CHAPTER

# 8

# FINDING LONG-TERM INFLUENCE AND SENSITIVITY OF GENES USING PROBABILISTIC GENETIC REGULATORY NETWORKS[a]

**Q.N. Tran**
*The University of South Dakota, Vermillion, SD, United States*

## 8.1 INTRODUCTION

Boolean networks are well-studied discrete models of biological networks such as gene regulatory networks where DNA segments in a cell interact with each other indirectly through their RNA and protein expression products or with other substances in the cell, thereby governing the rates at which genes in the network are transcribed into mRNA. A Boolean network consists of a set of Boolean variables whose state is determined by other variables in the network. They are a particular case of discrete dynamical networks, where time and states are discrete. A Boolean network can be considered as a directed graph where the nodes represent the expression status of genes and directed edges represent the actions of genes on other genes. Each node $x_i \in \{0,1\}$, $i = 1,\ldots,n$, is a Boolean variable whose state value at time $t + 1$ is completely determined by the state values of nodes $x_{j_1}, x_{j_2}, \ldots, x_{j_l}$ for some $1 \leq l \leq n$ at time $t$ by means of a Boolean function $f^{(i)} : \{0,1\}^l \rightarrow \{0,1\}$ when there are edges from $x_{j_k}$ to $x_i$ for all $k = 1,\ldots,l$. Thus, one can write $x_i(t+1) = f^{(i)}(x_{j_1}(t), x_{j_2}(t), \ldots, x_{j_l}(t))$, $i = 1,\ldots,n$.

Probabilistic Boolean genetic regulatory networks (PBNs) are probabilistic or stochastic generalizations of Boolean networks. In these models, the deterministic dynamics are replaced by probabilistic dynamics, which can be framed within the mature and well-established theory of Markov chains, for which many analytical and numerical tools have been developed. The value of node $x_i$ at time $t + 1$ is now specified by possibly different Boolean functions and state transition probabilities

$$x_i(t+1) = \begin{cases} f_1^{(i)}(x_{j_1}(t), x_{j_2}(t), \ldots, x_{j_l}(t)) & \text{prob. } p_1^{(i)} \\ \cdots \\ f_m^{(i)}(x_{j_1}(t), x_{j_2}(t), \ldots, x_{j_l}(t)) & \text{prob. } p_m^{(i)} \end{cases} \quad (1)$$

where $p_k^{(i)} \in [0,1]$, $1 \leq k \leq m$, and $\sum_{k=1}^{m} p_k^{(i)} = 1$.

This computational tool has been used in system biology to study biological systems from a holistic perspective to provide a comprehensive, system-level understanding of cellular behavior. PBN

[a]A preliminary version of this article was presented at the ACM-BCB 2015 conference in Atlanta, GA.

**Emerging Trends in Applications and Infrastructures for Computational Biology, Bioinformatics, and Systems Biology.**
http://dx.doi.org/10.1016/B978-0-12-804203-8.00008-0

modeling can be used for the design and analysis of intervention strategies for moving the networks out of undesirable states such as those associated with diseases into the more desirable ones. PBNs have also been used to study the analysis and control of biological networks to find a method for a suitable medication that can be used for drug discovery and cancer treatment [1–11].

Boolean networks are special cases of PBNs in which state transition probabilities are either 1 or 0. The probabilistic nature of this PBN model affords flexibility and power in terms of making inferences from data, which necessarily contain uncertainty, as well as in terms of understanding the dynamical behavior of biological networks, particularly in relation to their structure. PBN is a discrete-time Markov chain in that the behavior at each point in time can be described by a discrete probabilistic choice over several possible outcomes.

Unfortunately, modeling of gene regulatory networks often leads to dynamic models with huge state space surpassing the size of any computer systems by orders of magnitude. One of the key aspects in the analysis of PBN is the investigation of their long-term behavior such as the attractors of the system, which were hypothesized to characterize cellular phenotype [12–14].

Markov chain Monte Carlo (MCMC) has been proposed for analyzing long-term behavior distribution by running the Markov chain for a sufficiently long time until convergence into the stationary distribution and observing the proportion of time the process spent in the parts of the state space that represent the information of interest such as the joint stationary distribution of several specific genes [1, 2, 7, 9]. Due to the difficulties with the assessment of the convergence rate to the long-term distribution, approximation such as the two-state Markov chain can be used to empirically determine when to stop the simulation and output estimates. Unfortunately, the actual inference step of the two-state Markov chain is challenging and to our knowledge the method has not been widely applied for the analysis of large PBNs.

One of the biggest obstacles for the existing methods to analyze the long-term behavior genes of a PBN is the requirement to compute the state transition diagram, which has $2^n$ nodes for a given PBN with $n$ states. In this chapter, we propose a new method in which the state transition diagram is not needed. We utilize algebraic computation for the direct computation of the long-term influence and sensitivity of genes in a PBN. Our novel method only requires $O(n^2)$ memory space — a significant improvement in terms of space as well as time complexity. We are able to analyze the long-term behavior of genes in artificial PBNs with 500 genes within minutes on a desktop computer. We then compare our novel method with previously known methods and report experimental results to illustrate our theoretical arguments. Additionally, our method enables the calculation of likelihood of the occurrence of certain events of interest, thus allowing quantitative statements to be made about the system's behavior, expressed as probabilities or expectations of biological systems.

## 8.2 INFLUENCE AND SENSITIVITY FACTORS OF GENES IN PBNs

In a PBN, some genes may be more important than others in determining the value of a target gene. Finding the genes that have the most potent effect is an important task in studying the PBN. For example, if gene $x_1$ has the following predictors

$$x_1(t+1) = \begin{cases} f_1^{(1)}(x_1(t),x_2(t),x_3(t)) = x_2 & \text{prob. } 0.7 \\ f_2^{(1)}(x_1(t),x_2(t),x_3(t)) = x_2 + x_1 \cdot x_3 & \text{prob. } 0.3 \end{cases} \tag{2}$$

then $x_2$ is a more important variable in influencing gene $x_1$. Similarly, some genes may be more stable while other genes have little effect on it.

