

# The impact of function perturbations in Boolean networks

Yufei Xiao<sup>1</sup> and Edward R. Dougherty<sup>1,2,\*</sup><sup>1</sup>Department of Electrical & Computer Engineering, Texas A&M University, College Station, TX 77843, USA,<sup>2</sup>Computational Biology Division, Translational Genomics Research Institute, Phoenix, AZ 85004, USA

Received on November 21, 2006; revised on February 6, 2007; accepted on March 5, 2007

Advance Access publication March 22, 2007

Associate Editor: Trey Ideker

## ABSTRACT

**Motivation:** A network is said to be *robust* relative to a certain network characteristic if a small change in network structure does not significantly affect the characteristic. From the perspective of network stability, robustness is desirable; however, from the perspective of intervention to exert influence on network behavior, it is undesirable. For Boolean networks, there are two fundamental types of robustness. One type pertains to perturbing the state of the network and the other to perturbing the rule-based structure.

**Results:** This article explores the impact of function perturbations in Boolean networks from two aspects: (1) analysis: predict the impact on network state transitions and attractors via analytical approaches or identify a perturbation by observing its consequences; (2) synthesis: preserve or modify the network characteristics, especially attractors, by introducing a judicious change to the functions. The results are applied to achieve intervention that structurally alters the network to achieve a more favorable steady-state distribution and to identify the function perturbation that has led to altered observed behavior. The intervention procedure is applied to a WNT5A network to reduce the risk of metastasis in melanoma, and the identification procedure is applied to a *Drosophila melanogaster* segmentation polarity gene network to identify regulatory function perturbation.

**Contact:** edward@ece.tamu.edu

## 1 INTRODUCTION

A network is said to be *robust* relative to a certain network characteristic if a small change in network structure does not significantly affect the characteristic. In the case of a Boolean network (BN), which is a rule-based binary network, a key form of robustness is with respect to how a small change in a regulatory rule, say the flip of one value in its truth table, affects the steady state of the network. Robustness is a double-edged sword. For instance, BNs are used to model gene regulation, with gene expressions being quantized as 0s and 1s to represent *not expressed* and *expressed states*, respectively, and gene regulation is described by Boolean logic. Because network inference is inherently ill-posed on account of measurement error and the impact of latent variables,

which are either immeasurable or simply not included in the model (whose influence nevertheless still exists), model robustness is desirable for inference so that slightly differently inferred networks will exhibit similar fundamental characteristics. On the other hand, if the goal is to intervene in the network, for instance, to modify its long-run behavior so as to drive it away from undesirable states, say, metastasis in cancer, then robustness is undesirable because it impedes intervention. This article addresses the robustness of BNs relative to small perturbations of the regulatory rules.

The objective of this study is comprised of two aspects:

- Provide a theory to analytically predict the consequences of function perturbations in a BN, including the impacts on state transitions and steady-state properties (mainly attractors).
- Apply the analytical theory to intervene or control BNs to obtain desirable properties.

Given that BNs are often used to model genetic regulation, the preceding two aspects will help gain insight into gene regulatory network modeling in the following ways:

- By helping to analyze the influences of modeling uncertainty and latent variables, since these two common phenomena can often be formulated as changed Boolean functions, thereby falling into the function perturbation category.
- By aiding in the design of intervention methods to control gene regulation for the purpose of altering steady-state cell behavior.
- By providing the means to identify the regulatory perturbations underlying observed changes in cell behavior.

This study is motivated by the fact that these issues have not been treated thoroughly in the literature: (1) previous works usually concern state perturbation, which is temporary in nature, instead of function perturbation; (2) many works study ensembles of BNs by exploring their overall (statistical) behavior, but do not consider the effect on a specific BN; (3) many works do not mention attractors, and these characterize the long-run properties of BNs, which in the context of gene regulatory networks represent phenotypic properties. A review of the existing literature on BN robustness is provided below.

\*To whom correspondence should be addressed.

There exist two kinds of variations with respect to a BN: perturbation of the states and perturbation of the functions (the regulatory rules). The first kind of perturbation is temporary and entails a reset of the network state to a new state. Such a perturbation does not alter the structure of the BN and has no influence on its steady-state properties; however, it does affect the network dynamics by replacing the original time trajectory with a new trajectory. Since a BN possesses one or more attractors, and each attractor has its basin of attraction (BOA) (consisting of the states that will eventually transit to this attractor), after a perturbation of the state, the new trajectory may converge to the original trajectory and reach the same attractor, or it may go to another BOA and reach a different attractor. If it is preferable to reach the original attractor after the perturbation, then it is desirable that the BN be *stable* or *dynamically robust*, meaning that it has a tendency to resist disturbance of the state and converge to its original trajectory.

The second kind of perturbation, namely perturbation of the functions, has a more fundamental impact on the BN and has been less studied. The network steady-state distribution may undergo a permanent transformation: the basins of attraction for some attractors will enlarge, shrink or shift; some attractors will disappear; and new attractors may be created. Therefore, starting from the same initial state, the new trajectory may or may not reach the original attractor. Understanding the impact of function perturbation on steady-state properties is important for application. As noted, depending on the context, robustness may or may not be desirable: the robustness of an attractor and its BOA is desirable if we would like to preserve them.

State perturbations affect the network dynamics, not the steady-state properties. This issue has been well studied, often via the ensemble behavior of a large number of random synchronous BNs (all the functions in the BN are updated simultaneously at each time step). One study shows that a natural class of robust networks is composed of scale-free networks, in which a small (but significant) fraction of the elements are highly connected and the majority of the elements are poorly connected (Aldana *et al.*, 2003). Another demonstrates the robustness of BNs whose functions belong to certain Post classes (Shmulevich *et al.*, 2003). The conclusions by Aldana *et al.* (2003) and Shmulevich *et al.* (2003) are based on the ensemble behavior of random BNs. A different approach explores the robustness of annealed (the connections and functions will change at each time step) BNs through the bias map, which is a mapping  $b_t \rightarrow b_{t+1}$ ,  $b_t$  being the probability of a gene being 1 at time step  $t$  (Ramo *et al.*, 2005). It introduces the concept of a stabilizing Boolean function and relates it to network stability (dynamic robustness). It shows that many Post and canalizing functions are stabilizing functions, which agrees with Aldana *et al.* (2003) and Shmulevich *et al.* (2003). Another perspective is to define robustness as the expected probability of a single flipped input altering the output of a Boolean function over a distribution for  $K$ -input functions and averaged over all the nodes of the network (Kauffman *et al.*, 2004). Flipping the function input is a form of state perturbation, and the robustness measure concerns the ensemble behavior of

random BNs. The conclusion is that Boolean networks with canalizing functions are stable (the robustness measure is less than 1). Robustness to noise is treated in Qu *et al.* (2002), where the function output has a probability of flipping its value, and the first crossing time is defined as the time needed for two time trajectories with different initial states to cross. The article concludes that, for two categories of random BNs (in the chaotic and ordered phases), whether the initial states belong to the same BOA or not, the average first crossing time over a large number of networks behaves robustly.

In the above-cited studies, only state perturbations are considered. As a state perturbation affects the network in a temporary manner, meaning that only the network dynamics are affected, we naturally would like to pursue a further issue, namely, how a perturbation of the functions in the network will influence the attractors and the long-term behaviors of the network. The effect of function perturbation is less studied in the literature. One of such studies is based on the ensemble performance of a large number of random BNs: In Gerhenson *et al.* (2006), network stability is measured by the overlap of state space transitions of the original BN and the one-bit mutant BN resulting from a perturbation of a Boolean function through a one-bit change to its truth table. The authors discover that adding a redundant node can boost the robustness of one-bit mutant BNs. Here, although function perturbation is studied, emphasis on the network attractors is lacking. Another work, contributed by Chaves *et al.* (2005), studies a different kind of perturbation: updating the functions in an asynchronous setting. The effect of asynchronous updates of the functions on the dynamics of Boolean-type models for the *Drosophila melanogaster* segment polarity genes is considered and different asynchronous update schemes are tested. One model is found to be robust to changes in the initial state and robust to the update variability; certain restrictions (a minimal prepattern) on the update scheme must be satisfied to ensure convergence to the desired wild-type steady state. Owing to the nature of asynchronous networks, the attractor robustness is not treated in the sense that attractors exactly constitute the steady state in a synchronous BN.

