BIOINFORMATICS ORIGINAL PAPER Vol. 23 no. 12 2007, pages 1511–1518
doi:10.1093/bioinformatics/btm142

Gene expression

An approximation method for solving the steady-state probability distribution of probabilistic Boolean networks

Wai-Ki Ching[1], Shuqin Zhang[1,*], Michael K. Ng[2] and Tatsuya Akutsu[3]

[1]Advanced Modeling and Applied Computing Laboratory, Department of Mathematics, The University of Hong Kong, Pokfulam Road, Hong Kong, [2]Department of Mathematics, Hong Kong Baptist University, Kowloon Tong, Hong Kong and [3]Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji-city, Kyoto 611-0011, Japan

Received on September 25, 2006; revised and accepted on April 6, 2007

Advance Access publication April 26, 2007

Associate Editor: Trey Ideker

ABSTRACT

Motivation: Probabilistic Boolean networks (PBNs) have been proposed to model genetic regulatory interactions. The steady-state probability distribution of a PBN gives important information about the captured genetic network. The computation of the steady-state probability distribution usually includes construction of the transition probability matrix and computation of the steady-state probability distribution. The size of the transition probability matrix is 2^n-by-2^n where n is the number of genes in the genetic network. Therefore, the computational costs of these two steps are very expensive and it is essential to develop a fast approximation method.

Results: In this article, we propose an approximation method for computing the steady-state probability distribution of a PBN based on neglecting some Boolean networks (BNs) with very small probabilities during the construction of the transition probability matrix. An error analysis of this approximation method is given and theoretical result on the distribution of BNs in a PBN with at most two Boolean functions for one gene is also presented. These give a foundation and support for the approximation method. Numerical experiments based on a genetic network are given to demonstrate the efficiency of the proposed method.

Contact: sqzhang@hkusua.hku.hk

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

1 INTRODUCTION

Mathematical modeling and computational study of regulatory interactions between DNA, RNA, proteins and small molecules based on the microarray data are hot topics in bioinformatics and have been studied by a number of researchers (Celis *et al*., 2000; Hughes *et al*., 2001; Lipshutz *et al*., 1999; Lockhart and Winzeler, 2000; Schena *et al*., 1995). There have been many formalisms proposed in the literatures to study the genetic regulatory networks such as directed graphs, Bayesian networks, Boolean networks (BNs), probabilistic Boolean networks (PBNs), ordinary and partial differential equations and many other mathematical models (Jong, 2002). Among these models, BN and PBN (an extension from BN) attract much attention in the biophysics community. Reviews of BN models can be found in Huang (1999), Kauffman (1993) and Somogyi and Sniegoski (1996). In a BN model, gene expression states are quantified into two levels: on and off (represented as 1 and 0, respectively). Even though most biological phenomena manifest them in continuous domain, the binary expression can capture the qualitative relationships, and show promising and useful results (Shmulevich and Zhang, 2002; Szallasi and Liang, 1998; Wuensche, 1998). In a BN model, the target gene is predicted by several genes called its input genes via a Boolean function. Once the input genes and the Boolean functions are known, the BN model is constructed deterministically.

However, genetic regulation exhibits uncertainty in the biological level and microarray data for inferring the model may have errors due to experimental noise in the complex measurement processes. Thus, a deterministic model is not favorable to such real situations and to develop a model incorporating the uncertainty is needed, which results in the development of PBNs. PBNs have been recently developed and studied in the literatures. In a PBN, for each gene, there can be more than one Boolean function. The state transits into a number of states with certain probabilities according to the realizations of all possible BNs. Thus, the dynamics (transitions) of the system can be described by a Markov chain. Detailed explanations of extending BN to the instantaneously random PBN can be found in Shmulevich *et al*. (2002a). The random gene perturbations are introduced into the PBN model in the paper (Shmulevich *et al*., 2002b), where the perturbation describes the random inputs from the outside. Introducing the random gene perturbation into the system makes it stable in the long run. Further extension of PBN model to the context-sensitive PBN model was introduced by Pal *et al*. (2005). A brief review of the PBN model based on mathematical formulation will be given in the next section.