There are many examples of such biased regulation of genes from biologists. The cell cycle regulator gene p21, which is a potent cyclin-dependent kinase inhibitor, can be transcriptionally activated by a series of genes: p53, smad4, AP2, BRCA1, and others. Among those genes, p53 has the most potent effect [15].

In this chapter, we assume that a PBN has been built from experimental data. Much research on building a PBN can be found from recent works on building methods such as the coefficient of determination [16–19]. We will concentrate on analyzing the long-term influence and sensitivity of genes in a given PBN and will present a computational method for direct computation of the long-term influence and sensitivity of genes in a PBN.

Following the work of [10], we use the notion of partial derivatives of Boolean functions in defining the influence of a gene. Note that the expressiveness of Boolean algebras is significantly extended by Boolean differential calculus [20, 21]. The additionally defined differentials of Boolean variables, differentials, and further differential operators of Boolean functions as well as several derivative operations of Boolean functions allow to model changes of function values together with changes of the values of variables and many other properties of Boolean functions.

The partial derivatives of Boolean function $f(x)$ with respect to a variable $x_j$ is defined as

$$\frac{\partial f(x)}{\partial x_j} = f(x_{|x_j \leftarrow 0}) \oplus f(x_{|x_j \leftarrow 1})$$

where $\oplus$ is the *xor* operator and $x_{|x_j \leftarrow k} = (x_1, \ldots, x_{j-1}, k, x_{j+1}, \ldots, x_n)$ for $k = 0,1$. Intuitively, the partial derivative of a Boolean function with respect to the $j$th variable indicates whether or not the function differs along the $j$th dimension. The partial derivative is 0 if switching the value of variable $x_j$ does not change the value of the function and it is 1 otherwise.

The influence of variable $x_j$ on the function $f(x)$ is further defined as the expectation of the partial derivative with respect to the distribution $D(x)$:

$$I_{x_j}(f) = E_D\left[\frac{\partial f(x)}{\partial x_j}\right] = Pr\left[\frac{\partial f(x)}{\partial x_j} = 1\right] = Pr\{f(x_{|x_j \leftarrow 0}) \oplus f(x_{|x_j \leftarrow 1}) = 1\}$$

where $Pr\{f(x_{|x_j \leftarrow 0}) \oplus f(x_{|x_j \leftarrow 1}) = 1\}$ is the probability that a toggle of the $j$th variable changes the value of the function $f(x)$ [22].

### 8.2.1 INFLUENCE FACTOR OF GENES

Let $F_i = \{f_1^{(i)}, \ldots, f_{l(i)}^{(i)}\}$ be the set of predictors for gene $x_i$. The influence of variable $x_j$ on variable $x_i$ is defined as

$$I_{x_j}(x_i) = \sum_{k=1}^{l(i)} I_{x_j}(f_k^{(i)}) \cdot c_k^{(i)}.$$

Even though this intuitive formula has been used for other purposes before [22], we now construct a matrix $A$ of influences as $A_{i,j} = I_{x_j}(x_i)$ for calculating the long-term influence of genes based upon this formula. A graph of influence can be constructed where vertices are genes. There is an edge from node $j$ to node $i$ if gene $x_j$ should transfer its influence to gene $x_i$. For example, in Fig. 8.1 gene 1 has three outgoing edges, so it will transfer its influences to gene 1, 2, and 3. In general, if a node has $k$ outgoing

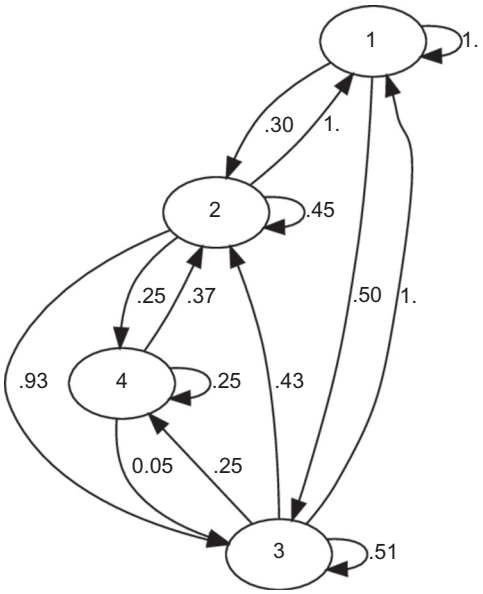



**FIGURE 8.1**

Influence graph for Ex.4.

edges, it will pass on its importance to each of the nodes that it links to. Hence, we will normalize each column of matrix $A$ so that $\sum_{i=1}^{n} A_{ij} = 1$, for $j = 1,\ldots,n$. In other words, we defined a column-stochastic matrix when all of its entries are nonnegative and the entries in each column sum to one.

Conversely, gene 3 has four incoming edges, so it will be influenced by gene 1 with probability 0.50, gene 2 with probability 0.93, gene 3 with probability 0.51, and gene 4 with probability 0.05 at time $t + 1$. However, gene 1 was influenced by gene 2 with probability 1.0 at the previous time $t$. That said, besides the direct influence with probability 0.93 from gene 2 to gene 3 at time $t + 1$, gene 2 also indirectly influences gene 3 at the previous time through gene 1. In general, if a node has $k$ incoming edges, it will be influenced by each of the nodes that it is linked to. From our calculation, if a gene is influenced at a higher rate then it is more sensitive to changes from other genes.

Suppose that initially the *influence factor* or sensitivity is uniformly distributed among the 4 genes, each getting 1/4. In fact, we can use any sensitivity value for the genes that we can determine at the initial time such as the normalized value of $\sum_{i=1}^{n} I_{x_j}(x_i)$ for the $j$th initial value. Denote by $v$ the initial influence or sensitivity vector. Each incoming link increases the influence factor of the gene, so at time $t + 1$, we update the influence factor of each gene by adding to the current value the influence of the incoming links. This is the same as multiplying the matrix $A$ with $v$. At time $t + 1$, the new influence vector is $v_1 = A \cdot v$. We can iterate the process; thus at time $t + 2$, the updated influence vector is $v_2 = A(A \cdot v) = A^2 \cdot v$. We notice that the sequences of iterates $v, A \cdot v,\ldots,A^k \cdot v$ tends to the equilibrium value. We call this the influence vector of our PBN. Clearly, our method requires only $O(n^2)$ space for matrix $A$ and there is no need to build the transition graph of size $O(2^n)$. Notice that Step 3 in Algorithm 1 can be replaced by $A \leftarrow A^2$ as an alternative way to calculate the influence vector of genes.

