BIOINFORMATICS ORIGINAL PAPER
Vol. 26 no. 12 2010, pages 1556–1563
doi:10.1093/bioinformatics/btq225

Systems biology
Advance Access publication April 25, 2010

A CoD-based reduction algorithm for designing stationary control policies on Boolean networks

Noushin Ghaffari[1], Ivan Ivanov[2], Xiaoning Qian[3] and Edward R. Dougherty[1,4,*]

[1]Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843,
[2]Department of Veterinary Physiology and Pharmacology, Texas A&M University, College Station, TX 77843,
[3]Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620 and
[4]Translational Genomics Research Institute, 400 N 5th Street, Suite 1600, Phoenix, AZ 85004, USA

Associate Editor: Olga Troyanskaya

ABSTRACT

Motivation: Gene regulatory networks serve as models from which to derive therapeutic intervention strategies, in particular, stationary control policies over time that shift the probability mass of the steady state distribution (SSD) away from states associated with undesirable phenotypes. Derivation of control policies is hindered by the high-dimensional state spaces associated with gene regulatory networks. Hence, network reduction is a fundamental issue for intervention.

Results: The network model that has been most used for the study of intervention in gene regulatory networks is the *probabilistic Boolean network* (PBN), which is a collection of constituent Boolean networks (BNs) with perturbation. In this article, we propose an algorithm that reduces a BN with perturbation, designs a control policy on the reduced network and then induces that policy to the original network. The *coefficient of determination* (CoD) is used to choose a gene for deletion, and a reduction mapping is used to rewire the remaining genes. This CoD-reduction procedure is used to construct a reduced network, then either the previously proposed mean first-passage time (MFPT) or SSD stationary control policy is designed on the reduced network, and these policies are induced to the original network. The efficacy of the overall algorithm is demonstrated on networks of 10 genes or less, where it is possible to compare the steady state shifts of the induced and original policies (because the latter can be derived), and by applying it to a 17-gene gastrointestinal network where it is shown that there is substantial beneficial steady state shift.

Availability: The code for the algorithms is available at: http://gsp.tamu.edu/Publications/supplementary/ghaffari10a/
Please contact Noushin Ghaffari at nghaffari@tamu.edu for further questions.

Contact: edward@ece.tamu.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on August 30, 2009; revised on April 10, 2010; accepted on April 20, 2010

1 INTRODUCTION

A key objective for modeling gene regulatory networks is to derive intervention strategies for beneficially altering cell dynamics (Datta *et al.*, 2007). To address the issue of changing the long-run behavior, stochastic control has been employed to find stationary control policies that affect a network's steady state distribution (SSD; Pal *et al.*, 2006); however, owing to the inherent computational complexity of optimal control methods (Akutsu *et al.*, 2007; Bertsekas, 2005), it is often infeasible to design optimal control policies for large networks. A possible approach to complexity reduction in the finite-horizon model is to use a discrete linear model (Ng *et al.*, 2006); however, we do not wish to be constrained to a finite horizon or a linear model. We proceed in the nonlinear case with network dynamics described by a Markov chain. Even restricting ourselves to the Boolean model, where gene states are binary, 0 and 1, if there are n genes in the network, then there are 2^n states and the transition probability matrix is of size $2^n \times 2^n$.

Given this level of complexity, even approximation via re-inforcement learning (Faryabi *et al.*, 2007) and greedy control methods (Qian *et al.*, 2009; Vahedi *et al.*, 2008) are quite restricted in the size of networks they can handle. For instance, rather than doing a full optimization relative to some objective function and facing the 'curse of dimensionality' associated with dynamic programming, greedy methods utilize statistical characteristics of the network, including mean first-passage times (MFPT; Vahedi *et al.*, 2008) and the SSD (Qian *et al.*, 2009). But these still require manipulating the transition probability matrix, which effectively limits their use to not more that 13-gene networks using our current workstation computing environment.

This article takes the approach of reducing the size of the network, designing a control policy on the reduced network, and from this inducing a control policy on the full network. It is motivated by a previously proposed network reduction algorithm that removes genes in such a way that the deleted gene induces a specific collapsing of pairs of states from the state space of the original network (Ivanov and Dougherty, 2004). A more detailed discussion about the reduction mapping from that article and its comparison to the current proposed algorithm for designing stationary control policies is provided in the Supplementary Material. While other reduction algorithms have been developed to obtain reduced models for Boolean or probabilistic Boolean networks (PBNs) to maintain either the structural consistency (Shmulevich and Dougherty, 2007) or the dynamical behavior of the original network (Ivanov *et al.*, 2007), the specific intent in this article is to find a reduction strategy

[*]To whom correspondence should be addressed.

that can provide beneficial stationary control policies for the original network.

In this study, we use Boolean networks (BNs) with perturbation, BN$_{p}s$, to model gene regulatory networks and ultimately design stationary control policies to alter the dynamics of the network. BNs have been used in variety of other contexts and with different objectives in biological applications. Huang (2007) generates Boolean gene regulatory networks for validating the hypothesis that the attractors of such networks represent functional cellular states, whereas Kauffman model (Kauffman, 1993) proposes that the cell types are the attractors. He introduces randomization into the networks in terms of environmental noise (random perturbation of the individual genes) and mutation, which refers to changes in the wiring of the network. In another prospective, BNs are claimed to have biologically meaningful behavior when they are in the critical phase (Kauffman, 1993). Ramo *et al.* (2006) study this phenomena and discover that gene regulatory networks have stable, near critical, dynamics. They also propose an analytical approximation of the size distribution of perturbation avalanches for BNs using the branching process. They test their analytical method with simulations using different synthetic BNs and also a biological dataset. Random BNs and their characteristics have been extensively studied by Aldana *et al.* (2002). In a random BN, the average function indegrees are constant and function outputs are assigned randomly. Serra *et al.* (2004) study the effects of perturbation in the context of random BNs by knocking out one gene. Additionally, the attractor structure of BN$_{p}$s (Brun *et al.*, 2005), inference of BN$_{p}$s (Yu *et al.*, 2009), and their use as models for gene regulation, in particular, intervention in BN$_{p}$s (Dougherty *et al.*, 2010; Qian and Dougherty, 2009), have been studied.