Given a PBN, a natural and important problem is to study its steady-state probability distribution (Brun *et al*., 2005, Shmulevich *et al*., 2003). It provides the first-order statistical information of a PBN. Based on such information of a PBN, one can understand a genetic network, and identify the influence of different genes in a network. Furthermore, one can figure out how to control some genes in a network, such

*To whom correspondence should be addressed.

that the whole system can evolve into a target state or desired steady-state probability distribution. Therapeutic gene intervention or gene control policy (Datta *et al*., 2003, 2004; Ng *et al*., 2006; Shmulevich *et al*., 2002b) can then be developed and studied.

Monte-Carlo simulation method has been proposed in Shmulevich *et al*. (2003) to estimate the steady-state probability distribution of a PBN. The idea is that, by simulating the underlying Markov chain for a sufficiently long time until it converges, one can get an approximation of the steady-state probability distribution. Although it has been shown that Monte-Carlo simulation method can perform well in a small PBN, it can be successfully used only if we are sufficiently confident that the system has evolved into its steady state before the algorithm stops. Theoretically, a priori bound on the number of iterations is too large to be useful even for a moderate size network (Rosenthal, 1995). Thus in practice, only empirical determination methods can be used to stop the chain and get an estimate of the steady-state probability distribution (Shmulevich *et al*., 2003). On the other hand, matrix-based method, a deterministic method can be used to obtain the steady-state probability distribution more accurately and efficiently.

It is well known that in Markov chain theory, if a Markov chain is irreducible and aperiodic, the steady-state probability distribution is independent of the initial condition. We remark that in a PBN with random gene perturbations (Shmulevich *et al*., 2002b), the underlying transition probability matrix can be shown to be irreducible and aperiodic. In Zhang *et al*. (2007), power method has been successfully applied to compute the steady-state probability distribution based on an efficient construction of the transition probability matrix of a PBN without random perturbation. The complexity of the construction of probability transition matrix is $O(n \cdot N \cdot 2^n)$, where N is the total number of BNs and n is the number of genes.

The main aim of this article is to propose an efficient and effective approximation method, based on the method presented in Zhang *et al*. (2007) to find the steady-state probability distribution for the general PBN model. The rest of the article is organized as follows. In Section 2.1, a brief review on the mathematical formulation of the PBN model is given. In Section 2.2, the methodology to compute the steady-state probability is introduced with an error analysis. In Section 3.1, numerical experiments are given to demonstrate the efficiency and effectiveness of the proposed method. In Section 3.2, we give the probability distribution of BNs in a PBN with at most two Boolean functions for one gene. Finally, in Section 4, we give a brief summary to conclude the article.

2 METHODS

2.1 A review of probabilistic Boolean networks

In this section, we give a brief review of the PBNs. A PBN is the generalization of a BN. A BN $G(V, F)$ consists of a set of nodes V and Boolean functions F where

$$V = \{v_1, v_2, \ldots, v_n\} \quad \text{and} \quad F = \{f_1, f_2, \ldots, f_n\}.$$

Let $v_i(t)$ be the state of v_i at time t, where $v_i = 0$ represents that the gene is unexpressed and $v_i = 1$ means it is expressed. The overall expression levels of all the genes in the network at time step t is given by the following column vector

$$v(t) = [v_1(t), v_2(t), \ldots, v_n(t)]^T.$$

This vector is called the gene activity profile (GAP) of the network at time t. We note that when $v(t)$ ranges from $[0, 0, \ldots, 0]^T$ (all entries are 0) to $[1, 1, \ldots, 1]^T$ (all entries are 1), it takes on all the 2^n possible states of the n genes. The list of Boolean functions represents the rules of the regulatory interactions among the nodes (genes):

$$v_i(t+1) = f_i(v(t)), \quad i = 1, 2, \ldots, n.$$

Here, each gene will update its state according to the states of its input genes in the previous step and its corresponding Boolean function. Thus, a BN is a deterministic dynamical system.