In this article, we study the effect of function perturbation on the attractors in a homogenous synchronous BN, meaning that the Boolean functions are updated simultaneously and the functions do not change over time. Our analysis is applicable to any individual BN instead of being targeted at the ensemble performance of a type of BNs such as in Aldana *et al.* (2003) and Kauffman *et al.* (2004), or a particular BN such as in Chaves *et al.* (2005). Therefore, the methods and results in our article are more general. We do not define a robust measure; rather, we explore the exact consequences of function perturbations and show how they can be utilized for analysis and synthesis. We focus on function perturbation in the form of a one-bit change of the truth table and explore its impact on the attractors. We address both the robustness and flexibility issues, and show that the latter can be useful for intervention in BNs. Since state transitions completely characterize the network dynamics and define a BN's attractors and basins of attraction, we propose to pursue our objective through the following issues: (1) impact on state

transitions; (2) impact on attractors, namely, which attractors will be invariant to the perturbations and which non-attractor states will become new attractors and (3) applications, including intervention (given a BN, design an intervention strategy to achieve a certain objective through function perturbation) and perturbation identification (given the observed state transitions of a BN and the state transitions after function perturbation, identify the perturbation).

## 2 SYSTEMS AND METHODS

### 2.1 Boolean networks

A BN is composed of  $n$  nodes  $x_1, x_2, \dots, x_n$ , each of which may take a value from  $\{0, 1\}$ . A node  $x_i$  has  $k_i$  ( $k_i \leq n$ ) parent nodes  $x_{i1}, x_{i2}, \dots, x_{ik_i}$ , with  $\{i1, \dots, ik_i\} \subseteq \{1, 2, \dots, n\}$ . The value of node  $x_i$  at time  $t+1$  is determined by its parent nodes at time  $t$  through a Boolean function

$$x_i(t+1) = f_i(x_{i1}(t), x_{i2}(t), \dots, x_{ik_i}(t)), \quad 1 \leq i \leq n.$$

$k_i$  is called the *connectivity* of node  $x_i$ . Assuming the Boolean functions are homogeneous (time invariant) and the nodes are updated synchronously, we can write the Boolean functions in the simplified form

$$x_i = f_i(x_{i1}, x_{i2}, \dots, x_{ik_i}), \quad 1 \leq i \leq n.$$

We use  $f_i(x_1, x_2, \dots, x_n)$  and  $f_i(x_{i1}, x_{i2}, \dots, x_{ik_i})$  interchangeably, with the latter specifying only the essential variables.

At time  $t$ , the *state* of the BN refers to the vector  $\mathbf{x}(t) = (x_1(t), x_2(t), \dots, x_n(t))$ . A specific state can be written as an  $n$ -dimensional binary vector  $(a_1, a_2, \dots, a_n)$ , where  $a_i \in \{0, 1\}$ ,  $1 \leq i \leq n$ , or written as  $a_1, a_2, \dots, a_n$  in a compact form. The state space is  $\{0, 1\}^n = \{00 \dots 0, 00 \dots 1, \dots, 11 \dots 1\}$ , whose size equals  $2^n$ . Given a current state  $\mathbf{x}(t)$ , one can obtain the state at the next time step  $\mathbf{x}(t+1)$  by evaluating the  $n$  Boolean functions. If one allows  $\mathbf{x}(t)$  to be any one of the  $2^n$  possible states, from  $00 \dots 0$  to  $11 \dots 1$ , and computes their respective next states, then a list of  $2^n$  one-step state transitions can be obtained. The  $2^n$  state transitions fully characterize a BN's dynamics. For instance, if a BN has three nodes, its one-step state transitions will consist of eight states, which are the successors of states  $000, 001, \dots, 110, 111$ , respectively.

Given the one-step state transitions, a directed graph  $T(V, E)$ , known as the *state transition diagram*, can thus be constructed for the BN.  $V$  is the set of  $2^n$  vertices, each vertex being a state of a BN.  $E$  is the set of  $2^n$  edges, each pointing from a state to its next state in pair-wise state transitions. If a state's next state is itself, then the edge is a loop.

The long-run behavior of a BN is characterized by its attractors. Starting from an initial state, it will eventually reach a set of states, called an *attractor cycle* (a closed path in the state transition diagram), that it will thereafter cycle through endlessly. A BN may have more than one attractor cycle. If an attractor cycle consists of one state only, then we call it a *singleton attractor*. The set of all states that eventually evolve into the same attractor cycle constitutes its BOA. Different basins of attraction are depicted in the state transition diagram as disjoint subgraphs. The attractor cycles in BNs modeling biological networks are typically associated with phenotypes and tend to be short (Kauffman *et al.*, 1993) with biological stability contributing to singleton attractors. For instance, singleton attractors have been associated with phenotypes such as cell proliferation and apoptosis (Huang *et al.*, 1999). Our basic results will be stated for singleton attractors, for which the analysis leads to tractable propositions, and we will then show how they can be extended to multiple-state attractor cycles, albeit, with increased complexity.

A Boolean function can be represented by a truth table shown below. If the function  $f_i$  depends on  $k_i$  input variables  $x_{i1}, x_{i2}, \dots, x_{ik_i}$ ,

then the evaluated input vector on row  $j$  ( $1 \leq j \leq 2^{k_i}$ ) of the truth table is denoted by  $\mathbf{a}_j^i$ , with  $\mathbf{a}_j^i \in \{0, 1\}^{k_i}$ . For instance, in the truth table below,  $\mathbf{a}_2^i = 00 \dots 1$ .

Row label	$x_{i1}x_{i2} \dots x_{ik_i}$	$f_i(\cdot)$
1	00 $\dots$ 0	$f_i(00 \dots 0)$
2	00 $\dots$ 1	$f_i(00 \dots 1)$
$\vdots$	$\vdots$	$\vdots$
$2^{k_i}$	11 $\dots$ 1	$f_i(11 \dots 1)$

The state transition  $\mathbf{s} \rightarrow \mathbf{w}$  depends on the vector function  $\mathbf{f} = (f_1, f_2, \dots, f_n)$ . Restricting our attention to  $f_i$  means considering the  $i$ th mapping  $\mathbf{s} \rightarrow w_i$ . Since  $f_i$  depends on  $(x_{i1}, x_{i2}, \dots, x_{ik_i})$ , the mapping depends on  $(s_{i1}, s_{i2}, \dots, s_{ik_i})$  only. If  $\mathbf{u} \neq \mathbf{s}$  but  $(u_{i1}, u_{i2}, \dots, u_{ik_i}) = (s_{i1}, s_{i2}, \dots, s_{ik_i})$ , then  $\mathbf{u}$  and  $\mathbf{s}$  both map to  $w_i$  under  $f_i$ . If we let  $\text{In}_i(\mathbf{s}) = (s_{i1}, s_{i2}, \dots, s_{ik_i})$  denote the input vector for function  $f_i$  as it operates on state  $\mathbf{s}$ , then  $\text{In}_i(\mathbf{u}) = \text{In}_i(\mathbf{s})$  implies  $f_i(\mathbf{u}) = f_i(\mathbf{s})$ . For instance, if  $n=5$ ,  $\mathbf{s} = 01001$ ,  $\mathbf{u} = 00011$  and  $x_{i1}x_{i2} \dots x_{ik_i} = x_1x_3x_5$ , then  $\text{In}_i(\mathbf{u}) = \text{In}_i(\mathbf{s}) = 001$  and  $f_i(\mathbf{u}) = f_i(\mathbf{s}) = f_i(001)$ .