**ALGORITHM 1 GENEINFLUENCE**

Input: a PBN
Output: influence vector of genes in the PBN
Step 1: Construct a matrix of influences $A$, and an initial influence or sensitivity vector, having all entries equal to $1/n$.
Step 2: Normalize $A$ to make it a column-stochastic matrix.
Step 3: While $\|A \cdot v - v\| > \epsilon$ do $v \leftarrow A \cdot v$.
Step 4: Return $v$.

**Lemma 1.** *Algorithm geneInfluence always terminates and gives an influence vector of genes in the PBN.*

*Proof.* Because $A$ is a column-stochastic square matrix, $A$ and its transpose $A^T$ share the same characteristic polynomial and hence have the same eigenvalues. It is easy to see that $A^T \cdot e = e$, so that 1 is an eigenvalue for $A^T$ and hence for $A$ where $e$ denote an $n$ dimensional column vector with all entries equal to 1. □

### 8.2.2 IMPACT FACTOR OF GENES

In the same manner, we can obtain a matrix for how a gene impacts other predictors by defining $B = A^T$. In other words, $B_{i,j} = A_{j,i}$. For example, in Fig. 8.1 gene 1 has three outgoing edges, so it will pass on its influences to gene 1, 2, and 3. Hence, in the corresponding graph for matrix $B$, which we call the impact graph of the PBN, the outgoing edges were reversed to become incoming edges for gene 1. In general, if a node has $k$ incoming edges in Fig. 8.1, it is impacted by the nodes that it is linked from.

Suppose that initially the importance or *impact factor* is uniformly distributed among the 4 nodes, each getting 1/4. In fact, we can use any impact value for the genes at the initial time such as the normalized value of $\sum_{j=1}^{n} I_{x_i}(x_j)$ for the $i$th initial value. Denote by $w$ the initial impact vector. Each incoming link increases the impact factor of the gene, so at time $t_0 - 1$, we update the impact factor of each gene by adding to the current value the impact of the incoming links. This is the same as multiplying matrix $B$ with $w$. At time $t_0 - 1$, the new influence vector is $w_1 = B \cdot w$. We can iterate the process, thus at time $t_0 - 2$, the updated influence vector is $w_2 = B(B \cdot w) = B^2 \cdot w$. We notice that the sequences of iterates $w$, $B \cdot w, \ldots, B^k \cdot w$ tends to the equilibrium value. We call this the impact vector of our PBN.

### 8.2.3 BOOLEAN ALGEBRA

In this section, we will lay down an algebraic framework for the calculation of the partial derivatives of Boolean function. In practice, for a PBN of $n$ genes the predictor functions have only $k$ variables, where $k \ll n$ and $k$ is the in-degree of the networks. In any case, the following algebraic approach will replace the need for building the truth tables for the calculation of the partial derivatives.

Boolean algebras, which were introduced by Boole in the 1850s to codify the laws of thought, have become a popular topic of research since then. The discovery in the 1930s of the duality between Boolean algebras and Boolean spaces by Stone [23–25] was a major breakthrough of the field. Stone also proved that Boolean algebras and Boolean rings are the same in the sense that one can convert from one algebraic structure to the other. In spite of its long history and elegant algebraic properties, the Boolean ring representation has rarely been used in the computational context.

**Definition 2.** A ring $\mathbf{K} = \langle K,+,\cdot,0,1\rangle$ is Boolean if $\mathbf{K}$ satisfies $x^2 \approx x, \forall x \in K$.

**Lemma 3.** *If $\mathbf{K}$ is a Boolean ring, then $\mathbf{K}$ is commutative and $x + x \approx 0$ [25].*

Every Boolean algebra $(K, \wedge, \vee)$ gives rise to a ring $(K, +, \cdot)$ by defining $a + b = (a \wedge \neg b) \vee (b \wedge \neg a)$ (this operation is called XOR in the case of logic) and $a \cdot b = a \wedge b$. The zero element of this ring coincides with the 0 of the Boolean algebra; the multiplicative identity element of the ring is the 1 of the Boolean algebra. Conversely, if a Boolean ring $\mathbf{K}$ is given, we can turn it into a Boolean algebra by defining $x \vee y = x + y + x \cdot y$ and $x \wedge y = x \cdot y$. Because these two sets of operations are inverses of each other, we can say that every Boolean ring arises from a Boolean algebra, and vice versa. Furthermore, a map $f : A \to B$ is a homomorphism of Boolean algebras if and only if it is a homomorphism of Boolean rings. The categories of Boolean rings and Boolean algebras are equivalent. By using these translations, there exists a Boolean polynomial for each Boolean formula and vice versa.

Since congruences on rings are associated with ideals, it follows that the same must hold for Boolean algebras. An ideal of the Boolean algebra $\mathbf{K}$ is a subset $I$ such that $\forall x,y \in I$ we have $x \vee y \in I$ and $\forall a \in K$ we have $a \wedge x \in I$. This notion of ideal coincides with the notion of ring ideal in the Boolean ring $\mathbf{K}$. An ideal $I$ of $R$ is called prime if $I \neq K$ and if $a \wedge b \in I$ always implies $a \in I$ or $b \in I$. An ideal $I$ of $K$ is called maximal if $I \neq K$ and if the only ideal properly containing $I$ is $K$ itself. These notions coincide with ring theoretic ones of prime ideal and maximal ideal in the Boolean ring $\mathbf{K}$.

By using Boolean ring representation we can convert our problem of counting satisfiable solutions for $Pr\{f(x_{|x_j \leftarrow 0}) \oplus f(x_{|x_j \leftarrow 1}) = 1\}$ into finding the solutions for a Boolean polynomial with degree of at most 1 in all variables. This is the main reason why we want to use Boolean algebra in this chapter.