In a PBN, for each target gene, instead of having only one single Boolean function, it has a number of Boolean functions having equivalent prediction abilities. All these Boolean functions can be selected randomly with some probabilities. We assume that for the ith gene, there are $l(i)$ possible Boolean functions:

$$F^{(i)} = \left\{ f_j^{(i)} : \text{ for } j = 1, \ldots, l(i) \right\}$$

and the probability of choosing function $f_j^{(i)}$ is $c_j^{(i)}$, where $f_j^{(i)}$ is a function with respect to the activity levels of n genes. A PBN is said to be independent if the elements from different $F^{(i)}$ are independent (Lähdesmäki *et al*., 2006). For an independent PBN of n genes, there are at most

$$N = \prod_{i=1}^{n} l(i) \tag{1}$$

different possible BNs. This means that there are totally N possible realizations of the genetic network. Suppose f_j be the j-th possible realization,

$$f_j = [f_{j_1}^{(1)}, f_{j_2}^{(2)}, \ldots, f_{j_n}^{(n)}], \quad 1 \le j_i \le l(i), \quad i = 1, 2, \ldots, n,$$

The probability to choose the j-th realization is:

$$P_j = \prod_{i=1}^{n} c_{j_i}^{(i)}, \quad j = 1, 2, \ldots, N. \tag{2}$$

If the joint probability distribution of $F^{(1)}, F^{(2)}, \ldots, F^{(n)}$ cannot be factorized as the product of $F^{(i)}$, then it is a dependent PBN. For a dependent PBN, we use the same notations as those for independent PBNs. But notice that now the expressions of N and P_j will be different from (1) and (2).

Let a and b be any two column vectors with n entries being either 0 or 1, which represent the states of the system at time $t+1$ and t. Then

$$\text{Prob}\{v(t+1) = a \mid v(t) = b\} = \sum_{j=1}^{N} \text{Prob}\{v(t+1) = a \mid v(t) = b, \text{ the } j\text{-th BN is selected}\} P_j.$$

By letting a and b ranging from $[0, 0, \ldots, 0]^T$ to $[1, 1, \ldots, 1]^T$ independently, one can get the transition probability matrix A with size $2^n \times 2^n$. It can be expressed as:

$$A = \sum_{j=1}^{N} P_j A_j$$

where A_j is the transition matrix corresponding to the j-th BN.

Random gene perturbation is the description of the random inputs from the outside due to external stimuli and this is meaningful in an open genome system. The effect of the random gene perturbation is to make the genes flip from state 1 to state 0 or vice versa. It makes the underlying Markov chain of the PBN ergodic and therefore all the 2^n states in the system are communicated (Shmulevich *et al*., 2002b).

When random gene perturbation is included, the transition probability matrix $\tilde{A}$ is

$$\tilde{A}(i,j) = (1-p)^n A(i,j) + \underbrace{p^{h(v(i),v(j))}(1-p)^{n-h(v(i),v(j))} I_{v(i)\neq v(j)}}_{\text{perturbation part}}. \quad (3)$$

Here $h(v(i), v(j))$ is Hamming distance between the two vectors $v(i)$ and $v(j)$, p is the perturbation probability of each gene and $I_{v(i)\neq v(j)}$ is the indicator function.

The instantaneously random PBN was extended to the context-sensitive PBN in Pal *et al*. (2005). In a context-sensitive PBN, at each time step the BN will be changed with a probability q. The transition probability matrix A without the gene perturbations can be obtained from the following:

$$\begin{aligned}
&\text{Prob}\left\{v(t+1) = a \mid v(t) = b\right\} \\
&= \sum_{j=1}^{N} \text{Prob}\left\{v(t+1) = a \mid v(t) = b,\ b \text{ is in the } j\text{-th BN}\right\} P_j \\
&= \sum_{j=1}^{N}\sum_{l=1}^{N} \text{Prob}\{v(t+1) = a + 2^n(l-1) | \\
&\quad v(t) = b + 2^n(j-1)\} P_j.
\end{aligned}$$

Here the probability

$$\text{Prob}\{v(t+1) = a + 2^n(l-1) | v(t) = b + 2^n(j-1)\}$$

describes the chance that state b will make a transition into state a in network l when b belongs to network j. The difference between a context-sensitive PBN and an instantaneously random PBN is that the column vectors a and b are assumed to be in a certain BN with some probability at each time point and it will change to other BNs with a probability q. Similar to the instantaneously random PBN, when the column vectors a and b run from state $[0, 0, \ldots, 0]^T$ to $[1, 1, \ldots, 1]^T$, we can get the transition matrix A. It can also be described as:

$$\begin{aligned}
A &= \sum_{j=1}^{N}\left((1-q)P_j A_j + \sum_{k\neq j} q P_j \frac{P_k}{\sum_{l\neq j} P_l} A_k\right) \\
&= \sum_{j=1}^{N} P_j \left((1-q) + q\sum_{k\neq j}\frac{P_k}{\sum_{l\neq k} P_l}\right) A_j. \quad (4)
\end{aligned}$$

The random gene perturbations can be introduced into a context-sensitive PBN similarly.