In a BN, a one-bit perturbation occurs when one chooses a function  $f_i$  and makes a one-bit change of its truth table by flipping the value on the  $j$ th entry ( $1 \leq j \leq 2^{k_i}$ ), that is, change 0 to 1 or change 1 to 0. We denote the new function by  $f_i^{(j)}$ , so that the one-bit perturbation on row  $j$  takes the form  $f_i \rightarrow f_i^{(j)}$ , where  $f_i^{(j)}(\mathbf{a}_j^i) = 1 - f_i(\mathbf{a}_j^i)$ . Since single-node flips play a key role in our analysis, we introduce the following notation: if  $\mathbf{s} = (s_1, s_2, \dots, s_n)$ , then  $\mathbf{s}^{(j)} = (s_1, \dots, s_{j-1}, 1 - s_j, s_{j+1}, \dots, s_n)$ .

### 2.2 Theoretical results

We begin with a proposition and corollaries describing the basic effects of a single one-bit perturbation on the state transitions of a BN.

**PROPOSITION 1.** *The state transition  $\mathbf{s} \rightarrow \mathbf{w}$  is affected by the one-bit perturbation  $f_i \rightarrow f_i^{(j)}$  if and only if  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$ . If the state transition is affected, then the new state transition will be  $\mathbf{s} \rightarrow \mathbf{w}^{(j)}$ .*

**PROOF.** The first statement follows at once from the fact the  $i$ th mapping transition  $\mathbf{s} \rightarrow w_i$  depends only on  $\text{In}_i(\mathbf{s})$ , and this is affected by the perturbation if and only if  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$ . Next, suppose the transition is affected by the perturbation. Then, absent perturbation, the  $i$ th mapping is given by  $\mathbf{s} \rightarrow f_i(\text{In}_i(\mathbf{s})) = f_i(\mathbf{a}_j^i) = w_i$ ; with perturbation, the  $i$ th mapping is  $\mathbf{s} \rightarrow f_i^{(j)}(\text{In}_i(\mathbf{s})) = f_i^{(j)}(\mathbf{a}_j^i) = 1 - f_i(\mathbf{a}_j^i) = 1 - w_i$ . Since the other  $n-1$  mappings remain unchanged, following the perturbation, state  $\mathbf{s}$  will transit to  $\mathbf{w}^{(j)}$ .

**COROLLARY 1.** *If  $|\mathbf{a}_j^i| = k_i$ , then the one-bit perturbation  $f_i \rightarrow f_i^{(j)}$  will result in  $2^{n-k_i}$  changed state transitions in the state transition diagram. This is equivalent to  $2^{n-k_i}$  altered edges in the state transition diagram.*

**PROOF.** According to the preceding proposition, the transition of a state  $\mathbf{s}$  is affected by the perturbation  $f_i \rightarrow f_i^{(j)}$  if and only if  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$ . Among the  $2^n$  states  $\mathbf{s}$ ,  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$  for exactly  $2^{n-k_i}$  of them.

**COROLLARY 2. (Invariant singleton attractor)** *Suppose state  $\mathbf{s}$  is a singleton attractor. It will no longer be a singleton attractor following the one-bit perturbation  $f_i \rightarrow f_i^{(j)}$  if and only if  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$ .*

**PROOF.** According to the proposition, subsequent to the perturbation,  $\mathbf{s} \rightarrow \mathbf{s}^{(j)}$  if  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$ , in which case it is no longer a singleton attractor, and  $\mathbf{s} \rightarrow \mathbf{s}$  if  $\text{In}_i(\mathbf{s}) \neq \mathbf{a}_j^i$ , in which case it remains a singleton attractor.

**COROLLARY 3. (Emerging singleton attractor)** *A non-singleton-attractor state  $\mathbf{s}$  becomes a singleton attractor as a result of the one-bit*

perturbation  $f_i \rightarrow f_i^{(j)}$ , if and only if the following are true: (1)  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$ , and (2) absent the perturbation,  $\mathbf{s} \rightarrow \mathbf{s}^{(j)}$ .

A natural question arises when a desirable singleton attractor  $\mathbf{s}$  is lost on account of a one-bit perturbation: can a second perturbation restore it? If the perturbation  $f_i \rightarrow f_i^{(j)}$  causes  $\mathbf{s}$  to no longer be a singleton attractor, then the new transition of  $\mathbf{s}$  must be  $\mathbf{s} \rightarrow \mathbf{s}^{(j)}$ . According to the previous corollary,  $\mathbf{s}$  will again become a singleton attractor as a result of the one-bit perturbation  $f_k \rightarrow f_k^{(l)}$  if and only if  $\text{In}_k(\mathbf{s}) = \mathbf{a}_l^k$ , and, absent the perturbation,  $\mathbf{s} \rightarrow \mathbf{s}^{(k)}$ . From the second condition, since we know that  $\mathbf{s} \rightarrow \mathbf{s}^{(j)}$ , we must have  $k=i$ . Hence, from the first condition we must have  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$ , but from the fact that  $\mathbf{s}$  has been affected by the perturbation  $f_i \rightarrow f_i^{(j)}$ , we know that  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$ . Hence,  $\mathbf{s}$  is restored to being a singleton attractor by the same one-bit perturbation  $f_i \rightarrow f_i^{(j)}$  that caused it to cease being a singleton attractor, and no other. This means that the original BN is restored.

According to Corollary 2, a singleton attractor  $\mathbf{s}$  is no longer a singleton attractor following a one-bit perturbation  $f_i \rightarrow f_i^{(j)}$  if  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$ , but could it remain an attractor state as part of an attractor cycle following perturbation? Indeed it could. Consider the following situation for three nodes: 000 is a singleton attractor,  $001 \rightarrow 010$  and  $010 \rightarrow 000$ , so that 001 and 010 are in the basin of 000. The three functions are defined by  $f_1(x_2, x_3) = 0$  for all  $x_2, x_3$ ;  $f_2(x_1, x_3) = 0$  for all  $x_1, x_3$  except for  $f_2(0, 1) = 1$ ; and  $f_3(x_2, x_3) = 0$  for all  $x_2, x_3$ . Consider the one-bit perturbation  $f_3 \rightarrow f_3^{(0)}$ . Following the perturbation, the third function becomes  $f_3^{(0)}(x_2, x_3) = 0$  for all  $x_2, x_3$ , except for  $f_3^{(0)}(0, 0) = 1$ . This leads to the following transitions,  $000 \rightarrow 001$ ,  $001 \rightarrow 010$  and  $010 \rightarrow 000$ , and hence the attractor cycle  $000 \rightarrow 001 \rightarrow 010 \rightarrow 000$ . Thus, our care in stating the results is not unwarranted.

From the preceding example, we see that a one-bit perturbation can result in a singleton attractor becoming a member in a multiple-state attractor cycle. On the other hand, it should be clear from Proposition 1 that a one-bit perturbation can affect a multiple-state attractor cycle. Suppose  $\mathbf{s}_1 \rightarrow \mathbf{s}_2 \rightarrow \dots \rightarrow \mathbf{s}_m \rightarrow \mathbf{s}_1$  is an  $m$ -state attractor cycle. It follows from Proposition 1 that this cycle will be affected by the one-bit perturbation  $f_i \rightarrow f_i^{(j)}$ , if and only if  $\mathbf{a}_j^i \in \{\text{In}_i(\mathbf{s}_1), \text{In}_i(\mathbf{s}_2), \dots, \text{In}_i(\mathbf{s}_m)\}$ . By being affected, we mean that the exact cycle is not an attractor cycle following the perturbation. For instance, suppose  $\mathbf{a}_j^i = \text{In}_i(\mathbf{s}_1)$ . Then, following the perturbation,  $\mathbf{s}_1 \rightarrow \mathbf{s}_2^{(j)} \neq \mathbf{s}_2$ . Of course,  $\mathbf{s}_1$  might still be an attractor state as part of some other attractor cycle.