Complexity reduction has been considered for other classes of models, including discrete network models such as BNs (Ivanov and Dougherty, 2004; Ivanov *et al.*, 2007) or logical regulatory graphs (Naldi *et al.*, 2009), and continuous biochemical networks (Ball *et al.*, 2006; Borisov *et al.*, 2005; Clarke, 1992; Conzelmann *et al.*, 2001, 2006; Gorban *et al.*, 2006; Hartwell *et al.*, 1999; Indic *et al.*, 2006; Radulescu *et al.*, 2008; Saez-Rodriguez *et al.*, 2005; Wang *et al.*, 2004). All past efforts focus on reducing the complexity while preserving network dynamics, either by maintaining the attractor structures as in discrete mathematical frameworks (Ivanov *et al.*, 2007; Naldi *et al.*, 2009) or partitioning large systems into smaller subsystems to enable better analysis and understanding for continuous network models as in Conzelmann *et al.* (2001), Wang *et al.* (2004), Indic *et al.* (2006), Borisov *et al.* (2005), Conzelmann *et al.* (2006), Clarke (1992), Ball *et al.* (2006), Radulescu *et al.* (2008), Hartwell *et al.* (1999), Saez-Rodriguez *et al.* (2005) and Gorban *et al.* (2006). In this article, we focus on BNs with perturbation owing to their role in modeling gene regulatory networks, a key point being that their dynamics can be modeled using Markov chains; thereby facilitating the development of control policies that can shift the network SSD toward desirable states. The main objective of our work is to design suboptimal stationary control policies and our proposed reduction mappings provide the means to this end. Although the literature mainly focuses on reduction of biochemical networks in a continuous simulation framework (Clarke, 1992; Conzelmann *et al.*, 2001, 2006; Gorban *et al.*, 2006; Radulescu *et al.*, 2008; Saez-Rodriguez *et al.*, 2005), while our work models gene regulatory networks in a discrete mathematical framework, the idea of partitioning large systems into multiscale or hierarchical small subsystems could be an interesting future research direction on model reduction for BN-based gene regulatory networks if one can suitably abstract the relationships within subnetworks.

As typically formulated, the control problem is characterized by a target gene whose expression is to be altered by the control policy and one or more control genes whose expressions are altered by intervention. For instance, in the well-studied control model for gene *WNT5A* in the case of metastatic melanoma, the upregulation of *WNT5A* is associated with increased metastatic competence, so that the goal of the control policy is to downregulate *WNT5A* (Shmulevich and Dougherty, 2007). The control gene in the *WNT5A* network is *pirin*. The control policy acts by observing the state of the network at each time point and, based on the state, decides whether to alter the value of the control gene.

From our perspective here, were the network is too large to derive the optimal control policy, we would like to delete one or more genes from the network (neither the target nor the control gene) so that we could derive a policy on the reduced network that would induce a suboptimal policy on the original network. There are four basic steps in this procedure: (i) apply an algorithm to the network to select a gene for deletion; (ii) apply an algorithm to construct the gene logic for the reduced network; (iii) apply a control algorithm to the reduced network to derive a control policy on the reduced network; and (iv) induce a control policy on the original network based on the control policy derived for the reduced network. Step 1 can be amended by selecting more than one gene for deletion. The method proposed herein employs the coefficient of determination (CoD; Dougherty *et al.*, 2000) to choose genes for deletion, adapts the collapsing heuristic of Ivanov and Dougherty (2004) to construct the wiring of the reduced network, designs a control policy on the reduced network using either the MFPT control policy (Vahedi *et al.*, 2008) or the SSD control policy (Qian *et al.*, 2009), and finishes with a procedure to induce a control policy on the original network. The MFPT and SSD control policies have been chosen on account of their computational efficiency and generally satisfactory performance; however, other control policies could be applied on the reduced network.

Performance of the CoD-based reduction procedure is evaluated by its effects on the SSD and on how well it approximates the stationary control designed on the full network. The main effect on the SSD that interests us is the shift of the probability mass toward the desirable states. This shift is computed as the absolute value of the difference between the SSDs of the network before and after applying the control policy. The algorithm is formulated for BNs (Huang, 1999; Kauffman, 1993, 1969) with perturbation; however, since a binary context-sensitive PBN (Brun *et al.*, 2005; Shmulevich *et al.*, 2002) is a collection of BNs with perturbation endowed with a selection probability structure (and a general PBN is a collection of more finely quantized versions of BNs), the algorithm can be applied to a PBN by applying it to each constituent BN for the same gene, thereby reducing the PBN. As formulated in this article, the control problem involves a single control gene and a single target gene, but there is no restriction relative to either the number of control genes or target genes insofar as the reduction algorithm is concerned, so long as the control algorithm applied on the reduced network has no such restrictions. Neither the MFPT nor the SSD control algorithms have such restrictions.

2 SYSTEMS AND METHODS

2.1 BNs

A *BN with perturbation p*, $\mathrm{BN}_p = (V, \mathbf{f})$, on n genes is defined by a set of nodes $V = \{x_1, \ldots, x_n\}$ and a vector of Boolean functions $\mathbf{f} = [f^1, \ldots, f^n]$. The variable $x_i \in \{0,1\}$ represents the expression level of gene i, with 1 representing high and 0 representing low expression (Shmulevich *et al.*, 2002). $\mathbf{f}$ represents the regulatory rules between genes. At every time step, the value of x_i is predicted by the values of a set, W_i, of genes at the previous time step, based on the regulatory function f^i. $W_i = \{x_{i_1}, \ldots, x_{i_{k_i}}\}$ is called the *predictor set* and the function f^i is called the *predictor function* of x_i. A state of the BN_p is a vector $\mathbf{s} = (x_1, \ldots, x_n) \in \{0,1\}^n$, and the *state space* of the BN_p is the collection S of all states of the network. The perturbation parameter $p \in (0,1]$ models random gene mutations, i.e. at each time point there is a probability p of any gene changing its value uniformly randomly. The underlying model of a BN_p is a finite Markov chain and its dynamics are completely described by its $2^n \times 2^n$ transition probability matrix, $P = (p(\mathbf{s}_i, \mathbf{s}_j))_{i,j=1}^{2^n}$, where $p(\mathbf{s}_i, \mathbf{s}_j)$ is the probability of the chain undergoing the transition from the state $\mathbf{s}_i$ to the state $\mathbf{s}_j$. The perturbation probability p makes the chain ergodic and therefore it possesses a steady state probability distribution π defined by $\pi^T P = \pi^T$, where T denotes transpose.

2.2 CoD

CoD measures how a set of random variables improves the prediction of a target variable, relative to the best prediction in the absence of any conditioning observation (Dougherty *et al.*, 2000). Let $\mathbf{X} = (X_1, X_2, \ldots, X_n)$ be a vector of predictor binary random variables, Y a binary target variable and f a Boolean function such that $f(\mathbf{X})$ predicts Y. The mean square error of $f(\mathbf{X})$ as a predictor of Y is the expected squared difference, $E[|f(\mathbf{X}) - Y|^2]$. Let $\varepsilon_{\mathrm{opt}}(Y, \mathbf{X})$ be the minimum MSE among all predictor functions $f(\mathbf{X})$ for Y and $\varepsilon_0(Y)$ be the error of the best estimate of Y without any predictors. The CoD is defined as

$$\mathrm{CoD}_{\mathbf{X}}(Y) = \frac{\varepsilon_0(Y) - \varepsilon_{\mathrm{opt}}(Y, \mathbf{X})}{\varepsilon_0(Y)}. \tag{1}$$