We now provide an example PBN and show how to build an influence graph and how to calculate the influence factor as well as the impact factor of the PBN.

Example 4. Given a PBN consisting of four genes $V = \{v_1, v_2, v_3, v_4\}$ and a set of predictors in Fig. 8.2. Notice that the probabilities have been rounded off. The actual numbers are [[1.], [.2461853940, .3784104050, .3609895426, 0.1441465845e-1], [.9319607618, 0.6803923820e-1], [1.]] for $v_1, v_2, v_3$, and $v_4$, respectively. The influence matrix of this PBN before normalization is shown in Table 8.1 and the influence graph of the PBN is shown in Fig. 8.1.

Suppose that initially the influence factor or sensitivity is uniformly distributed among the 4 genes, each getting 1/4. The long-term sensitivity or influence vector converged to [0.450595824829880, 0.207132994504042, 0.264470757717693, 0.0778004230596858]. That said, gene 4 is the least sensitive gene or in other words the most stable gene in the long run. Gene 1 is the most sensitive gene in the PBN.

$$v_1(t+1) = v_1 + v_2 + v_3 \quad \text{prob. } 1.00$$

$$v_2(t+1) = \begin{cases} v_1 \cdot v_2 \cdot v_3 + v_1, & \text{prob. } 0.246 \\ v_2 \cdot v_3 \cdot v_4 + v_3 + v_4 + 1, & \text{prob. } 0.378 \\ v_1 \cdot v_2 \cdot v_4 + v_1 \cdot v_3 \cdot v_4 + v_2 \cdot v_3 \cdot v_4 + v_1 \cdot v_2 + v_2 \cdot v_3 + v_2 \cdot v_4 + v_3 \cdot v_4, & \text{prob. } 0.361 \\ v_1 \cdot v_2 \cdot v_4 + v_2 \cdot v_4 + v_2 & \text{prob. } 0.014 \end{cases}$$

$$v_3(t+1) = \begin{cases} v_1 \cdot v_3 + v_2 + 1, & \text{prob. } 0.932 \\ v_1 \cdot v_3 \cdot v_4 + v_3 \cdot v_4 + v_1 + v_3 + v_4 & \text{prob. } 0.068 \end{cases}$$

$$v_4(t+1) = v_2 \cdot v_3 \cdot v_4 + v_2 \cdot v_3 \quad \text{prob. } 1.00$$

**FIGURE 8.2**

Set of predictors for Ex. 4.

| Table 8.1 Influence Matrix for Ex. 4 | | | |
|---|---|---|---|
| **1** | **1** | **1** | **0** |
| 0.30 | 0.45 | 0.43 | 0.37 |
| 0.50 | 0.93 | 0.51 | 0.05 |
| 0 | 0.25 | 0.25 | 0.25 |

Similarly, The long-term impact factor or impact vector converged to [0.210910910659655, 0.357142856843150, 0.292460317400817, 0.139485914740911]. That said, gene 2 is the most impacting gene in the PBN. Notice that the most stable gene of the PBN is not necessarily the most impacting gene of the networks. We have many examples showing that they are in fact not correlated.

## 8.3 A BIOLOGICAL CASE STUDY

To illustrate the efficiency and the accuracy of our novel algorithm, we use not only randomly generated PBNs but also the data from a human glioma gene expression data set [26]. Fig. 8.3 shows the averaged running time in seconds of our algorithm for finding the impact factor of genes in randomly generated PBNs with 20, 30, 50, 100, 250, and 500 genes. There are two sets of randomly generated PBNs, one with the in-degree of 5 and one with the in-degree of 10. Furthermore, for each data point we generated 10 PBNs and took the averaged running time. Our algorithms were implemented as a software package in Maple 2015. All experiments were performed on a workstation using an i7 CPU, 16GB of RAM and Linux Ubuntu 14.04.3 LTS.

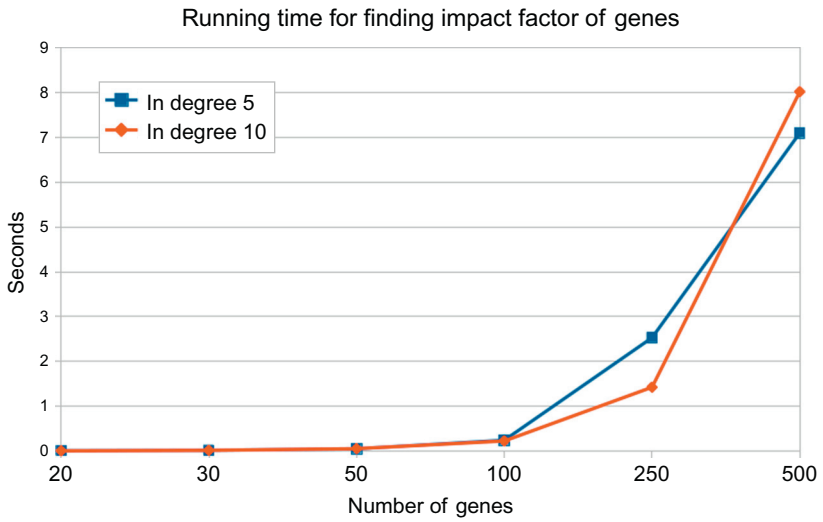



**FIGURE 8.3**

Experiments with randomly generated PBNs.

### 8.3.1 GLIOMAS CASE STUDY

Gliomas are the most common form of primary malignancies of the central nervous system (CNS) mainly affecting adults. These tumors have a histological resemblance to different types of glial cells and are categorized into astrocytomas, oligodendrogliomas, oligoastrocytomas, and ependymomas, based on the predominant cell type(s) in the respective tumor.

A PBN network of 597 genes was inferred using the coefficient of determination as in [2, 10]. A small subnet of 15 genes is shown in Fig. 8.4 with the weights on the edges representing the influences of the genes. The influence matrix of this PBN after normalization is shown in Table 8.2.