Corollary 1 puts an upper bound on the number of singleton attractors that can be lost owing to a one-bit perturbation  $f_i \rightarrow f_i^{(j)}$ , namely,  $\min\{2^{n-k_i}, N\}$ , where  $|\mathbf{a}_j^i| = k_i$  and  $N$  is the number of singleton attractors. Taking a network view, the total number of singleton attractors lost is bounded by

$$\min\{2^n \sum_{i=1}^n 2^{-k_i}, N\} \leq \min\{n2^{n-k_{\min}}, N\}$$

where,  $k_{\min}$  is the minimum connectivity among the nodes. Increased connectivity provides greater robustness relative to the loss of singleton attractors via one-bit perturbations.

Thus far, except for considering a second perturbation to restore a singleton attractor lost on account of a one-bit perturbation, we have focused on single one-bit perturbations. Increasing the number of one-bit perturbations increases the complexity of the problem; indeed, any BN on the same variables can be obtained from any other via a sufficiently long sequence of one-bit perturbations. If we consider two one-bit perturbations, then there are two cases: (1) the same function is changed and a flip occurs on two rows and (2) two functions are changed. The two cases can be expressed as: (1)  $f_i \rightarrow f_i^{(j)}$  and  $f_i^{(j)} \rightarrow f_i^{(l)}$ ,  $j \neq l$ ; (2)  $f_i \rightarrow f_i^{(j)}$  and  $f_k \rightarrow f_k^{(l)}$ ,  $i \neq k$ . To extend Proposition 1 to two one-bit perturbations, we let  $\mathbf{w}^{(i,k)}$  denote the state obtained from  $\mathbf{w}$  by flipping the  $i$ th and  $k$ th nodes. We state

separate extensions for the two cases: (1) the state transition  $\mathbf{s} \rightarrow \mathbf{w}$  is affected by the two-bit perturbation  $f_i \rightarrow f_i^{(j,l)}$ ,  $j \neq l$ , if and only if  $\text{In}_i(\mathbf{s}) \in \{\mathbf{a}_j^i, \mathbf{a}_l^i\}$ , and if it is affected, then  $\mathbf{s} \rightarrow \mathbf{w}^{(j,l)}$ . (2) the state transition  $\mathbf{s} \rightarrow \mathbf{w}$  is affected by the two one-bit perturbations  $f_i \rightarrow f_i^{(j)}$  and  $f_k \rightarrow f_k^{(l)}$  ( $i \neq k$ ), if and only if  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$  or  $\text{In}_k(\mathbf{s}) = \mathbf{a}_l^k$ , and if it is affected, then  $\mathbf{s} \rightarrow \mathbf{w}^{(j,l)}$  when  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$  and  $\text{In}_k(\mathbf{s}) \neq \mathbf{a}_l^k$ ,  $\mathbf{s} \rightarrow \mathbf{w}^{(k,l)}$  when  $\text{In}_k(\mathbf{s}) = \mathbf{a}_l^k$  and  $\text{In}_i(\mathbf{s}) \neq \mathbf{a}_j^i$ , and  $\mathbf{s} \rightarrow \mathbf{w}^{(i,k)}$  when  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$  and  $\text{In}_k(\mathbf{s}) = \mathbf{a}_l^k$ .

The corollaries of Proposition 1 are extended to the two one-bit perturbations as follows.

**COROLLARY 4.** *Supposing two one-bit perturbations take place in the BN, consider the following two cases.*

*Case (a):  $f_i \rightarrow f_i^{(j,l)}$  (i.e.  $f_i$  is perturbed on rows  $j$  and  $l$ ,  $j \neq l$ ) will result in  $2^{n-k_i+1}$  changed state transitions*

*Case (b):  $f_i \rightarrow f_i^{(l_1)}$  and  $f_j \rightarrow f_j^{(l_2)}$  ( $i \neq j$ ), assuming  $|\mathbf{a}_{l_1}^i| = k_i$ ,  $|\mathbf{a}_{l_2}^j| = k_j$ , and  $f_i$  and  $f_j$  have  $k_{ij}$  input variables in common,*

*(b1) if each of the  $k_{ij}$  variables takes the same value in  $\mathbf{a}_{l_1}^i$  and in  $\mathbf{a}_{l_2}^j$ , then there will be  $2^{n-k_i} + 2^{n-k_j} - 2^{n-k_i-k_j+k_{ij}}$  changed state transitions;*

*(b2) otherwise, there will be  $2^{n-k_i} + 2^{n-k_j}$  changed state transitions.*

**COROLLARY 5. (Invariant singleton attractor)** *Suppose state  $\mathbf{s}$  is a singleton attractor. It will no longer be a singleton attractor following the two-bit perturbation  $f_i \rightarrow f_i^{(j,l)}$ , if and only if  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$  or  $\mathbf{a}_l^i$ .  $\mathbf{s}$  will cease to be a singleton attractor following the two one-bit perturbations  $f_i \rightarrow f_i^{(l_1)}$  and  $f_j \rightarrow f_j^{(l_2)}$ , if and only if  $\text{In}_i(\mathbf{s}) = \mathbf{a}_{l_1}^i$  or  $\text{In}_j(\mathbf{s}) = \mathbf{a}_{l_2}^j$ .*

**COROLLARY 6. (Emerging singleton attractor)** *A non-singleton-attractor state  $\mathbf{s}$  becomes a singleton attractor as a result of the two-bit perturbation  $f_i \rightarrow f_i^{(j,l)}$ , if and only if the following is true:  $\text{In}_i(\mathbf{s}) = \mathbf{a}_j^i$  or  $\mathbf{a}_l^i$ , and absent the perturbation,  $\mathbf{s} \rightarrow \mathbf{s}^{(j,l)}$ .*

*$\mathbf{s}$  will become a singleton attractor following the two one-bit perturbations  $f_i \rightarrow f_i^{(l_1)}$  and  $f_j \rightarrow f_j^{(l_2)}$ , if and only if one of the following is true:*

- (1)  $\text{In}_i(\mathbf{s}) = \mathbf{a}_{l_1}^i$  and  $\text{In}_j(\mathbf{s}) = \mathbf{a}_{l_2}^j$ , and absent the perturbation,  $\mathbf{s} \rightarrow \mathbf{s}^{(i,j)}$ ;*
- (2)  $\text{In}_i(\mathbf{s}) = \mathbf{a}_{l_1}^i$  and  $\text{In}_j(\mathbf{s}) \neq \mathbf{a}_{l_2}^j$ , and absent the perturbation,  $\mathbf{s} \rightarrow \mathbf{s}^{(i)}$ ;*
- (3)  $\text{In}_i(\mathbf{s}) \neq \mathbf{a}_{l_1}^i$  and  $\text{In}_j(\mathbf{s}) = \mathbf{a}_{l_2}^j$ , and absent the perturbation,  $\mathbf{s} \rightarrow \mathbf{s}^{(j)}$ .*

From the extension of Proposition 1 and its corollaries to two one-bit perturbations, it is clear how to extend them to more than two one-bit perturbations, albeit, with an increased number of cases.

## 3 ALGORITHMS

### 3.1 Network intervention

One objective of network modeling is to use the model to design intervention strategies for affecting the dynamic evolution of the network. The methods apply whether or not a network possesses a steady-state distribution. We will point out whether the steady-state distribution is affected if it exists. Such intervention studies have focused on three general approaches: state perturbation, optimal control and function perturbation.

With state perturbation, the state of the network is reset to an initial state and the network is allowed to evolve from there, the point being that the new trajectory will visit more desirable states (Shmulevich *et al.*, 2002a). The network structure is not changed. If the network possesses a steady-state distribution, then that distribution is not changed.

