical biology*, vol. 223, no. 1, pp. 1–18, 2003.

[10] M. I. Davidich and S. Bornholdt, "Boolean network model predicts cell cycle sequence of fission yeast," *PloS one*, vol. 3, no. 2, p. e1672, 2008.

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

[12] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, "The yeast cell-cycle network is robustly designed," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 14, pp. 4781–4786, 2004.

[13] I. Shmulevich and E. R. Dougherty, *Genomic signal processing*. Princeton University Press, 2014.

[14] I. Shmulevich and E. Dougherty, "Probabilistic Boolean networks: The modeling and control of gene regulatory networks, siamsociety for industrial and applied mathematics," *Philadelphia, PA*, 2009.

[15] I. Shmulevich, E. R. Dougherty, and W. Zhang, "From Boolean to probabilistic Boolean networks as models of genetic regulatory networks," *Proceedings of the IEEE*, vol. 90, no. 11, pp. 1778–1792, 2002.

[16] D. Cheng and H. Qi, "A linear representation of dynamics of Boolean networks," *IEEE Transactions on Automatic Control*, vol. 55, no. 10, pp. 2251–2258, 2010.

[17] D. Cheng, H. Qi, and Z. Li, *Analysis and control of Boolean networks: a semi-tensor product approach*. Springer Science & Business Media, 2010.

[18] M. Imani and U. Braga-Neto, "Maximum-likelihood adaptive filter for partially-observed Boolean dynamical systems," *IEEE transaction on Signal Processing*, vol. 65, pp. 359–371, 2017.

[19] M. Imani and U. Braga-Neto, "Particle filters for partially-observed Boolean dynamical systems," *Automatica*, 2017.

[20] M. Imani and U. Braga-Neto, "Multiple model adaptive controller for partially-observed Boolean dynamical systems," in *Proceedings of the 2017 American Control Conference (ACC 2017), Seattle, WA*, 2017.

[21] M. Imani and U. 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*, pp. 4208–4213, IEEE, 2016.

[22] A. Y. Ng, S. J. Russell, *et al.*, "Algorithms for inverse reinforcement learning.," in *Icml*, pp. 663–670, 2000.

[23] Y. Chen, E. R. Dougherty, and M. L. Bittner, "Ratio-based decisions and the quantitative analysis of cDNA microarray images," *Journal of Biomedical optics*, vol. 2, no. 4, pp. 364–374, 1997.

[24] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold, "Mapping and quantifying mammalian transcriptomes by RNA-seq," *Nature methods*, vol. 5, no. 7, pp. 621–628, 2008.

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

[26] M. Imani and U. 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)*, pp. 972–976, IEEE, 2015.

[27] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning.," in *AAAI*, vol. 8, pp. 1433–1438, Chicago, IL, USA, 2008.

[28] D. Ramachandran and E. Amir, "Bayesian inverse reinforcement learning," *Urbana*, vol. 51, no. 61801, pp. 1–4, 2007.

[29] G. Neu and C. Szepesvári, "Apprenticeship learning using inverse reinforcement learning and gradient methods," *arXiv preprint arXiv:1206.5264*, 2012.

[30] M. Lopes, F. Melo, and L. Montesano, "Active learning for reward estimation in inverse reinforcement learning," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 31–46, Springer, 2009.

[31] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998.

[32] E. R. Dougherty, R. Pal, X. Qian, M. L. Bittner, and A. Datta, "Stationary and structural control in gene regulatory networks: basic concepts," *International Journal of Systems Science*, vol. 41, no. 1, pp. 5–16, 2010.

[33] E. G. Birgin and J. M. Martínez, "Improving ultimate convergence of an augmented Lagrangian method," *Optimization Methods and Software*, vol. 23, no. 2, pp. 177–195, 2008.

[34] M. Imani and U. Braga-Neto, "State-feedback control of partially-observed Boolean dynamical systems using RNA-seq time series data," in *American Control Conference (ACC), 2016*, pp. 227–232, IEEE, 2016.