With the complexity of predictor functions, some gene influences are pretty small. To deal with many small influences we use the same idea of damping factor in Google's PageRank [27] where the influence matrix is replaced by $(1 - p) \cdot A + p \cdot T$ where $T$ is the $n \times n$ "teleportation" matrix; that is, the matrix each of whose entries is $1/n$. We use $p = 0.10$ in our calculation.

Suppose that initially the influence factor or sensitivity is uniformly distributed among the genes. The long-term sensitivity or influence vector converged and after normalizing we have a *converged influence vector* [0.987752376216898, 0.613897953484682, 0.509838752830795, 0.163205931648281, 0.150098397974440, 0.800165646879760, 0.181809171192399, 0.656777507172725, 0.0496073065684904, 0.519537291701068, 0.369183660877093, 0.238311660580107, 0.0814175288392782, 1., 0.981858486296723] for genes Tie-2, TGFB3, ERCC1, HSP40, TDPX2, GSTP1, GNB1, NDKB, TOP2A, SCYB10, PDGFA, NKEFB, $\beta$-Actin, NKFB1, and BCL2A1, respectively. Some obvious expectation that a gene such as TOP2A should be stable and should not be sensitive to the influence of other genes can be verified from its lowest influence factor.

It is worth nothing that our novel method is not only very efficient (finishing in 0.225 s) but also capable of identifying stable genes and sensitive genes that were not known before, as we will explain the next three subsections.

### 8.3.2 STABLE GENES

From the converged influence vector, it shows that genes HSP40, TDPX2, GNB1, NKEFB, and $\beta$-Actin are stable or not sensitive to the influence of other genes in a long run. In Table 8.3, we see the similar pattern of these genes when the influence vector converges to its fixed point.

Compared with the results from previous computational methods such as the steady state analysis using the two-state Markov chain approach [1], our results are much closer to what biologists have known about this disease. For example, the two-state Markov chain approach can only tell us that among three genes Tie-2, TGF$\beta_3$, and NKFB, the state where Tie-2 is OFF, TGF$\beta_3$ is ON, and NKFB is ON will have the highest probability.

### 8.3.3 SENSITIVE GENES

On the other hand, we found that genes Tie-2, GSTP1, NFKB1, and BCL2A1 are very sensitive to the influence of other genes in a long run. Again, Table 8.3 shows the similar pattern of these genes when the influence vector converges to its fixed point. This finding is in line with what we have learned from biologists as [28] found that Tie2 activation was related to the up-regulation of integrin beta1 levels and