For optimal control, there exist one or more controllable variables that affect the transition probabilities of the network and these can be used to desirably affect its dynamic evolution (Datta *et al.*, 2003). The network structure is not changed. If the network possesses a steady-state distribution and the control is applied over a finite time horizon and then stopped, then the

**Table 1.** Definition of functions in the PBN

$x_1x_2x_3$	$f_{11}$	$f_{12}$	$f_{21}$	$f_{31}$	$f_{32}$
000	0	0	0	0	0
001	1	1	1	0	0
010	1	1	1	0	0
011	1	0	0	1	0
100	0	0	1	0	0
101	1	1	1	1	0
110	1	1	0	1	0
111	1	1	1	1	1
$c_{ij}$	0.6	0.4	1	0.5	0.5

steady-state distribution is not changed (Datta *et al.*, 2003; Pal *et al.*, 2005); however, if the control is applied over infinite time (forever), then the steady-state distribution is changed to one whose mass is more concentrated in favorable states (Pal *et al.*, 2006). For instance, based upon a study finding that blocking the Wnt5a protein from activating its receptor could substantially reduce Wnt5a’s ability to induce a metastatic phenotype (Weeraratna *et al.*, 2002), optimal control theory has been applied to an expression-based network including the WNT5A gene in such a manner as to down-regulate WNT5A (Datta *et al.*, 2003; Pal *et al.*, 2005, 2006) the objective being to decrease the likelihood of metastasis.

In the case of function perturbation, one or more Boolean functions are changed to desirably alter the network. The network structure is changed (Shmulevich *et al.*, 2002b). If the network possesses a steady-state distribution, then that distribution is changed.

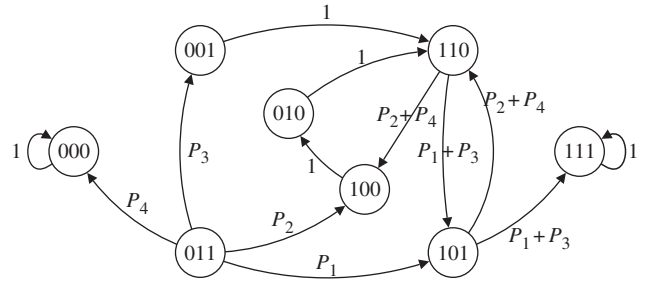
These applications have been in the context of probabilistic BNs; however, since BNs are a special case of probabilistic BNs, the results apply at once. For instance, complexity issues relating to optimal control have been studied in the context of BNs (Akutsu *et al.*, 2006). If a BN possesses random node flips, so that at any time point there is a positive probability of any node flipping from 0 to 1 or from 1 to 0, then it possesses a steady-state distribution. The classical deterministic BNs are free of such random node flips (Kauffman *et al.*, 1993). Thus, unless there is a single attractor cycle, there does not exist a steady-state distribution.

Before formally providing the procedure to control the stationary probabilities in a Probabilistic Boolean Network (PBN) via function perturbation, we revisit a problem considered in Shmulevich *et al.* (2002b) to motivate and illustrate the methodology. In the Discussion section, we will apply the procedure to alter a WNT5A network in order to decrease the likelihood of metastasis.

Given two sets of states, perhaps representing two different cellular functional states or phenotypes, the general problem is to specify some optimization criterion regarding the stationary probabilities of the states and to discover some multiple perturbation of a single function that best achieves the optimization. The analysis is for a PBN. These have been discussed in many places, including review articles, so we leave

**Table 2.** The four components of PBN

Network $i$	Network function $f_i$	Probability $P_i$
BN1	$(f_{11}, f_{21}, f_{31})$	0.3
BN2	$(f_{11}, f_{21}, f_{32})$	0.3
BN3	$(f_{12}, f_{21}, f_{31})$	0.2
BN4	$(f_{12}, f_{21}, f_{32})$	0.2



**Fig. 1.** State transition diagram of the PBN (probability of state perturbation  $P=0$ ).

their precise mathematical formulation to the literature (Shmulevich *et al.*, 2002c). Let us simply state that the analysis in (Shmulevich *et al.*, 2002b) corresponds to an instantaneously random PBN, which means that at each time point each node function is randomly chosen from a set of possible functions and at each time point there is a probability  $P$  that a node value can be flipped. Each set of selected functions corresponds to a single BN, so that probabilistically, each of these corresponds to a realization of the network.

In the example considered by Shmulevich *et al.* (2002b), the PBN consists of three nodes,  $x_1$ ,  $x_2$  and  $x_3$ . Two functions correspond to  $x_1$  with probabilities 0.6 and 0.4 of using the first and second functions, respectively; node  $x_2$  has a single function, and node  $x_3$  also has two functions with equal likelihood of using either. Table 1 lists the functions and their selection probabilities ( $c_{ij}$ ). Thus, the PBN has  $2 \times 1 \times 2 = 4$  constituent BNs (Table 2). The probability of a node flip is  $P=0.01$ , in which case the stationary probabilities of the two singleton attractors are  $\pi(000) = 0.0752$  and  $\pi(111) = 0.7310$ . The intervention objective is to use function perturbation to make both new stationary probabilities,  $\mu(000)$  and  $\mu(111)$ , close to 0.4, which means minimizing the error criterion  $|\mu(000) - 0.4| + |\mu(111) - 0.4|$ , while maintaining their total stationary mass. This corresponds to the objective of reducing the stationary probability of the undesirable state 111 while increasing the stationary probability of the desirable state 000 [see Shmulevich *et al.* (2002b) for a detailed discussion].

The state transition diagrams absent node perturbation ( $P=0$ ) for the four BNs are shown in Figure 2. Let  $\text{BOA}\{x_1x_2x_3\}$  denote the BOA of  $x_1x_2x_3$  and  $|\text{BOA}\{x_1x_2x_3\}|$  denote its size. It can be seen that in both BN1 and BN3,  $|\text{BOA}\{000\}| = 1$  and  $|\text{BOA}\{111\}| = 7$ . In BN2,  $|\text{BOA}\{000\}| = |\text{BOA}\{111\}| = 1$  and in BN4,  $|\text{BOA}\{000\}| = 2$

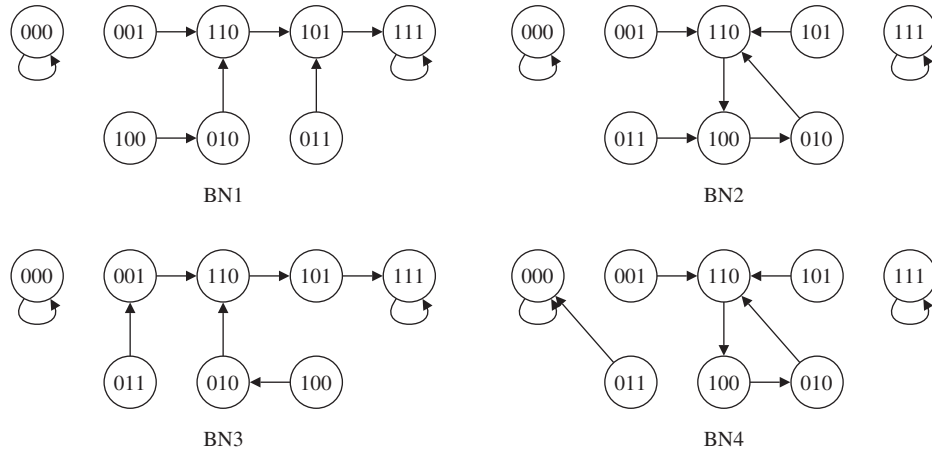


Fig. 2. State transition diagram of the four BNs (probability of state perturbation  $P=0$ ).

and  $|BOA\{111\}| = 1$ . Since the sum of the stationary probabilities of 000 and 111 must not change, we need to increase the BOA of 000 and decrease the BOA of 111. Since function  $f_{21}$  is used in all four BNs, its perturbation will result in changes in all BNs. On the other hand, if we perturb any of the other four functions, only two BNs will be affected. So, we prefer not to perturb  $f_{21}$  unless necessary. Recalling Corollary 1, in this example,  $n = k_i = 3$  and  $2^{n-k_i} = 1$ , so perturbing one row of a function truth table will affect only one state transition.