Letting $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{2^n}$ denote the 2^n possible values for $\mathbf{X}$, running from $(0,0,\ldots,0)$ to $(1,1,\ldots,1)$, the relevant quantities are given by

$$\varepsilon_{\mathrm{opt}}(Y, \mathbf{X}) = \sum_{j=1}^{2^n} P(\mathbf{X} = \mathbf{x}_j) \min[P(Y=0|\mathbf{x}_j), P(Y=1|\mathbf{x}_j)] \tag{2}$$

and

$$\varepsilon_0(Y) = \min[P(Y=0), P(Y=1)] \tag{3}$$

(Dougherty *et al.*, 2000). For more information and a numerical example, refer to Supplementary Material. The CoD determines the strength of the connection between a target gene and its predictors and has been used since the early days of DNA microarray analysis to characterize the nonlinear multivariate interaction between genes (Kim *et al.*, 2000) and has more recently been used to characterize canalizing genes (Martins *et al.*, 2008) and contextual genomic regulation (Dougherty *et al.*, 2009). We have restricted ourselves to the Boolean case; thereby arriving at the preceding representations of $\varepsilon_{\mathrm{opt}}(Y, X)$ and $\varepsilon_0(Y)$; however, the basic definition for $\mathrm{CoD}_X(Y)$ is not so restricted (Dougherty *et al.*, 2000).

2.3 MFPT control policy

Optimal intervention is usually formulated as an optimal stochastic control problem (Bertsekas, 2005). We focus on intervention via a single control gene g, and stationary control policies $\mu_g : S \to \{0,1\}$ based on g. The values 0/1 are interpreted as off/on for the application of the control: 1 meaning that the current value of g is flipped, and 0 meaning that no control is applied.

The *MFPT* policy is based on the comparison between the MFPTs of a state $\mathbf{s}$ and its flipped (with respect to g) state $\tilde{\mathbf{s}}^g$ (Vahedi *et al.*, 2008). When considering intervention, the state space S can be partitioned into desirable D and undesirable U states according to the expression values of a given *target* set W of genes. For simplicity, we assume $W = \{x\}$, the target gene x is the leftmost gene in the state's binary representations, i.e. $x_1 = x$, $\mathbf{s} = (x, x_2, \ldots, x_n)$, and the desirable states correspond to the value $x = 0$. With these assumptions, the transition probability matrix P of the network can be written as

$$P = \begin{pmatrix} P_{DD} & P_{DU} \\ P_{UD} & P_{UU} \end{pmatrix} \tag{4}$$

Using this representation, one can compute the MFPT required for a state $\mathbf{s}$ to reach the boundary between desirable and undesirable states. Computation of these average times is performed in the time scale used for the state transitions of the network. If one uses the states of the network to index the components of the vectors in the 2^n-dimensional Euclidean space $\mathbb{R}^{2^n}$, then one can form the vectors K_U and K_D that contain the MFPTs needed for the states in D and U to reach the undesirable and the desirable states, respectively. For example, the coordinate $K_D(\mathbf{s})$ of K_D gives the MFPT for the undesirable state $\mathbf{s}$ to reach the set D of desirable states. The two vectors K_U and K_D are of dimension 2^{n-1}, and, according to a well-known result from the theory of Markov chains (Norris, 1998), are given as solutions to the following system of linear equations:

$$K_U = e + P_{DD} K_U \tag{5}$$

$$K_D = e + P_{UU} K_D \tag{6}$$

where e denotes the vector of dimension 2^{n-1} with all of its coordinates equal to 1. For more details on the definitions and theorems behind MFPT equations, refer to the Supplementary Material. To understand the intuition behind the MFPT algorithm, it is important to notice that, because the control gene g is different from the target gene, every state $\mathbf{s}$ belongs to the same class of states, D or U, as its flipped state $\tilde{\mathbf{s}}^g$. With this in mind, if a desirable state $\mathbf{s}$ reaches U on average faster than $\tilde{\mathbf{s}}^g$, it is reasonable to apply control and start the next network transition from its flipped state $\tilde{\mathbf{s}}^g$. Thus, the design of the stationary MFPT control policy is based on the differences $K_D(\mathbf{s}) - K_D(\tilde{\mathbf{s}}^g)$ and $K_U(\tilde{\mathbf{s}}^g) - K_U(\mathbf{s})$. To avoid too frequent application of control, the MFPT algorithm uses a tuning parameter $\gamma > 0$, and these differences are compared with the value of γ, which is related to the cost of applying control. For example, γ is set to a larger value when the ratio of the cost of control to the cost of the undesirable states is higher, the intent being to apply the control less frequently (Vahedi *et al.*, 2008).

2.4 SSD control policy

The *SSD* policy (Qian *et al.*, 2009) uses the analytic formulation of the steady state mass of a perturbed Markov chain given in Qian and Dougherty (2008) to quantify the shifted steady state mass after applying possible controls. A perturbation (change) in the logic defining a BN results in the original transition probability matrix P and SSD π being changed to $\tilde{P}$ and $\tilde{\pi}$, respectively. In Qian and Dougherty (2008), the fundamental matrix, Z, is used to represent $\tilde{\pi}$ in terms of π. $Z = [I - P + e\pi^T]^{-1}$, where T denotes transpose and e is a column vector whose components are all unity (Schweitzer, 1968). For a *rank-one perturbation*, the perturbed Markov chain has the transition matrix $\tilde{P} = P + ab^T$, where a, b are two arbitrary vectors satisfying $b^T e = 0$, and ab^T represents a rank-one perturbation to the original Markov chain P. In the special case where the transition mechanisms before and after perturbation differ only in one state, say state $\mathbf{k}$,

$$\tilde{\pi}^T = \pi^T + \frac{\pi^T e(\mathbf{k})}{1 - b^T Z e(\mathbf{k})} b^T Z = \pi^T + \frac{\pi(\mathbf{k})}{1 - \beta(\mathbf{k})} \beta^T \tag{7}$$

where $\beta^T = b^T Z$ and $e(\mathbf{k})$ is the elementary vector with a 1 in the $\mathbf{k}$-th position and 0s elsewhere (Hunter, 2005; Qian and Dougherty, 2008; Schweitzer, 1968).

The results for these special cases can be extended to arbitrary types of perturbations so that it is possible to compute the SSDs of arbitrarily perturbed Markov chains in an iterative fashion (Qian and Dougherty, 2008).