2.2 Computation of the steady-state probability distribution

In this section, a computational method for approximating the steady-state probability distribution is introduced. The computational method consists of two steps: (i) the construction of the transition probability matrix of the PBN without perturbation and the construction of the perturbation matrix, with which we can get the final transition matrix; (ii) computing the eigenvector corresponding to the maximum eigenvalue. The eigenvector in the normalized form is the steady-state probability vector. From Equation (3), we observe that the final transition matrix $\tilde{A}$ is the sum of the transition matrix without perturbation A multiplied by $(1-p)^n$ and the perturbation matrix. The perturbation matrix is same for different networks since it only depends on the number of genes and the random gene perturbation probability. Although the construction of this matrix may cost much time, once the matrix is constructed, it can be used later for all the networks with same number of genes and same perturbation probability. When there is no perturbation, the transition matrix is sparse, while it is dense if there is perturbation. In the third step, the power method, an efficient and widely used method, is applied to solve the dominant eigenvalue and the corresponding eigenvector. We remark that in our case, the dominant eigenvector is actually the steady-state probability vector. In our numerical tests, the computation of the eigenvector can be finished within one minute. When there are 12 genes, it cost ~5 s and when there are 14 genes, it cost ~13 s only. Now our main aim is to reduce the computational cost for construction of the transition matrix A for both the instantaneously random PBN and the context-sensitive PBN without gene perturbations.

In Shmulevich *et al*. (2002c), the transition probability matrix A is constructed by computing all the entries one by one. For each entry, all the BNs have to be considered so as to determine if the network contributes to it or not. Take for example, if one wants to compute the entry $A(j, i)$, one needs to consider if the first BN is applied, whether state j can be visited by i. And then consider the second network, and so on. Since the matrix A is large and sparse in practice, much time was wasted in computing the zero entries. In Zhang *et al*. (2007), an efficient algorithm has been proposed to construct the transition probability matrix. The idea is to consider the non-zero entries only. The method is based on the state-space of the Markov chain. Given a state i, if a specific Boolean function can lead it to state j, then $A(j, i)$ will have value corresponding to the probability of this BN. If another BN also can lead state i to state j, then the probability will be greater by the probability corresponding to the BN. Although this is only an improvement in computing the transition probability matrix, it already saves much time and makes significant progress in computing the steady-state probability. We remark that the computational complexity is $O(n \cdot N \cdot 2^{2n})$ for method proposed in Shmulevich *et al*. (2002c), while it is $O(n \cdot N \cdot 2^n)$ in Zhang *et al*. (2007) where n is the number of genes and N is the number of BNs in the PBN. Moreover if there are m states for each gene, the computational complexity will change from $O(n \cdot N \cdot m^{2n})$ to $O(n \cdot N \cdot m^n)$.

We observe that in many realizations of a PBN, a lot of BNs have slim chances to be chosen. Therefore our proposed method here is to consider only those BNs with probability greater than a given threshold. This of course can save much time in the construction of the transition probability matrix. Since the idea here only involves the probabilities of choosing the BNs, it can be extended to the PBNs having multiple values. Moreover, it can be applied to both the dependent and independent PBNs. In the next section, we will show some numerical experiments for the proposed method. The following is a simple explanation of the error due to removal of the BNs.

Suppose the steady-state probability vector of $\tilde{A}x = x$ is X. There are n_0 BNs being removed whose corresponding transition matrix are $(A_1, A_2, \ldots, A_{n_0})$ and their probability of being chosen are given by $p_1, p_2, \ldots, p_{n_0}$, respectively. Then after the removal of these n_0 BNs and making a normalization, the transition probability matrix becomes

$$\begin{aligned}
\hat{A} &= (1-p)^n \times \frac{1}{1-(p_1+p_2+\ldots+p_{n_0})} \\
&\quad \times (A - (p_1A_1 + p_2A_2 + \ldots + p_{n_0}A_{n_0})) + \tilde{P}. \quad (5)
\end{aligned}$$