First, consider BN1 and BN3. Can we increase the BOA of 000 by finding any state whose successor state differs from 000 by only one bit (preferably on the first and third bit)? The answer is 011 of BN3. However, even if we make a one-bit perturbation of function  $f_{31}$  to let  $011 \rightarrow 000$  in BN3,  $|BOA\{000\}|$  will increase by only 1. This perturbation also affects BN1 ( $011 \rightarrow 100$ ), but has no effect on  $BOA\{000\}$  or  $BOA\{111\}$ . Thus we give up this attempt.

Now consider BN2 and BN4. Can we increase  $|BOA\{000\}|$  by a one-bit function perturbation (preferably not  $f_{21}$ )? One possibility is to change the state transition  $110 \rightarrow 100$  to  $110 \rightarrow 000$  in BN2 and BN4, i.e. perturb  $f_{11}$  or  $f_{12}$ . Another possibility is to change the state transition  $100 \rightarrow 010$  to  $100 \rightarrow 011$  in BN4, namely perturb  $f_{32}$  to  $f_{32}^{(5)}$ . Consider the following choices:

Choice 1: Perturb  $f_{11}$  to  $f_{11}^{(7)}$  where  $\mathbf{a}_7^{11} = 110$ . As a consequence, in BN1,  $110 \rightarrow 001$  and  $|BOA\{111\}|$  decreases to 3; in BN2,  $110 \rightarrow 000$  and  $|BOA\{000\}|$  increases to 7. This is a possible candidate.

Choice 2: Perturb  $f_{12}$  to  $f_{12}^{(7)}$ . As a consequence, in BN3,  $110 \rightarrow 001$  and  $|BOA\{111\}|$  decreases to 2; in BN4,  $110 \rightarrow 000$  and  $|BOA\{000\}|$  increases to 7. This is a possible candidate.

Choice 3: Perturb  $f_{32}$  to  $f_{32}^{(5)}$ , where  $\mathbf{a}_5^{32} = 100$ . As a consequence, in BN4,  $100 \rightarrow 011$  and  $|BOA\{000\}|$  increases to 7; in BN2,  $100 \rightarrow 011$ , which does not affect  $BOA\{111\}$  or  $BOA\{000\}$ . Since BN1 and BN3 adopt  $f_{31}$  rather than  $f_{32}$ , they are unaffected. Overall, this perturbation only increases  $|BOA\{000\}|$  in BN4, without affecting  $BOA\{111\}$ . Thus, it cannot achieve the desired goal, which requires increasing

$|BOA\{000\}|$  and decreasing  $|BOA\{111\}|$ . This choice is ruled out.

Simulations show that choice 1 yields stationary probabilities  $\mu(000) = 0.6$  and  $\mu(111) = 0.25$ . Choice 2 yields  $\mu(000) = 0.43$  and  $\mu(111) = 0.41$ , which are close to the goal. Hence, we adopt choice 2. Compare our solution to that of Shmulevich *et al.* (2002b), where the function  $f_{12}$  is perturbed from 0, 1, 1, 0, 0, 1, 1, 1 to 0, 0, 0, 1, 0, 1, 0, 1 (four-bit perturbation, on rows 2, 3, 4 and 7), and the resulting stationary probabilities are  $\mu(000) = 0.4068$  and  $\mu(111) = 0.4128$ . The solution of Shmulevich *et al.* (2002b) is obtained by exhaustive search of all possible 1280 function perturbations (allowing one function to be perturbed by any number of flips in its truth table). Our solution is close to optimal with only a single-bit perturbation and it does not require an exhaustive search.

Using the preceding example as a guide, we have the following general procedure for optimizing the stationary probabilities in BNs or PBNs. Here, we assume one-bit perturbation only.

- (1) Recognize the goal of optimization and formulate the error criterion. In the preceding example, the goal is to have equal stationary probability mass for the target states 000 and 111, while the sum of the probability masses remains unchanged. The error criterion is  $|\mu(000) - 0.4| + |\mu(111) - 0.4|$ . Therefore, we must increase the probability mass of 000 and decrease that of 111 at the same time.
- (2) Determine the priority of perturbation for the functions. For instance, in a PBN, if some of the functions are common in two or more constituent BNs, it is preferred to perturb a function that affects as few BNs as possible. For another example, in a BN, it is more favorable to perturb a function that results in fewer changes in the state transitions.
- (3) Plot the state transition diagrams. Analyze the BOAs of the target states by taking into consideration the BOA sizes and the probabilities of BNs (in the case of a PBN). To increase the BOA of a target state  $\mathbf{s}$  by a one-bit

perturbation, find a candidate state outside  $\text{BOA}\{\mathbf{s}\}$  whose next state differs from a state within  $\text{BOA}\{\mathbf{s}\}$  by only one bit. Find all such candidates. To decrease the BOA of a target state  $\mathbf{s}$  by a one-bit perturbation, find a candidate state in  $\text{BOA}\{\mathbf{s}\}$  whose next state differs from a state outside  $\text{BOA}\{\mathbf{s}\}$  by only one bit. Notice that in a PBN, perturbation of one function can result in changes in two or more constituent BNs.

- (4) List all the options of perturbation, from the highest priority to the lowest. For each option, draw new state transition diagrams and analyze the BOAs of the target states again. Throw away the options that are far from the goal. Notice that the steady-state probability mass of an attractor state is mainly affected by the size of its BOA, but also has to do with the BOA structure and the node-flipping probability  $P$ . Therefore, the BOA sizes can be used to estimate (but are not deterministic of) the probability masses of the target states.
- (5) For the remaining options, make computations either by simulation or by direct computation through Markov chain analysis [see Shmulevich *et al.* (2002a) for details], and pick the option that is closest to the goal. If two or more options are equally good, pick the one with the highest priority (e.g. perturbed function is present in the least number of BNs, or carries the least weight in a pre-defined importance rank, etc.).

### 3.2 Identifying function perturbations

Suppose we have knowledge of the state transitions of a BN. Imagine a perturbation occurs in the Boolean model unbeknownst to us except that we observe the new state transitions. By comparing the state transitions before and after perturbation, we may ask two questions: (1) which Boolean function is perturbed? (2) on which row of the truth table is the perturbation? These are identification problems, useful in diagnosing changes in gene regulatory networks, such as changes caused by a disease, radiation therapy, drug treatment, etc.

We now present a general procedure for identifying function perturbations in a BN:

- (1) Find out perturbed function(s).  
Let the list of state transition of the original BN be given by  $\mathbf{s}_0, \mathbf{s}_1, \dots, \mathbf{s}_{2^n}$ , these being the successor states of  $00 \dots 0, 00 \dots 1, \dots, 11 \dots 1$ , respectively. The list of state transitions of the perturbed BN is denoted by  $\mathbf{s}'_0, \mathbf{s}'_1, \dots, \mathbf{s}'_{2^n}$ . According to Proposition 1, if a state transition is affected by a one-bit perturbation on Boolean function  $f_i$ , then the new successor state differs from the old by the value of node  $x_i$ . Thus, by comparing the two lists, one can tell which function(s) is(are) perturbed.
- (2) Locate the flipped entries of the perturbed function(s).  
For the simple case of a one-bit perturbation in one function  $f_i$ , recall from Corollary 1 that  $2^{n-k_i}$  states will have changed successors. Moreover, those states share the same value on nodes  $x_{j_1}, \dots, x_{j_{k_i}}$ . Assume the differences between the state transition rules before and

after perturbation are given by the states  $\mathbf{s}_{d_1}, \mathbf{s}_{d_2}, \dots, \mathbf{s}_{d_{m_i}}$  versus the states  $\mathbf{s}'_{d_1}, \mathbf{s}'_{d_2}, \dots, \mathbf{s}'_{d_{m_i}}$  and  $k_i$  is unknown. We can find  $k_i$  by computing  $k_i = n - \log_2 m_i$ . Knowing that those states are the successor states of  $(d1)_2, (d2)_2, \dots, (d_{m_i})_2$ , which are the length- $n$  binary representations of decimal numbers  $d1, d2, \dots, d_{m_i}$  (e.g. for a three-node BN,  $(6)_2 = 110$ ), we may compare the states  $(d1)_2, (d2)_2, \dots, (d_{m_i})_2$  to find the common bits in order to identify the parent nodes of  $x_i$ , which are  $x_{i_1}, \dots, x_{i_{k_i}}$ . Moreover, if  $\text{In}_i((d1)_2) = \text{In}_i((d2)_2) \dots = \text{In}_i((d_{m_i})_2) = (u_1, \dots, u_{k_i}) = \mathbf{a}_j^i$ , then we can conclude that the perturbation on  $f_i$  takes place on the  $j$ th row of its truth table.