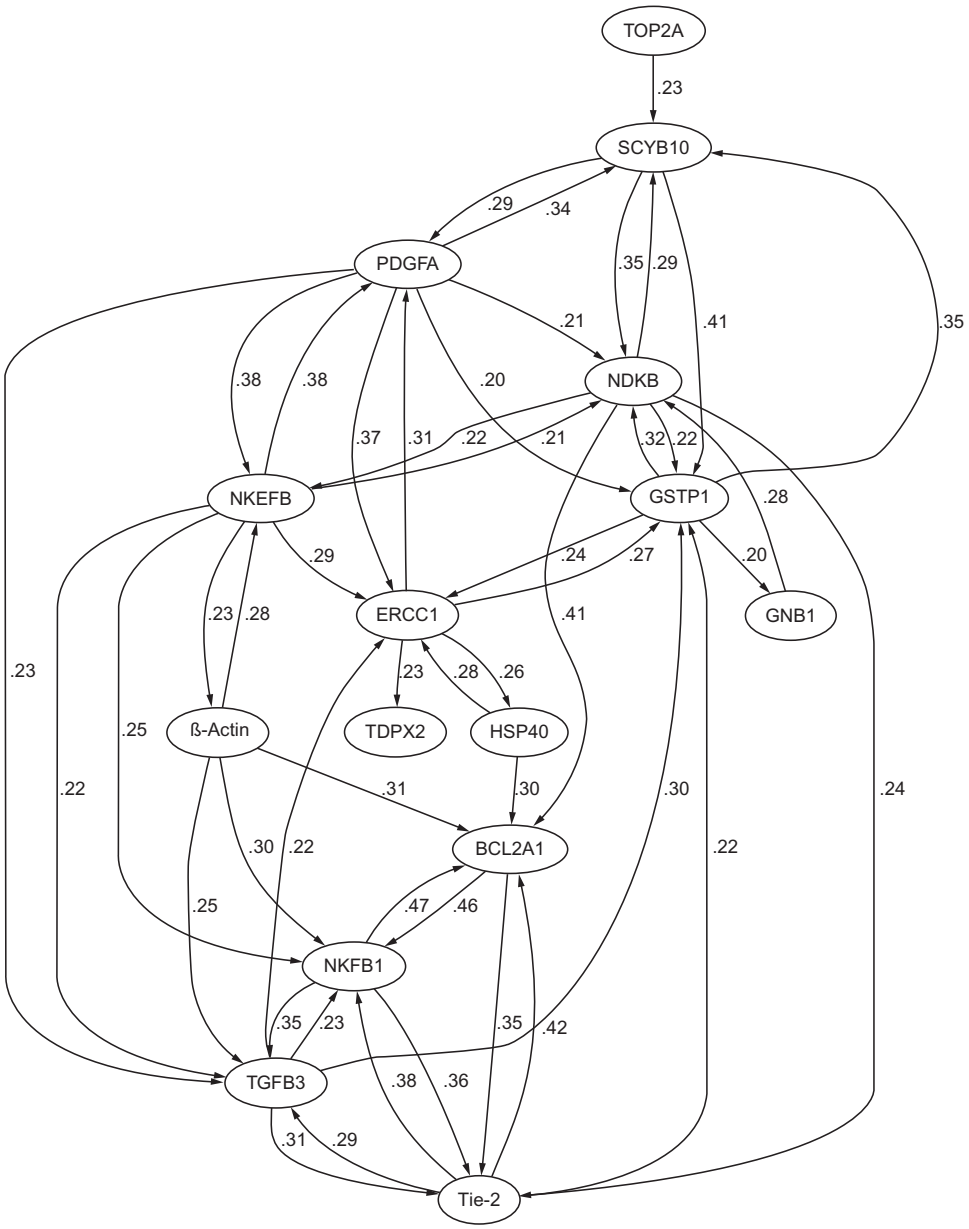
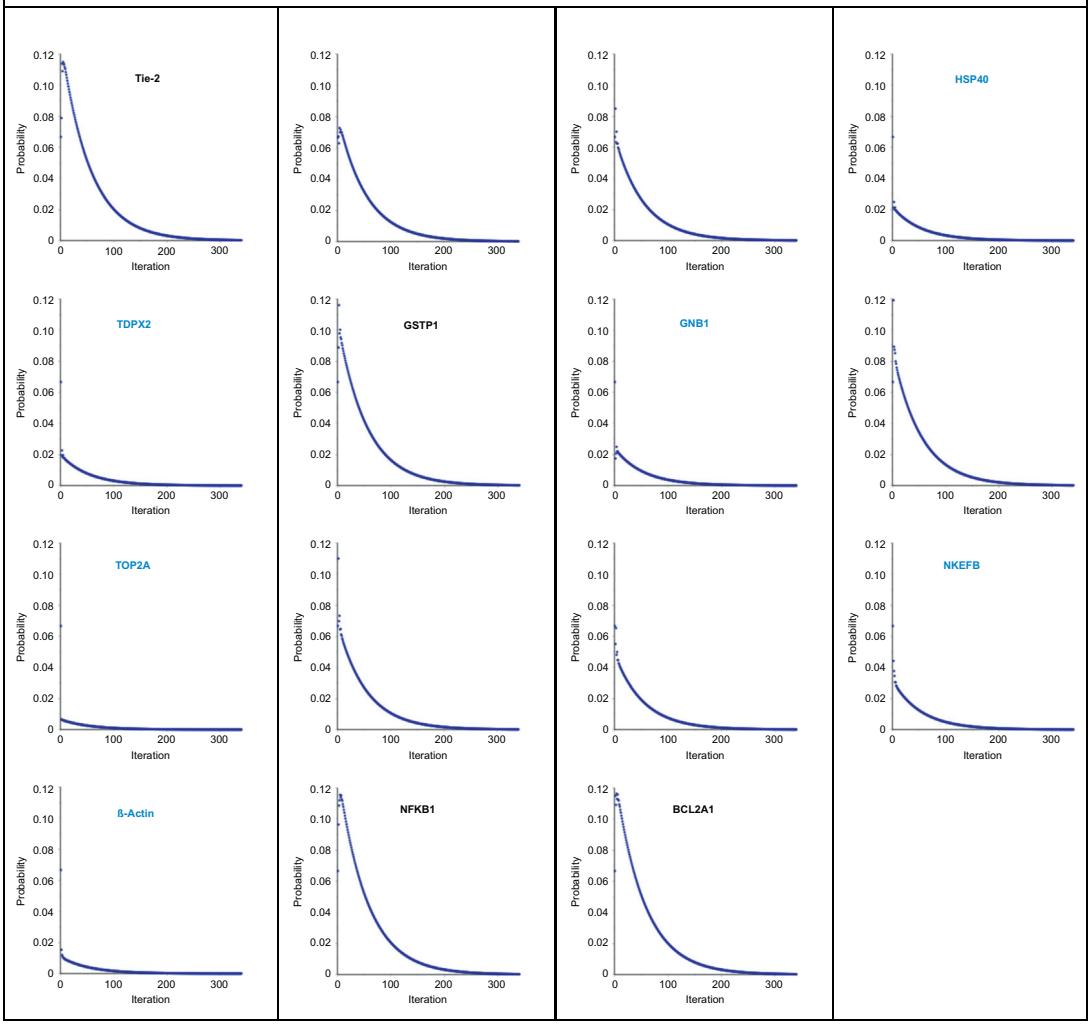



**FIGURE 8.4**

Influence graph for gliomas.

| **Table 8.2 Influence Matrix for Gliomas** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.29 | 0 | 0 | 0 | 0 | 0 | 0.17 | 0 | 0 | 0 | 0 | 0 | 0.31 | 0.43 |
| 0.22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.13 | 0.14 | 0.22 | 0.30 | 0 |
| 0 | 0.21 | 0 | 0.48 | 0 | 0.22 | 0 | 0 | 0 | 0 | 0.21 | 0.18 | 0 | 0 | 0 |
| 0 | 0 | 0.24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0.21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0.17 | 0.28 | 0.25 | 0 | 0 | 0 | 0 | 0.16 | 0 | 0.39 | 0.12 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0.18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0.29 | 1.00 | 0 | 0 | 0.33 | 0.12 | 0.13 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0.32 | 0 | 0.21 | 1.00 | 0 | 0.20 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0.29 | 0 | 0 | 0 | 0 | 0 | 0 | 0.28 | 0 | 0.24 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.16 | 0 | 0 | 0.22 | 0 | 0.25 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.15 | 0 | 0 | 0 |
| 0.29 | 0.22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.16 | 0.26 | 0 | 0.57 |
| 0.32 | 0 | 0 | 0.52 | 0 | 0 | 0 | 0.30 | 0 | 0 | 0 | 0 | 0.27 | 0.40 | 0 |