To define the SSD control policy, let state $\tilde{\mathbf{s}}^g$ be the flipped state (with respect to control gene g) corresponding to state $\mathbf{s}$ (as with the MFPT control policy). Let π_U be the original steady state mass of the undesirable states and let $\tilde{\pi}_U(\mathbf{s})$ and $\tilde{\pi}_U(\tilde{\mathbf{s}}^g)$ denote the steady state masses of the undesirable states resulting from altering the original transition probability matrix by forcing $\mathbf{s}$ to $\tilde{\mathbf{s}}^g$ and forcing $\tilde{\mathbf{s}}^g$ to $\mathbf{s}$, respectively. The SSD policy is defined on pairs of states, $\mathbf{s}$ and $\tilde{\mathbf{s}}^g$, in the following manner: if both $\tilde{\pi}_U(\mathbf{s})$ and $\tilde{\pi}_U(\tilde{\mathbf{s}}^g)$ are larger than π_U, then control is applied to neither; otherwise, if $\tilde{\pi}_U(\mathbf{s}) \le \tilde{\pi}_U(\tilde{\mathbf{s}}^g)$, then control is applied to $\mathbf{s}$, and if $\tilde{\pi}_U(\mathbf{s}) > \tilde{\pi}_U(\tilde{\mathbf{s}}^g)$, then control is applied to $\tilde{\mathbf{s}}^g$.

2.5 Selection policies

Assuming gene d is to be deleted from the network, a reduction mapping can be used to define the transition rules for states in the reduced network (Ivanov and Dougherty, 2004). The critical issue is how to design the reduction mapping. For every two states $\tilde{\mathbf{s}}^d$ and $\mathbf{s}$ that differ only in the value of d, consider the states to which they transition: $\mathbf{s} \to \mathbf{w}$ and $\tilde{\mathbf{s}}^d \to \mathbf{v}$. Following deletion, d becomes a latent variable and, under the reduction mapping, the states $\mathbf{s}$ and $\tilde{\mathbf{s}}^d$ collapse to a state $\check{\mathbf{s}}$ in the reduced network. The state $\check{\mathbf{s}}$ is obtained from either $\mathbf{s}$ or $\tilde{\mathbf{s}}^d$ by removing their d-th coordinate. The design of a reduction mapping is equivalent to filling in the entries of the truth table of the reduced network by selecting one of the two possible transitions $\check{\mathbf{s}} \to \check{\mathbf{w}}$ or $\check{\mathbf{s}} \to \check{\mathbf{v}}$, where the states $\check{\mathbf{w}}$ and $\check{\mathbf{v}}$ are the collapsed states in the reduced network that correspond to $\mathbf{w}$ and $\mathbf{v}$ respectively. Thus, one arrives at the following definition:

Definition 1. *A* selection policy ν^d *corresponding to the deleted gene d is a* 2^n*-dimensional vector,* $\nu^d \in \{0,1\}^{2^n}$*, indexed by the states of S and having components equal to* 1 *at exactly one of the positions corresponding to each pair* $(\mathbf{s},\tilde{\mathbf{s}}^d)$*,* $\mathbf{s} \in S$.

For each gene d there are $2^{2^{n-1}}$ different selection policies. Using this definition, the reduction mapping Π_{ν^d} corresponding to the selection policy ν^d is constructed by selecting the transition $\check{\mathbf{s}} \to \check{\mathbf{w}}$ if $\nu^d(\mathbf{s}) = 1$ or $\check{\mathbf{s}} \to \check{\mathbf{v}}$, otherwise. The goodness of a selection policy (in its own right) depends on how it preserves structural relationships of the original network in the reduced network, for instance, steady state behavior and control policies.

3 ALGORITHM

The procedure we propose has three aspects. First, the original network must be reduced. This entails two steps: gene deletion and a reduction mapping to construct the regulatory structure for the remaining genes. Second a control policy must be designed on the reduced network. We will use either the MFPT or SSD algorithm to design the control policy. Lastly, a control policy for the original network must be induced from the control policy designed on the reduced network. We refer to the overall algorithm as *CoD-Reduce*.

3.1 Deletion

The algorithm selects the gene to delete based on two criteria. If there are genes isolated from the rest of the genes, then the algorithm randomly selects one of them as the candidate for deletion. A gene is called isolated if it does not predict any other genes and no other gene predicts it. Otherwise, the combination of three genes that has the smallest steady state CoD in determining the target gene is found and the gene chosen for deletion is the one with the weakest influence in terms of CoD value on the target gene from that triple of genes. We have based the CoD on triples of genes because, as Kauffman points out, the average connectivity of the model cannot be too high if its dynamics is not chaotic (Kauffman, 1993), and threee-predictor connectivity is commonly assumed in BN and PBN modeling (Shmulevich and Dougherty, 2007). The procedure we have adopted ensures that the candidate gene for deletion from the network has small influence on the target gene if the model has reached its SSD. The deletion procedure is described in detail in Algorithm 1.

Algorithm 1 CoD-Reduce: Selecting Best Gene for Deletion

```
Create connectivity table of the BN on $n$ genes
Exclude self-predictions from the connectivity table
Compute set $C = \{c_1, c_2, ..., c_n\}$, where each $c_i$ is the total number of genes that predict $g_i$ or being predicted by $g_i$
Find all $c_i = 0$ and put their corresponding $g_i$ in the constant gene set: CONSTANT
if CONSTANT $\neq \emptyset$ then
    GENE for DELETION $\leftarrow$ randomly selected gene from the CONSTANT
else
    Compute set COMBINATIONS: includes all 3-gene combinations, excluding the target gene
    for all the sets in COMBINATIONS do
        $\Theta_j \leftarrow$ CoD of the 3-gene set $j$ w.r.t. target gene
    end for
    Find a 3-gene combination with minimum $\Theta_j$: MINCOD
    GENE for DELETION $\leftarrow g_i \in$ MINCOD with minimum individual CoD w.r.t target gene
end if
return (GENE for DELETION)
```

Finding an optimal selection policy would require testing each one of the $2^{2^{n-1}}$ possible reduced networks, which is computationally infeasible for large networks. In the present article, a heuristically chosen selection policy is combined with an inducement procedure to design a control policy on the original network. The selection policy we use here is designed by considering the SSD of the network. The intuition behind this approach relies on two facts: first, attractors are an essential part of the network and therefore should be preserved during the reduction; second, states with larger steady state probabilities are more likely to be visited during the long-run transitions of the network. Based on these considerations, the selection policy proceeds as follows: for states $\mathbf{s}$ and $\tilde{\mathbf{s}}^d$ that only differ in the deleted gene d, the state transitions of the states possessing larger steady state probability mass will be kept as transitions for the reduced states, excluding the gene for deletion; however, if either $\mathbf{s}$ or $\tilde{\mathbf{s}}^d$ is an attractor and the other is not, then the attractor state is chosen to determine the function structure. Algorithm 2 represents the steps of reduction mapping. This selection policy has 1 for the states whose functions are kept as the result of reduction and 0 for the rest. If two states that differ only in their d coordinate, then only one of them can have a 1 in the corresponding selection policy.

3.2 Inducement

Suppose the original network has n genes, the reduced network has $m < n$ genes based on $n-m$ deletion–reduction applications, and, without loss of generality, suppose the last $n-m$ genes have been deleted. Then, for any state $(x_1, x_2, \ldots, x_m)$ in the reduced network, there are 2^{n-m} states in the original network of the form $(x_1, \ldots, x_m, z_1, \ldots, z_{n-m})$. If μ_{red} is the control policy designed on the reduced network, then the induced policy on the original network is defined by

$$\mu_{ori}(x_1,\ldots,x_m,z_1,\ldots,z_{n-m})=\mu_{red}(x_1,x_2,\ldots,x_m) \qquad (8)$$

for any $z_1,\ldots,z_{n-m}\in\{0,1\}$.