- (3) For the case of two-bit perturbations, we can refer to Corollaries 4–6 for a similar analysis. Likewise, we can treat more complex perturbations, albeit with increased difficulty.

## 4 DISCUSSION

In this section, we will apply the intervention procedure to beneficially control the gene WNT5A in a network related to melanoma and to identify a function perturbation in a *D.melanogaster* segmentation polarity gene network.

### 4.1 Intervention in a WNT5A network

We consider a BN taken from a context-sensitive PBN model for a WNT5A network constructed for an intervention study (Pal *et al.*, 2005). The original PBN model was inferred from the data collected in a metastatic melanoma study (Bittner *et al.*, 2000) which found that the abundance of the messenger RNA of the WNT5A gene was highly discriminating between cells with properties typically associated with high and low metastatic competence. A subsequent study (Weeraratna *et al.*, 2002) validated and expanded this finding by experimentally increasing the levels of the Wnt5a protein secreted by a melanoma cell line, as a result of which the metastatic competence of the cell line was directly altered. It was also found that through an intervention blocking the Wnt5a protein from activating its receptor, Wnt5a's ability to induce metastasis can be substantially reduced. These results suggest that using an intervention to downregulate the WNT5A gene can lower the chance of metastasis in a cell line.

In Pal *et al.* (2005) seven genes, WNT5A, pirin, S100P, RET1, HADHB and STC2, were selected for inference and intervention in a melanoma network. The objective was to exert a control variable for a finite time to steer the network dynamics towards desirable states, those for which  $\text{WNT5A} = 0$ , not highly expressed. Their method uses an external control to change the state transitions temporarily instead of changing the network structure. In our study, we will use a different approach and permanently alter the network structure by a minimal amount of function perturbation.

To study the chosen BN, we relabel the seven genes, WNT5A, pirin, etc. as  $x_1, x_2, \dots, x_7$ . The functions are defined in Table 3, where under the heading 'function values' each item is a binary string whose  $i$ th bit represents the function value on

**Table 3.** Definition of a WNT5A BN

Function	Input variables	Function values
$f_1$	$x_6$	10
$f_2$	$x_2, x_4, x_6$	00010111
$f_3$	$x_3, x_4, x_7$	10101010
$f_4$	$x_4, x_6, x_7$	00001111
$f_5$	$x_2, x_5, x_7$	10101111
$f_6$	$x_2, x_3, x_4$	01110111
$f_7$	$x_2, x_7$	1101

the  $i$ th row of the truth table. For instance, in the last row the entry 1101 means that for the inputs 00, 01, 10, 11 the function outputs are 1, 1, 0, 1, respectively.

This BN has four attractors 0101111, 0110110, 0111110 and 1000001. Their BOA sizes are 48, 4, 16 and 60, respectively. The last attractor  $\mathbf{s} = 1000001$  is undesirable, because WNT5A gene is up-regulated. Moreover,  $\mathbf{s}$  has a large BOA (consisting of nearly 50% of the total number of states), so the possibility of reaching it is high. Our objective is to eliminate this attractor or minimize its BOA if elimination is impossible, that is,  $\min|\text{BOA}\{\mathbf{s}\}|$ . We will achieve this goal through function perturbations, with two constraints: (1) to perturb as few bits in the functions as possible; (2) to affect as few state transitions as possible.

For constraint (2), recall that the number of affected state transitions by one-bit perturbation equals  $2^{n-k_i}$ , and it is preferable to choose from the following functions for perturbation:  $f_2, f_3, f_4, f_5$  and  $f_6$ . As a result, 16 state transitions (1/8 of the total) will be affected.

To eliminate attractor  $\mathbf{s}$ , we will choose from the above five functions for one-bit perturbation and follow the steps below.

- (i) In an effort to eliminate attractor  $\mathbf{s}$  by a one-bit perturbation of a function, we wish to change the state transition from  $\mathbf{s} \rightarrow \mathbf{s}$  to  $\mathbf{s} \rightarrow \mathbf{u}$ , such that  $\mathbf{u}$  differs from  $\mathbf{s}$  by exactly one bit. Since the five preferred functions to perturb are given, the candidate states will be  $\mathbf{u}_1 = 1100001$ ,  $\mathbf{u}_2 = 1010001$ ,  $\mathbf{u}_3 = 1001001$ ,  $\mathbf{u}_4 = 1000101$  and  $\mathbf{u}_5 = 1000011$ .
- (ii) Consider the following five options.

Option 1: Change the state transition to  $\mathbf{s} \rightarrow \mathbf{u}_1$ . Noticing  $\text{In}_2(\mathbf{s}) = 000$ , we can achieve it through perturbation  $f_2 \rightarrow f_2^{(1)}$ . By applying the theoretical results, we can easily find that after the perturbation, the other three attractors are unaffected, and that now  $\mathbf{s} \rightarrow 1100001 \leftrightarrow 1000101$ . It can be seen that a new attractor cycle is formed, and its constituent states have  $x_1=1$ , which is undesirable.

Option 2: Change the state transition to  $\mathbf{s} \rightarrow \mathbf{u}_2$ . This can be done through  $f_3 \rightarrow f_3^{(2)}$ . As a result, the other three attractors remain the same, and a new attractor cycle,  $\{1000011, 0010001\}$ , will be formed, and the first constituent state is undesirable ( $x_1=1$ ).

Option 3: Change the state transition to  $\mathbf{s} \rightarrow \mathbf{u}_3$ . This can be done through  $f_4 \rightarrow f_4^{(2)}$ . As a result, the other

three attractors are still the same, while  $\mathbf{s}$  disappears. This is a viable option.

Option 4: Change the state transition to  $\mathbf{s} \rightarrow \mathbf{u}_4$  by  $f_5 \rightarrow f_5^{(2)}$ . The result is a new undesirable attractor cycle,  $\{\mathbf{s}, 1000101\}$ , while the other 3 attractors do not change.

Option 5: Change the state transition to  $\mathbf{s} \rightarrow \mathbf{u}_5$  by  $f_6 \rightarrow f_6^{(2)}$ . The result is a new attractor 0000011, while the other three attractors do not change. This is also a viable option.

- (i) Both options 3 and 5 can eliminate the undesirable attractor without creating a new undesirable one; however, option 3 does not create any new attractor, while option 5 creates a new attractor 0000011. By comparison, option 3 has less impact on the original network, so it is the better solution.

By following the outlined procedure, we are able to eliminate the undesirable attractor associated with high competence of cellular metastasis with a one-bit function perturbation with no other attractors affected or any new attractor created. Moreover, the perturbation is chosen so that a minimum number of state transitions will be affected. All of these are achieved without exhaustive search, and without any complex computation.