**Table 8.3 Convergence of Influence Factors**



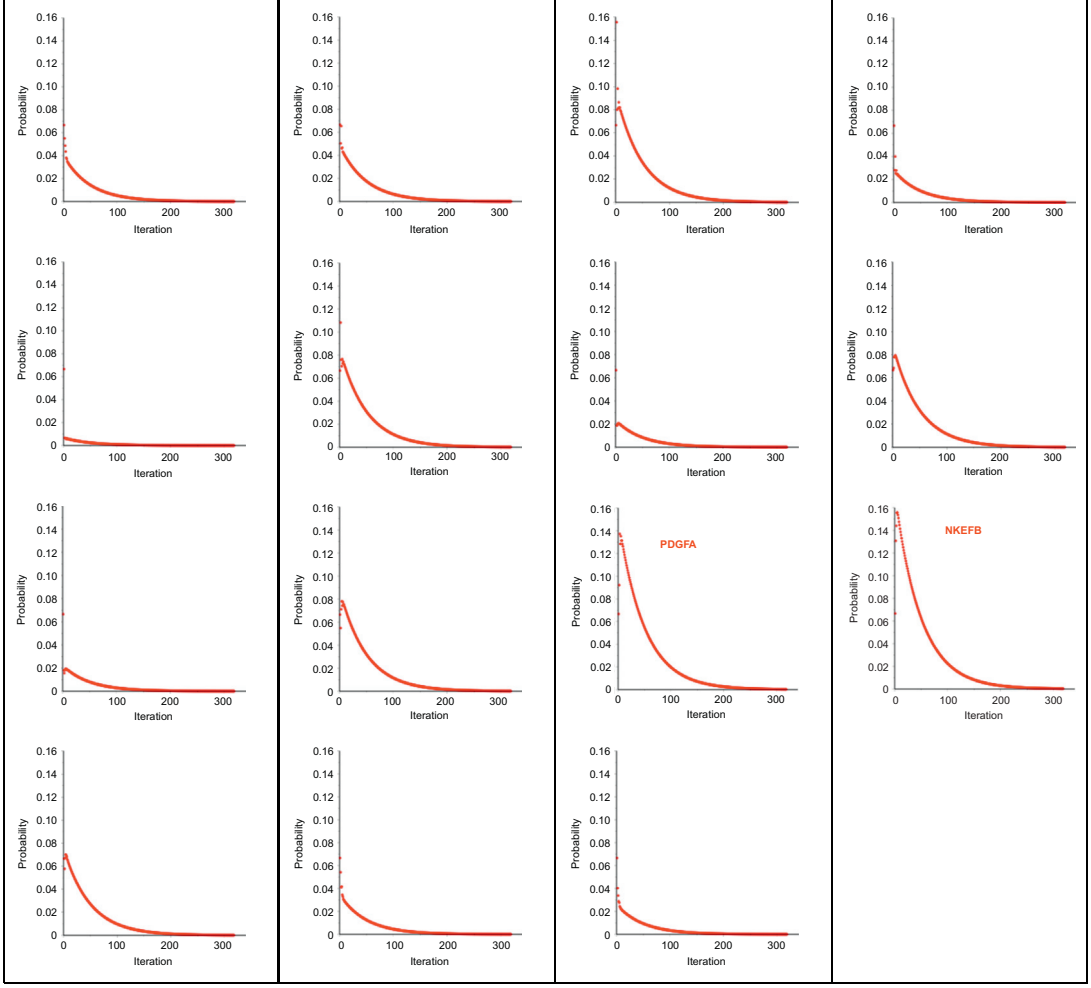

the formation of focal adhesions. Together with the fact that malignant gliomas express high levels of Ang1, biologists suggest the existence of an autocrine loop in malignant gliomas and that a Tie2-dependent pathway modulates cell-to-extracellular matrix adhesion. Furthermore, biologists also showed that Tie2 expression in the neoplastic glial cells was significantly associated with progression from a lower to the higher grade. Tie2 regulates glioma cell adhesion to the extracellular matrix, and the down-regulation of Tie2 levels by small interference RNA or the addition of soluble Tie2 abrogated the Ang1-mediated effect on cell adhesion. Our method provides a quantitative tool to verify these sensitivities of genes.

### 8.3.4 HI-IMPACT GENES

Similarly, suppose that initially the impact factor is uniformly distributed among the genes. The long-term impact factor converged and after normalizing we have a *converged impact vector* [0.230510562021945, 0.281733274014609, 0.554286767759576, 0.162393143989202, 0.0389022223245352, 0.496877449208407, 0.135282472604153, 0.513376834870215, 0.128454945353256, 0.512887379689246, 0.875556528547034, 0.999999999947150, 0.436019685843127, 0.200929165773528, 0.150127497384119] for genes Tie-2, TGFB3, ERCC1, HSP40, TDPX2, GSTP1, GNB1, NDKB, TOP2A, SCYB10, PDGFA, NKEFB, $\beta$-Actin, NKFB1, and BCL2A1, respectively.

From the converged impact vector, it shows that genes PDGFA and NKEFB are the highest impacting genes in a long run.Table 8.4 shows the similar pattern of these genes when the impact vector converges to its fixed point. This finding is in line with what we have learned from biologists,

**Table 8.4 Convergence of Impact Factors**



as [29] showed the family of platelet-derived growth factors (PDGFs) plays a number of critical roles in normal embryonic development, cellular differentiation, and response to tissue damage. As it is a multifaceted regulatory system, numerous pathological conditions are associated with aberrant activity of the PDGFs and their receptors. As it has been shown by [29], human gliomas, especially glioblastoma, express all PDGF ligands and both the two-cell surface receptors, PDGFR-$\alpha$ and -$\beta$. The cellular distribution of these proteins in tumors indicates that glial tumor cells are stimulated via PDGF/PDGFR-$\alpha$ autocrine and paracrine loops, while tumor vessels are stimulated via the PDGFR-$\beta$.

## 8.4 CONCLUSION

We presented an algebraic method for direct computation of the long-term influence and sensitivity of genes in a PBN. Our novel method only requires $O(n^2)$ memory space in contrast to other known methods in the literature that require the construction of the network's transition probability matrix with a huge size of $2^n \times 2^n$ where $n$ is the number of genes on the PBN. We are able to analyze the long-term behavior of genes in artificial PBNs with 500 genes within minutes on a desktop computer.

Our biological case study, using PBN from a human glioma gene expression data set, showed that the long-term influence and sensitivity of genes we found in these gliomas PBN are in line with what we have learned from biologists for a long time. Compared with the results from previous computational methods such as the steady state analysis using the two-state Markov chain approach, our results are much closer to what the biologists have known about this disease. Furthermore, our results also present some knowledge that was not known before and can be used to select genes of interest as an important step before mining heterogeneous large-scale genomic data. We hope that this new finding will help to direct more wet bench research on long-term influence and sensitivity of genes.