Algorithm 2 CoD-Reduce: Reduction Mapping

```
Put all the attractor states in a set called: ATTRACTORS
Find the SSD of the network: $\pi$
for all the states s in the state space do
    find its flipped state w.r.t. gene for deletion: $\tilde{s}^d$
    if (($\mathbf{s}\in ATTRACTORS$) && ($\tilde{s}^d \notin ATTRACTORS$)) then
        Selection Policy (s) = 1
        Selection Policy ($\tilde{s}^d$) = 0
    else if (($\mathbf{s}\notin ATTRACTORS$) && ($\tilde{s}^d \in ATTRACTORS$)) then
        Selection Policy (s) = 0
        Selection Policy ($\tilde{s}^d$) = 1
    else
        if ($\pi(s) > \pi(\tilde{s}^d)$) then
            Selection Policy (s) = 1
            Selection Policy ($\tilde{s}^d$) = 0
        else
            Selection Policy (s) = 0
            Selection Policy ($\tilde{s}^d$) = 1
        end if
    end if
end for
for all the states s in the state space that have (Selection Policy (s) = 1) do
    Keep the transitions of the s excluding the d coordinate as the transitions of $\check{s}$: reduced state
end for
```

4 DISCUSSION

We have carried out several simulations to study the performance of the CoD-Reduce algorithm.

4.1 Randomly generated networks

An inherent difficulty in a simulation study on the performance of *CoD-Reduce* is how to choose the target and control genes. In practice, the target gene is chosen in such a way that its behavior is closely related to the phenotype of interest and the control gene can either be selected via biological knowledge or according to some criterion related to its ability to control the target gene. If it were not for the computational burden, one might simply choose all non-target genes as control genes and see which one has the largest beneficial impact. One approach that has been taken to identify a good control gene in the case of both finite- and infinite-horizon dynamic programming-based control is to choose the gene whose *influence* is the greatest with respect to the target gene (Pal *et al.*, 2006). Here, we use the fact that the CoD provides a measure of gene interdependence and use it to choose the control gene. Specifically, we can compute for each non-target gene the CoD as a predictor of the target gene and choose the control gene to be the one with the largest CoD. It is important to recognize that a strong CoD for gene g_1 predicting gene g_2 does not imply that g_1 and g_2 are directly connected in the model or, if the model were inferred from expression data, that they are directly biochemically connected, nor does it mean that g_1 is the best control gene for g_2 based on the MFPT control policy or, for that matter, any other control policy.

In random network simulations, we also have the issue of how to choose the target gene, since the randomly generated networks are of a purely computational nature. Since we are interested in networks in which the target gene is controllable, we need to choose a target gene for which there exists a non-target gene that can exercise control over it. A simple way to do that is to consider all gene-to-gene CoDs and pick the control and target genes to be the two genes possessing the largest gene-to-gene CoD, the former being the control gene and the latter being the target gene. While it is true that this choice provides greater controllability than would normally be expected in a real biological problem, it affords us the opportunity to get a good measure of the loss of controllability that results from *CoD-Reduce*, that is, from deletion, reduction and inducement.

Given this protocol for choosing the control–target pair, to study the performance of the CoD-Reduce algorithm, we have performed a simulation study on sets of 100 randomly generated Boolean networks with perturbation, with $7,8,9$ and 10 genes. These have been generated using the algorithm developed in Pal *et al.* (2005), subject to the constraint that for each network half of its attractors are among the desirable states. For each network of size $n\in\{7,8,9,10\}$, we have reduced the original network to $n-1,n-2,\ldots,4$ genes and for each reduction designed either the MFPT or SSD control policy on the reduced network and induced a policy on the original network of size n. We have limited networks to 10 genes because we need to compute the control policy on each originally generated network in order to make the comparisons. However, in general *CoD-Reduce* is not restricted to any number of genes and can handle large networks, as long as the SSD is obtainable. In such large networks *CoD-Reduce* is used to make reductions by deleting genes until the point that it is feasible to compute the control policy of the reduced network. For each generated network of size 7–10 genes, we apply the MFPT and SSD algorithms directly to it and compute the amount of steady state mass shifted from undesirable to desirable states. Letting $\pi_D=(\pi_{D1},\pi_{D2},\ldots,\pi_{Dm})$ and $\omega_D=(\omega_{D1},\omega_{D2},\ldots,\omega_{Dm})$ denote the probability vectors composed of steady state masses of the desirable states before and after control, respectively, the shift is defined by

$$\Delta=\sum_{k=1}^{m}\omega_{Dk}-\pi_{Dk} \qquad (9)$$

Δ provides a measure of the effectiveness of the overall algorithm—deletion, reduction and inducement—the goal being to decrease the probability of being in undesirable states and increase the probability of being in desirable states in the long run. We also apply the induced policies arising from reductions to $n-1,n-2,\ldots,4$ genes and compute the mass shifts. Figure 1 shows, for each n, the average shift of the SSD under the MFPT policy on the original network and the average shift for the induced policies arising from reductions to $n-1,n-2,\ldots,4$ genes. Figure 2 gives analogous results using the SSD policy. The salient point regarding the 9- and 10-gene networks is that, after an initial drop off for a few-gene reduction, the shift tends to stabilize for further reduction and, in all cases, the induced policy achieves significant beneficial results.

Note that the beneficial steady state shift when designing the control policy on the original network is, on average, slightly better for the SSD policy in comparison with the MFPT policy, and that this agrees with the findings in Qian *et al.* (2009). On the other