Here $\tilde{P}$ is the perturbation matrix. Suppose that the steady-state probability vector for the linear system $\hat{A}x = x$ is $\hat{X}$, then from (5), we have

$$\begin{aligned}
&((1-p)^n(A - p_1A_1 - p_2A_2 - \cdots - p_{n_0}A_{n_0}) \\
&\quad + (1-(p_1+p_2+\cdots+p_{n_0}))\tilde{P})\hat{X} \\
&= (1-(p_1+p_2+\cdots p_{n_0}))\hat{X}. \quad (6)
\end{aligned}$$

Therefore we have,

$$\begin{aligned}
&\|\tilde{A}\hat{X}-\hat{X}\|_\infty \\
&\quad=\|((1-p)^n A+\tilde{P})\hat{X}-\hat{X}\|_\infty \\
&\quad=\|(1-p)^n(p_1A_1+p_2A_2+\cdots+p_{n_0}A_{n_0})\hat{X} \\
&\quad+(p_1+p_2+\cdots+p_{n_0})\tilde{P}\hat{X}-(p_1+p_2+\cdots p_{n_0})\hat{X}\|_\infty \\
&\quad=\|p_1((1-p)^nA_1+\tilde{P}-I)\hat{X}+p_2((1-p)^nA_2+\tilde{P}-I)\hat{X} \\
&\quad+\cdots+p_{n_0}((1-p)^nA_{n_0}+\tilde{P}-I)\hat{X}\|_\infty \\
&\quad\le p_1\|((1-p)^nA_1+\tilde{P}-I)\hat{X}\|_\infty \\
&\quad+p_2\|((1-p)^nA_2+\tilde{P}-I)\hat{X}\|_\infty \\
&\quad+\ldots+p_{n_0}\|((1-p)^nA_{n_0}+\tilde{P}-I)\hat{X}\|_\infty \qquad (7)
\end{aligned}$$

We note that in each column of $A_k(k=1,2,\ldots,n_0)$, there is one non-zero entry and it is equal to one. We do not know the exact form of each A_k. Here, we assume that the position of the non-zero entry follows the uniform distribution. Let

$$y_i=\sum_{j=1}^{2^n}[(1-p)^nA_k+\tilde{P}]_{ij},\ Y=\max\{y_1,y_2,\ldots,y_{2^n}\}.$$

Since the term $(1-p)^nA_k+\tilde{P}$ is the transition probability matrix with perturbation corresponding to the k-th BN, $E(y_i)=1$. We remark that the Chernoff bound in Motwani and Rahavan (1995, p. 72) states that:

Let $Z_1,Z_2,\ldots,Z_m$ be independent Poisson trials such that for $i=1,\ldots,m$, $\text{Prob}(Z_i)=p_i$ where $0<p_i<1$. Then, for

$$Z=\sum_{i=1}^m Z_i,\qquad \mu=E(Z)=\sum_{i=1}^m p_i$$

and $\delta>2e-1$,

$$\text{Prob}(Z>(1+\delta)\mu)<2^{-(1+\delta)\mu}.$$

By letting $m=2^n$, $\delta=2n$ and $\mu=1$, we can have

$$\begin{aligned}
&\text{Prob}(y_i>(1+2n))<\frac{1}{2^{1+2n}}, \\
&\text{Prob}(Y>(1+2n))<2^n\times\frac{1}{2^{1+2n}}=\frac{1}{2^{1+n}}.
\end{aligned}$$

We should note that n should be larger than 2 since $n>(2e-1)/2$ is assumed (e is the base of natural log). However, if $n\le 2$, $Y\le 1+2n$ always holds.

Thus, we have

$$E(Y)<\max(1+2n,2^n\times\frac{1}{2^{1+n}})=1+2n,$$

$$\begin{aligned}
E\Big(\|((1-p)^nA_k+\tilde{P})\hat{X}-\hat{X}\|_\infty\Big)&<(1+E(Y))\|\hat{X}\|_\infty \\
&<(2+2n)\|\hat{X}\|_\infty.
\end{aligned}$$

Finally, from (7) and the above, the expected residual is bounded by

$$E\Big(\|\tilde{A}\hat{X}-\hat{X}\|_\infty\Big)<(p_1+p_2+\cdots+p_{n_0})(2+2n)\|\hat{X}\|_\infty.$$

If $\|X\|_\infty$ is equal to or very close to $\|\hat{X}\|_\infty$, we can see

$$E\left(\frac{\|\tilde{A}\hat{X}-\hat{X}\|_\infty}{\|X\|_\infty}\right)<(p_1+p_2+\cdots+p_{n_0})(2+2n).$$

Since this error estimate only gives an expected upper error bound, it cannot be applied for all the cases to estimate n_0. From the analysis, we can see generally, the error bound can be determined by total probability of the removed BNs and the number of genes in the PBN.