#### 4.2 Perturbation identification in a *D.melanogaster* segmentation polarity gene network

We will now apply the perturbation identification strategy to a *D.melanogaster* segmentation polarity gene network (Albert *et al.*, 2003). Consider a BN described by Equation (4) of Albert *et al.* (2003). There are eight nodes (genes),  $wg_1, wg_2, wg_3, wg_4, PTC_1, PTC_2, PTC_3$  and  $PTC_4$ . The functions are defined as follows,

$$\begin{aligned}
 wg_1 &= wg_1 \cdot \overline{wg_2} \cdot \overline{wg_4}, \\
 wg_2 &= wg_2 \cdot \overline{wg_1} \cdot \overline{wg_3}, \\
 wg_3 &= wg_1 + wg_3, \\
 wg_4 &= wg_2 + wg_4, \\
 PTC_1 &= \overline{wg_2} \cdot \overline{wg_4} + PTC_1 \cdot \overline{wg_1} \cdot \overline{wg_3}, \\
 PTC_2 &= \overline{wg_1} \cdot \overline{wg_3} + PTC_2 \cdot \overline{wg_2} \cdot \overline{wg_4}, \\
 PTC_3 &= 1, \\
 PTC_4 &= 1.
 \end{aligned}$$

This network has 10 singleton attractors, 00001111 (15), 00010111 (23), 00011111(31), 00101011 (43), 00101111 (47), 00110011 (51), 01010111 (87), 01011111 (95), 10101011 (171), 10101111 (175) [See Albert *et al.* (2003), Fig. 6]. In the parentheses following each attractor are the corresponding decimal numbers.

Among the attractors, 23 and 31 lead to a wild-type pattern and a variant of a wild-type pattern, respectively. States 43 and 47 lead to patterns without parasegment. Those patterns are well known experimentally. States 87 and 95 lead to patterns similar to wild-types, and 171 and 175 lead to patterns similar to non-parasegment patterns, but these patterns are not

observed experimentally. Assume the network is modified so that the attractors 87, 95, 171 and 175 disappear, leaving six attractors. Under the modification, suppose state  $87 \in \text{BOA}\{23\}$ ,  $95 \in \text{BOA}\{31\}$ ,  $171 \in \text{BOA}\{43\}$  and  $175 \in \text{BOA}\{47\}$ , each reaching its attractor in one step. Suppose we have no knowledge about other changes in the network. Based on the partial knowledge, can we find out how the network is modified?

States 87 (01010111) and 23 (00010111) differ by the second bit. States 95 (01011111) and 31 (00011111) also differ by the second bit. States 171 and 43 differ by the first bit. States 175 and 47 differ by the first bit too. According to Proposition 1, perturbations occur on the function for gene  $wg_2$  and on the function for gene  $wg_1$ . The former function has genes  $wg_1$ ,  $wg_2$  and  $wg_3$  as inputs, and in both states 87 and 95,  $(wg_1, wg_2, wg_3) = 010 = \mathbf{a}_3^2$ , so the third row of the truth table is flipped. The latter function has genes  $wg_1, wg_2$  and  $wg_4$  as inputs, and in both states 171 and 175,  $(wg_1, wg_2, wg_4) = 100 = \mathbf{a}_3^1$ , so the fifth row of the truth table is flipped. The new definitions for the two functions are:

$$wg_1 = \begin{cases} wg_1 \cdot \overline{wg_2} \cdot \overline{wg_4}, & \text{if } (wg_1, wg_2, wg_4) \neq 100, \\ 1 - wg_1 \cdot \overline{wg_2} \cdot \overline{wg_4}, & \text{if } (wg_1, wg_2, wg_4) = 100, \end{cases}$$

$$wg_2 = \begin{cases} wg_2 \cdot \overline{wg_1} \cdot \overline{wg_3}, & \text{if } (wg_1, wg_2, wg_3) \neq 010, \\ 1 - wg_2 \cdot \overline{wg_1} \cdot \overline{wg_3}, & \text{if } (wg_1, wg_2, wg_3) = 010 \end{cases}$$

Simulation results of the new BN with the above modifications agree with our conclusion.

To make the above identification, we do not require complete knowledge of the state transitions. This is because even a one-bit difference in a single function can result in  $2^{n-k_i}$  changes in state transitions, and when  $n > k_i$ , there is a lot of redundant information.

### 4.3 Concluding remarks

This article provides several analytical results concerning the perturbation of functions in a BN, and in doing so extends previous work on network stability in a new direction: the effect of structural perturbation on network stability and long-run behavior. It shows how to apply the analytical results to control the stationary probabilities of states in a PBN and has applied the method to intervene in a WNT5A network to avoid high-competence metastatic cellular states. It shows how to use the analytical results to identify function perturbations when changes in network behavior are observed and has applied this method to identify structural changes made in a *D.melanogaster* polarity gene

network. The application procedures do not require exhaustive searches or complex computation.

### ACKNOWLEDGEMENTS

This research was supported in part by the National Science Foundation (CCF-0514644 and BES-0536679).

*Conflict of Interest:* none declared.

### REFERENCES

- Akutsu, T. *et al.* (2007) Control of Boolean networks: hardness results and algorithms for tree structured networks. *Theor. Biol.*, **244**, 670–679.
- Albert, R. and Othmer, H.G. (2003) The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in *Drosophila melanogaster*. *J. Theor. Biol.*, **223**, 1–18.
- Aldana, M. and Cluzel, P. (2003) A natural class of robust networks. *Proc. Natl Acad. Sci. USA*, **100**, 8710–8714.
- Bittner, M. *et al.* (2000) Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, **406**, 536–540.
- Chaves, M. *et al.* (2005) Robustness and fragility of Boolean models for genetic regulatory network. *J. Theor. Biol.*, **235**, 431–449.
- Datta, A. *et al.* (2003) Control in Markovian genetic regulatory networks. *Mach. Learn.*, **52**, 169–191.
- Gershenson, C. *et al.* (2006) The role of redundancy in the robustness of random Boolean networks. *Artificial Life X, Proceedings of the Tenth International Conference on the Simulation and Synthesis of Living Systems* (MIT Press).
- Huang, S. (1999) Gene expression profiling, genetic networks, and cellular states: An integrating concept for tumorigenesis and drug discovery. *Mol. Med.*, **77**, 469–480.
- Kauffman, S.A. (1993) *The Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press, New York.
- Kauffman, S.A. *et al.* (2004) Genetic networks with canalizing Boolean rules are always stable. *Proc. Natl Acad. Sci. USA*, **101**, 17102–17107.
- Pal, R. *et al.* (2005) Intervention in context-sensitive probabilistic Boolean networks. *Bioinformatics*, **21**, 1211–1218.
- Pal, R. *et al.* (2006) Optimal infinite horizon control for probabilistic Boolean networks. *IEEE Trans. Signal Process.*, **54**, 2375–2387.
- Qu, X. *et al.* (2002) Numerical and theoretical studies of noise effects in the Kauffman model. *J. Stat. Phys.*, **109**, 967–986.
- Ramo, P. *et al.* (2005) Stability of functions in Boolean models of gene regulatory networks. *Chaos*, **15**, 034101.
- Shmulevich, I. *et al.* (2002a) Gene perturbation and intervention in probabilistic Boolean networks. *Bioinformatics*, **18**, 1319–1331.
- Shmulevich, I. *et al.* (2002b) Control of stationary behavior in probabilistic Boolean networks by means of structural intervention. *J. Biol. Syst.*, **10**, 431–445.
- Shmulevich, I. *et al.* (2002c) From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proc. IEEE*, **90**, 1778–1792.
- Shmulevich, I. *et al.* (2003) The role of certain Post classes in Boolean network models of genetic networks. *Proc. Natl Acad. Sci. USA*, **100**, 10734–10739.
- Weeraratna, A. T. *et al.* (2002) Wnt5a signaling directly affects cell motility and invasion of metastatic melanoma. *Canc. Cell*, **1**, 279–288.