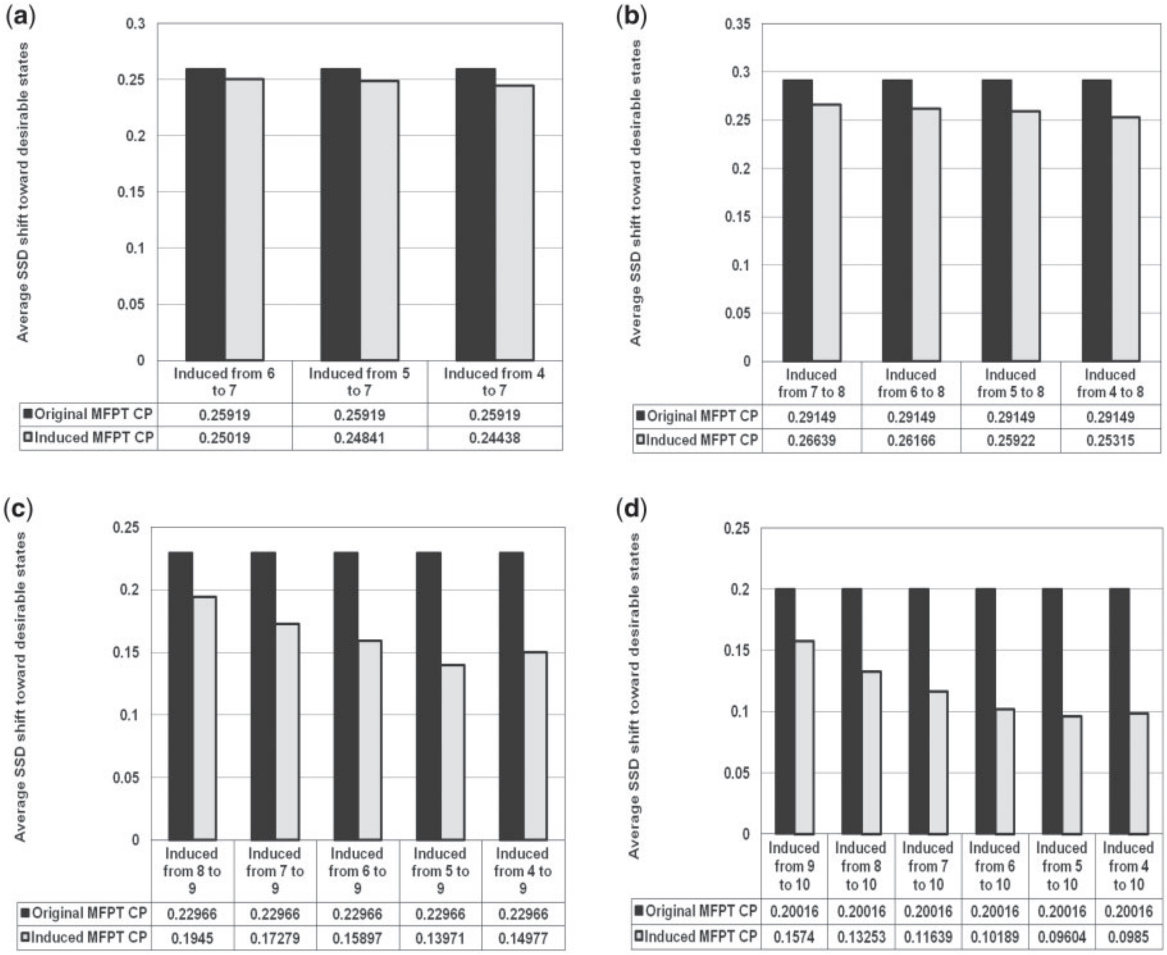
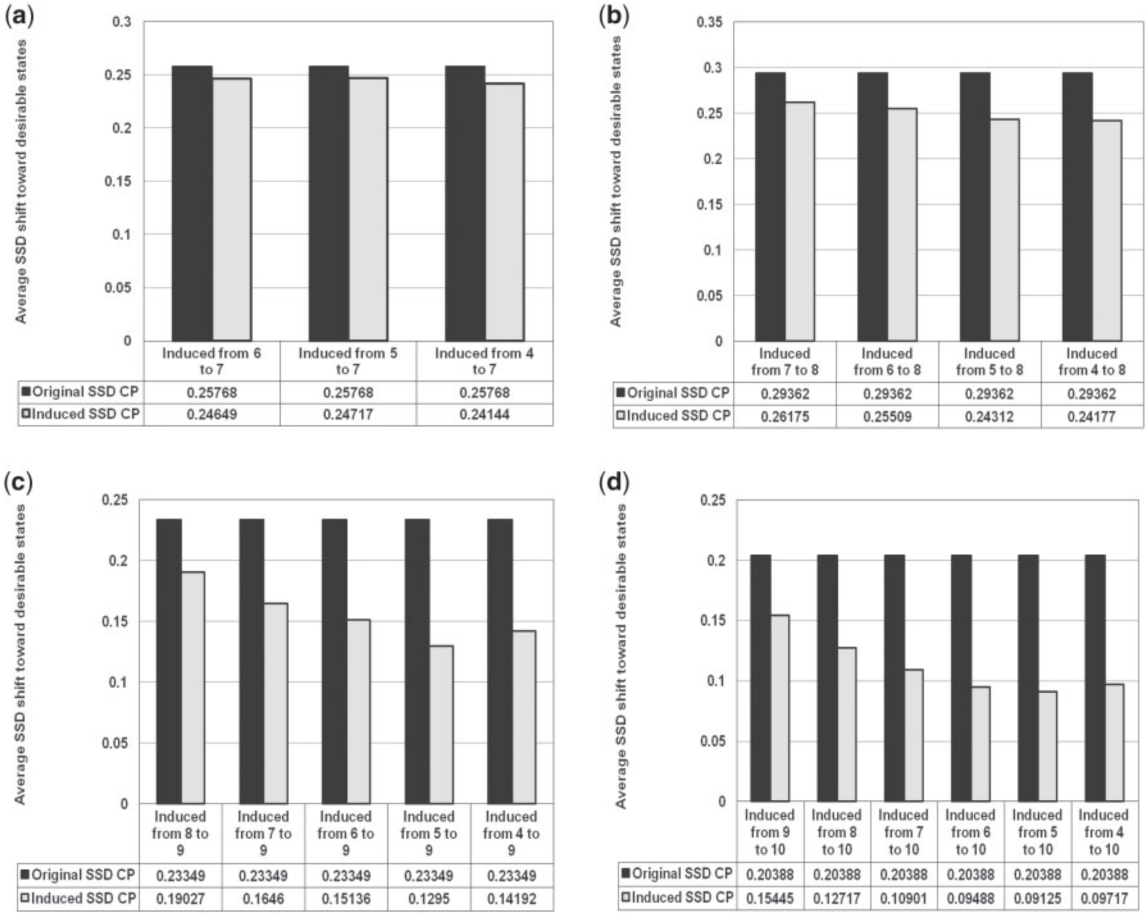


Fig. 1. The average shifts of the SSD produced by applying the original MFPT and the stationary-induced control policies, using different number of genes: (**a**) 7 genes, (**b**) 8 genes, (**c**) 9 genes and (**d**) 10 genes. The original MFPT control policies were obtained before any reductions. The induced control policies were designed on the reduced networks after applying reduction several times and then inducing the control policy of the reduced networks back to the original network. Each one of the four sets of 100 BN$_p$*s* was generated using randomly generated attractor sets; attractors are evenly distributed between desirable and undesirable states.

hand, the induced policy arising from the MFPT policy designed on the reduced network slightly outperforms the induced policy arising from the SSD policy designed on the reduced network.