Take for example, if $n=10$, the total probability of all the removed BNs is 0.01 (the remaining networks is 0.99), and $\|\hat{X}\|_\infty<0.05$. Then the expected residual norm $\|\tilde{A}\hat{X}-\hat{X}\|_\infty$ of the new steady-state probability vector is bounded by

$$22(p_1+p_2+\ldots+p_{n_0})||\hat{X}||_\infty<0.011.$$

We remark that for the context-sensitive PBN, from (4), we can see

$$P_j((1-q)+q\sum_{k\neq j}\frac{P_k}{\sum_{l\neq k}P_l})$$

corresponds to P_j in the instantaneously random PBN. With the same method, the error can be estimated.

3 RESULTS

3.1 Numerical experiments

In this section, we present some numerical experiments based on a network described in Shmulevich *et al.* (2003). Gene (TOP2A) is the only input gene of gene (SCYB10); (INP10); (IP10) and the in-degree of it is zero, then it is not considered in the tests. Thus, the total number of genes studied in the network is 14 and the total number of possible states is equal to 2^{14}, i.e. 16 384. For simplicity of discussion, here we consider independent PBNs only. The settings of the experiments are the same as those in Zhang *et al.* (2007). The number of Boolean functions of each gene is given by

2 2 2 1 1 3 1 2 2 2 2 1 2 2

and the probabilities for choosing the Boolean functions is shown in Table 1. In the numerical experiment, the number of input genes in a BN is set to be no more than three and the input genes are randomly selected. The Boolean functions are also generated randomly.

All the numerical experiments are done in a notebook computer with the following configuration: Intel Pentium M 1.5 GHz and RAM: 768 MB. Figure 1 presents the distribution of all the 1536 BNs. We observe that a relatively small number of BNs with high probability constitute most of the total probability. Table 2 shows the experimental results for the instantaneously random PBNs and Table 3 shows the experimental results for the context-sensitive PBNs. The perturbation probability for each gene is set to be 0.01. In the results of the

Table 1. Selection probability of all the Boolean functions in the network with 14 genes

f^1	f^2	f^3	f^4	f^5	f^6	f^7
0.8560	0.2768	0.6759	1.0000	1.0000	0.0264	1.0000
0.1440	0.7232	0.3241			0.4983	
					0.4754	
f^8	f^9	f^{10}	f^{11}	f^{12}	f^{13}	f^{14}
0.0857	0.5595	0.0751	0.8508	1.0000	0.8697	0.6004
0.9143	0.4405	0.9249	0.1492		0.1303	0.3996

The column corresponding to each f^i is the probability of choosing the corresponding Boolean function f^i_j, where j is the row number of the probabilities in the table.

context-sensitive model, the probability of the transition from one network to others is set to be 0.01. We set a lower bound of all P_j. If P_j is less than the lower bound, the corresponding network is not considered in the construction of the transition probability matrix. We compute the error between the true steady-state probability distribution and the approximate one with the following vector norms: $\|\cdot\|_1$, $\|\cdot\|_2$ and $\|\cdot\|_\infty$ and compare them with the infinity norm of the true steady-state probability distribution by dividing the errors by it. We remark that the error does not depend on the perturbation probability much. For example, the proportion between the relative error when $p = 0.01$ and $p = 0.001$ is between 0.97 and 1.03. The total number of BNs removed is denoted by n_0. We also give the computational time of constructing the transition matrix without perturbation. From the results in the tables, it is easy to see that the computational time is linear with respect to the number of BNs. The first 1000 states with largest probability are chosen and compared with those in the true case. We find that even though the lower bound is set to be 10^{-4}, almost all of the 1000 states are from those in the true case (991 out of 1000). The total probability of these 1000 states is about 0.8 in this test case. The number of matched states is denoted by n_{match} in the table.