## ACKNOWLEDGMENTS

Supported in part by NSF through award CCF-1450146.

## REFERENCES

[1] Brun M, Dougherty E, Shmulevich I. Steady-state probabilities for attractors in probabilistic Boolean networks. Signal Process 2005;85(10):1993–2013. http://dx.doi.org/10.1016/j.sigpro.2005.02.016.

[2] Dougherty E, Shmulevich I. Mappings between probabilistic Boolean networks. Signal Process 2003;83 (4):799–809. http://dx.doi.org/10.1016/S0165-1684(02)00480-2.

[3] Gershenson C, Kauffman S, Shmulevich I. The role of redundancy in the robustness of random Boolean networks, corr abs/nlin/0511018. http://arxiv.org/abs/nlin/0511018.

[4] Hashimoto R, Kim S, Shmulevich I, Zhang W, Bittner M, Dougherty E. Growing genetic regulatory networks from seed genes. Bioinformatics 2004;20(8):1241–7. http://dx.doi.org/10.1093/bioinformatics/bth074.

[5] Hayashida M, Tamura T, Akutsu T, Zhang S, Ching WK. Algorithms and complexity analyses for control of singleton attractors in Boolean networks. EURASIP J Bioinform Syst Biol 2008;2008:521407. http://dx.doi.org/10.1155/2008/521407.

[6] Lähdesmäki H, Hautaniemi S, Shmulevich I, Yli-Harja O. Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. Signal Process 2006;86 (4):814–34. http://dx.doi.org/10.1016/j.sigpro.2005.06.008.

[7] Lähdesmäki H, Shmulevich I, Yli-Harja O. On learning gene regulatory networks under the Boolean network model. Mach Learn 2003;52(1–2):147–67.
[8] Liu W, Lähdesmäki H, Dougherty E, Shmulevich I. Inference of Boolean networks using sensitivity regularization. EURASIP J Bioinform Syst Biol 2008;2008:780541. http://dx.doi.org/10.1155/2008/780541.
[9] Shmulevich I, Dougherty E. Probabilistic Boolean Networks — The Modeling and Control of Gene Regulatory Networks. Philadelphia, PA: SIAM; 2010. http://www.ec-securehost.com/SIAM/OT118.html.
[10] Shmulevich I, Dougherty E, Kim S, Zhang W. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics 2002;18(2):261–74. http://dx.doi.org/10.1093/bioinformatics/18.2.261.
[11] Zhang S, Hayashida M, Akutsu T, Ching WK, Ng MP. Algorithms for finding small attractors in Boolean networks. EURASIP J Bioinform Syst Biol 2007;2007:20180. http://dx.doi.org/10.1155/2007/20180.
[12] Dubrova E, Teslenko M, Martinelli A. Kauffman networks: analysis and applications. In: ICCAD '05: proceedings of the 2005 IEEE/ACM international conference on computer-aided design. Washington, DC: IEEE; 2005. p. 479–84.
[13] Kauffman S. Metabolic stability and epigenisis in randomly constructed genetic nets. J Theor Biol 1969;22:437–67.
[14] Kauffman S. The origins of order: self-organization and selection in evolution. Oxford: Oxford University Press; 1993.
[15] Gartel A, Tyner A. The role of the cyclin-dependent kinase inhibitor p21 in apoptosis. Mol Cancer Res 2002;1(8):639–49.
[16] Grimaldi M, Visintainer R, Jurman G. RegnANN: reverse engineering gene networks using artificial neural networks. PLoS ONE 2011;6(12):e28646. http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3247226.
[17] Noman N, Iba H. Inferring gene regulatory networks using differential evolution with local search heuristics. IEEE/ACM Trans Comput Biol Bioinform 2007;4(4):634–47.
[18] Tamada Y, Bannai H, Imoto S, Katayama T, Kanehisa M, Miyano S. Utilizing evolutionary information and gene expression data for estimating gene networks with Bayesian network models. J Bioinform Comput Biol 2005;3(6):1295–314. http://dx.doi.org/10.1142/S0219720005001569.
[19] Tamada Y, Imoto S, Araki H, Nagasaki M, Print C, Charnock-Jones S, et al. Estimating genome-wide gene networks using nonparametric Bayesian network models on massively parallel computers. IEEE/ACM Trans Comput Biol Bioinform 2011;8(3):683–97. http://dx.doi.org/10.1109/TCBB.2010.68.
[20] Bochmann D, Posthoff C. Binäre Dynamische Systeme. München: R. Oldenbourg Verlag; 1981.
[21] Steinbach B, Posthoff C. Boolean differential equations, synthesis lectures on digital circuits and systems. San Rafael, CA: Morgan & Claypool Publishers; 2013.
[22] Kahn J, Kalai G, Linial N. The influence of variables on Boolean functions. In: FOCS: IEEE symposium on foundations of computer science (FOCS); 1988.
[23] Stone M. The theory of representation for Boolean algebras. Trans Am Math Soc 1936;40(1):37–111.
[24] Stone M. Applications of the theory of Boolean rings to general topology. Trans Am Math Soc 1937;41:375–481.
[25] Burris S, Sankappanavar H. A course in universal algebra. Berlin: Springer-Verlag; 1981.
[26] Fuller G, Rhee C, Hess K, Caskey L, Wang R, Bruner J, et al. Reactivation of insulin-like growth factor binding protein 2 expression in glioblastoma multiforme: a revelation by parallel gene expression profiling. Cancer Res 1999;59(17):4228–32.
[27] Brin S, Page L. The anatomy of a large-scale hypertextual web search engine. Comput Netw ISDN Syst 1998;30(1–7):107–17.
[28] Lee O, Xu J, Fueyo J, Fuller G, Aldape K, Alonso M, et al. Expression of the receptor tyrosine kinase Tie2 in neoplastic glial cells is associated with integrin beta1-dependent adhesion to the extracellular matrix. Mol Cancer Res 2006;4(12):915–26.
[29] Nazarenko I, Hede SM, He X, Hedrén A, Thompson J, Lindstroem M, et al. PDGF and PDGF receptors in glioma. Ups J Med Sci 2012;117(2):99–112.