4.2 Gastrointestinal cancer network application

To test *CoD-Reduce* on a larger, real-world network, we have applied it to a 17-gene network designed from a gastrointestinal cancer dataset (Price *et al.*, 2007). To infer the BN$_p$, the microarray data were normalized, filtered and binarized using methods from Shmulevich and Zhang (2002). Network inference is based on a network-growing algorithm that requires an initialization with a seed gene, a variant of the algorithm proposed in Hashimoto *et al.* (2004). As the seed, we use *OBSCN*, one of two genes composing the best classifier in Price *et al.* (2007). The network is grown by adding genes that have strong connectivity to this gene, as measured by the CoD. The seed gene, *OBSCN*, is set as the target gene and the second gene added to the network, *GREM2*, is set as the control gene. Unless there is a biologically known relation between a target gene and a particular phenotype, as in the case of *WNT5A* and metastatic competence in melanoma, there is no standard way to select a target and control gene pair; however, it is reasonable to expect that the best 1-gene classifier, *OBSCN*, that discriminates between two types of cancer can also be a potential target for a possible therapeutic intervention. *GERM2* has the strongest CoD connection to this gene and thus, could be viewed as a good candidate for a control gene. At each iterative step, the gene having the strongest connectivity, measured by the CoD, to one of the genes from the current network is added to the network. Then, the network is rewired taking into account that a new gene in the network can change the way genes influence each other. The 17-gene network includes the following genes: *OBSCN*, *GREM*2, *HSD*11*B*1, *UCHL*1, *A_24_P*920699, *BNC*1, *FMO*3, *LOC*441047, *THC*2123516, *NLN*, *COL*1*A*1, *IBSP*, *C*20*orf* 166, *KUB*3, *TPM*1, *D*90075 and *BC*042026. The CoD computations are based on the SSD of the current network. After generating the 17-gene network, *CoD-Reduce* is applied consecutively to reduce it to a 10-gene network. The final 10 genes that are remained in the network are: *OBSCN*, *GREM*2,



(a)

	Induced from 6 to 7	Induced from 5 to 7	Induced from 4 to 7
■Original SSD CP	0.25768	0.25768	0.25768
□Induced SSD CP	0.24649	0.24717	0.24144

(b)

	Induced from 7 to 8	Induced from 6 to 8	Induced from 5 to 8	Induced from 4 to 8
■Original SSD CP	0.29362	0.29362	0.29362	0.29362
□Induced SSD CP	0.26175	0.25509	0.24312	0.24177

(c)

	Induced from 8 to 9	Induced from 7 to 9	Induced from 6 to 9	Induced from 5 to 9	Induced from 4 to 9
■Original SSD CP	0.23349	0.23349	0.23349	0.23349	0.23349
□Induced SSD CP	0.19027	0.1646	0.15136	0.1295	0.14192

(d)

	Induced from 9 to 10	Induced from 8 to 10	Induced from 7 to 10	Induced from 6 to 10	Induced from 5 to 10	Induced from 4 to 10
■Original SSD CP	0.20388	0.20388	0.20388	0.20388	0.20388	0.20388
□Induced SSD CP	0.15445	0.12717	0.10901	0.09488	0.09125	0.09717

Fig. 2. The average shifts of the SSD produced by applying the original SSD and the stationary-induced control policies, using different number of genes: (**a**) 7 genes, (**b**) 8 genes, (**c**) 9 genes and (**d**) 10 genes. The original SSD control policies were obtained before any reductions. The induced control policies were designed on the reduced networks after applying reduction several times and then inducing the control policy of the reduced networks back to the original network. Each one of the four sets of 100 BN$_p$*s* was generated using randomly generated attractor sets; attractors are evenly distributed between desirable and undesirable states.

*HSD*11*B*1, *BNC*1, *LOC*441047, *NLN*, *C*20*orf*166, *KUB*3, *D*90075 and *BC*042026. At each reduction step, using an algorithm from Kim *et al.* (2002), the network SSD, which is required for the CoD computations, is estimated by first running the network for a long time and using the Kolmogorov–Smirnov test to decide if the network has reached its steady state. After reducing the original network down to 10 genes, the MFPT control policy for the reduced network is determined and induced back on the original 17-gene network. Figure 3 shows the considerable shift in the SSD of the network, about 0.15 of the probability mass after applying the induced control policy, toward desirable states.

4.3 Concluding remarks

The overall performance of CoD-Reduce depends on the individual performance of its components: gene deletion, the reduction mapping and inducement, and how these components interact with each other and with the control policy designed on the reduced network. While the CoD does not measure regulation directly, one can be fairly confident that if the CoD for gene *d* predicting the target gene is small, then *d* is unlikely to have much to do

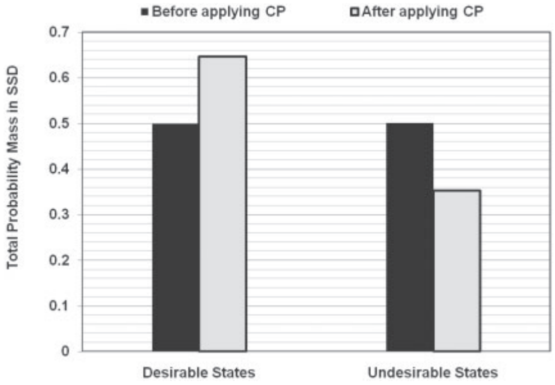


Fig. 3. SSD shift toward the desirable states in gastrointestinal cancer network after applying the induced control policy from the reduced network. There is ~15% shift in SSD toward the desirable states after reducing 17–10 genes and then inducing control policy designed on 10 gene network back to the original 17-gene network.

with controlling the target and therefore deletion of d from the network is unlikely to seriously impact a control policy for the target. It is impossible to say that our proposed selection policy is the best one available, but it is based on considerations closely tied to the long-run behavior of the network, whose control is our objective. Finally, the inducement procedure introduces some loss of optimality because, in effect, it identifies several states in the original network with a single state in the reduced network; nevertheless, it maintains the integrity of the control on the gene state vector for which it has been designed and this kind of criterion has a long history in system decompression in the absence of any further qualifying knowledge. Of course, the performance of *CoD-Reduce* depends on how these three components work in consort. The simulations for 10-gene and smaller networks indicate that *CoD-Reduce* provides significant beneficial steady state shift relative to the shift for the control policy designed on the original network. Perhaps more importantly, even though we cannot compare the shifts on the 17-gene gastrointestinal network produced by *CoD-Reduce* and the control policy designed on the original network (since we cannot obtain the latter), *CoD-Reduce* yields a substantial shift in the steady state mass, which is our pragmatic goal.

Conflict of Interest: none declared.

REFERENCES

Akutsu,T. *et al.* (2007) Control of Boolean networks: Hardness results and algorithms for the tree structured networks. *J. Theor. Biol.*, **244**, 670–677.

Aldana,M. *et al.* (2002) Boolean Dynamics with Random Couplings. In Kaplan,E. *et al.* (eds), *Perspectives and Problems in Nonlinear Science. A Celebratory Volume in Honor of Lawrence Sirovich*. Applied Mathematical Sciences Series, Springer.

Ball,K. *et al.* (2006) Asymptotic analysis of multiscale approximations to reaction networks. *Ann. Appl. Probab.*, **16**, 1925–1961.

Bertsekas,D. (2005) *Dynamic Programming and Optimal Control*. Athena Scientific.

Borisov,N. *et al.* (2005) Signaling through receptors and scaffolds: independent interactions reduce combinatorial complexity. *Biophys J.*, **89**, 951–966.

Brun,M. *et al.* (2005) Steady-state probabilities for attractors in probabilistic Boolean networks. *EURASIP J. Signal Process.*, **85**, 1993–2013.

Clarke,B.L. (1992) General method for simplifying chemical networks while preserving overall stoichiometry in reduced mechanisms. *J. Phys. Chem.*, **97**, 4066–4071.

Conzelmann,H. *et al.* (2001) Reduction of mathematical models of signal transduction networks: simulation-based approach applied to egf receptor signalling. *Syst. Biol.*, **1**, 159–169.

Conzelmann,H. *et al.* (2006) A domain-oriented approach to the reduction of combinatorial complexity in signal transduction networks. *BMC Bioinformatics*, **7**, 159–169.

Datta,A. *et al.* (2007) Control approaches for probabilistic gene regulatory networks. *IEEE Signal Process. Mag.*, **24**, 54–63.

Dougherty,E. *et al.* (2000) Coeffcient of determination in nonlinear signal processing. *Signal Processing*, **80**, 2219–2235.

Dougherty,E. *et al.* (2009) A conditioning-based model of contextual regulation. *IEEE/ACM Trans. Comput. Biol. Bioinform.*, **6**, 310–320.

Dougherty,E.R. *et al.* (2010) Stationary and structural control in gene regulatory networks: basic concepts. *Int. J. Syst. Sci.*, **41**, 5–16.

Faryabi,B. *et al.* (2007) On approximate stochastic control in genetic regulatory networks. *IET Syst. Biol.*, **1**, 361–368.

Gorban,A. *et al.* (2006) *Model Reduction and Coarse-graining Approaches for Multiscale Phenomena*. Springer, Berlin/Heidelberg.

Hartwell,L. *et al.* (1999) From molecular to modular cell biology. *Nature*, **402** (Suppl. 6761), C47–C52.

Hashimoto,R. *et al.* (2004) A directed-graph algorithm to grow genetic regulatory subnetworks from seed genes based on strength of connection. *Bioinformatics*, **20**, 1241–1247.

Huang,S. (1999) Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery. *J. Mol. Med.*, **77**, 469–480.

Huang,S. (2007) *Cell State Dynamics and Tumorigenesis in Boolean Regulatory Networks*. Springer, Berlin, Heidelberg.

Hunter,J. (2005) Stationary distributions and mean first passage times of perturbed Markov chains. *Linear Algebra Appl.*, **410**, 217–243.

Indic,P. *et al.* (2006) Development of a two-dimension manifold to represent high dimension mathematical models of the intracellular mammalian circadian clock. *Biol. Rhythms*, **21**, 222—-232.

Ivanov,I. and Dougherty,E. (2004) Reduction mappings between probabilistic Boolean networks. *EURASIP JASP*, **1**, 125–131.

Ivanov,I. *et al.* (2007) Dynamics preserving size reduction mappings for probabilistic Boolean networks. *IEEE Trans. Signal Process.*, **55**, 2310–2322.

Kauffman,S. (1993) *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press, New York.

Kauffman,S.A. (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, **22**, 437–467.

Kim,S. *et al.* (2000) A general framework for the analysis of multivariate gene interaction via expression arrays. *Biomed. Optic*, **5**, 411–424.

Kim,S. *et al.* (2002) Can Markov chain models mimic biological regulation. *Biol. Syst.*, **10**, 447–458.

Martins,D. *et al.* (2008) A general framework for the analysis of multivariate gene interaction via expression arrays. *IEEE J. Sel.Top. Signal Process.*, **2**, 424–439.

Naldi,A. *et al.* (2009) A reduction method for logical regulatory graphs preserving essential dynamical properties. *Lect. Notes Bioinform.*, **5688**, 266–280.

Ng,M. *et al.* (2006) A control model for markovian genetic regulatory networks. *Trans. Comput. Syst. Biol. V*, **4070**, 36–48.

Norris,J. (1998) *Markov Chains*. Cambridge University Press.

Pal,R. *et al.* (2005) Generating boolean networks with a prescribed attractor structure. *Bioinformatics*, **54**, 4021–4025.

Pal,R. *et al.* (2006) Optimal infinite-horizon control for probabilistic Boolean networks. *IEEE Trans. Signal Process.*, **54**, 2375–2387.

Price,N.D. *et al.* (2007) Highly accurate two-gene classifier for differentiating gastrointestinal stromal tumors and leiomyosarcomas. *Proc. Natl Acad. Sci. USA*, **104**, 3414–3419.

Qian,X. and Dougherty,E.R. (2008) Effect of function perturbation on the steady-state distribution of genetic regulatory networks: Optimal structural intervention. *IEEE Trans. Signal Process.*, **56**, 4966–4975.

Qian,X. and Dougherty,E.R. (2009) On the long-run sensitivity of probabilistic Boolean networks. *J. Theor. Biol.*, **257**, 560–577.

Qian,X. *et al.* (2009) Intervention in gene regulatory networks via greedy control policies based on long-run behavior. *BMC Syst. Biol.*, **3**, 1–16.

Radulescu,O. *et al.* (2008) Robust simplifications of multiscale biochemical networks. *BMC Syst. Biol.*, **2**, 1–25.

Ramo,P. *et al.* (2006) Perturbation avalanches and criticality in gene regulatory networks. *J. Theor. Biol.*, **242**, 164–170.

Saez-Rodriguez,J. *et al.* (2005) Dissecting the puzzle of life: modularization of signal transduction networks. *Comput. Chem. Eng.*, **29**, 619–629.

Schweitzer,P. (1968) Perturbation theory and finite Markov chains. *J. Appl. Probab.*, **5**, 401–413.

Serra,R. *et al.* (2004) Genetic network models and statistical properties of gene expression data in knock-out experiments. *J. Theor. Biol.*, **227**, 149–157.

Shmulevich,I. and Dougherty,E.R. (2007) *Genomic Signal Processing*. Princeton University Press, Princeton.

Shmulevich,I. and Zhang,W. (2002) Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics*, **18**, 555–565.

Shmulevich,I. *et al.* (2002) Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, **18**, 261–274.

Vahedi,G. *et al.* (2008) Intervention in gene regulatory networks via a stationary mean-first-passage-time control policy. *IEEE Trans. Biomed. Eng.*, **55**, 2319–2331.

Wang,R. *et al.* (2004) Modelling periodic oscillation of biological systems with multiple timescale networks. *IET Syst. Biol. J.*, **1**, 71–84.

Yu,L. *et al.* (2009) Inference of transition probabilities between the attractors in boolean networks with perturbation. *In IEEE Workshop on Genomic Signal Processing and Statistics*. Minneapolis, MN.