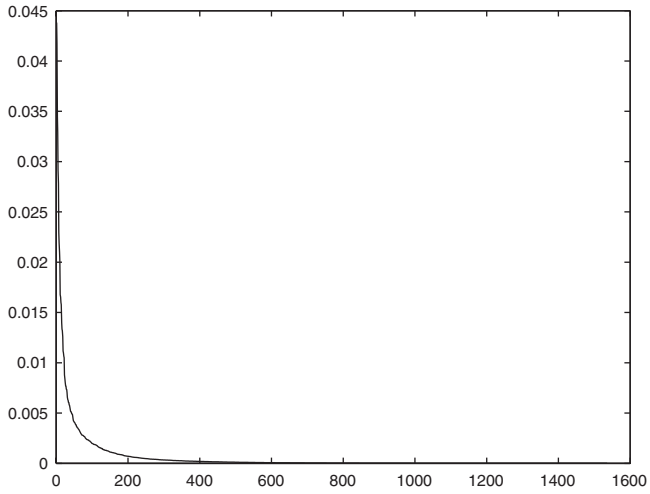

Fig. 1. The distribution of all the BNs in a network taken from Shmulevich *et al*. (2003) with total 1536 BNs.

3.2 Probability distribution of the Boolean networks

Since the number of input genes of one gene cannot be very large (Guelzim *et al*., 2002), the number of Boolean functions should be very small. For the independent PBN, if the maximum number of Boolean functions for all the target genes is two, and the probability of choosing one Boolean function follows the uniform distribution, then the number of BNs dropped given a threshold will follow some interesting probability distribution.

We first consider the case that for any gene i the probability that the first Boolean function is chosen is given by c. This means that

$$c_1^{(i)} = c \quad \text{and} \quad c_2^{(i)} = 1 - c.$$

We may further assume $c > 0.5$. In the case of $c = 0.5$, all BNs have the same probability 2^{-n} of being chosen. Thus given a threshold, either all the BNs are removed or all are retained. If the threshold level is given by β, then the condition that a BN will be removed is given by

$$c^k(1-c)^{n-k} < \beta$$

Table 2. Numerical results of the approximation method for the network using the instantaneously random PBN model

Lower bound of all P_j	$Error_{\|\cdot\|_1}$	$Error_{\|\cdot\|_2}$	$Error_{\|\cdot\|_\infty}$	n_0	Time (s)	n_{match} out of 1000
10^{-4}	1.68×10^{0}	7.03×10^{-2}	2.61×10^{-2}	1009	1369	991
10^{-5}	1.66×10^{-1}	6.60×10^{-3}	2.30×10^{-3}	535	2459	997
10^{-6}	9.00×10^{-3}	3.47×10^{-4}	1.15×10^{-4}	182	3257	1000

The BNs with probabilities less than the lower bound of all P_j are removed during the construction process of the probability transition matrix. Three different vector norms $\|\cdot\|_1, \|\cdot\|_2, \|\cdot\|_\infty$ are used to evaluate the relative errors to the maximum steady-state probability of the original system(divided by it). Here, n_0 is the total number of the BNs removed and there are n_{match} states remaining in the 1000 states with highest probability of the original system after taking the approximation. The random gene perturbation probability is 0.01. The total computational time for the construction of the true transition probability matrix without gene perturbation is 3727 s.

Table 3. Numerical results of the approximation method for the network using the context-sensitive PBN model

Lower bound of all P_j	$Error_{\|\cdot\|_1}$	$Error_{\|\cdot\|_2}$	$Error_{\|\cdot\|_\infty}$	n_0	Time (s)	n_{match} out of 1000
10^{-4}	1.68×10^{0}	7.04×10^{-2}	2.61×10^{-2}	1009	1380	991
10^{-5}	1.66×10^{-1}	6.60×10^{-3}	2.30×10^{-3}	535	2493	997
10^{-6}	9.00×10^{-3}	3.48×10^{-4}	1.15×10^{-4}	182	3333	1000

Explanations of the table is similar to that for Table 2. The transition probability of one BN to other BNs is 0.01. The random gene perturbation probability is 0.01. The total computational cost for the construction of the true transition probability matrix without perturbation is 3782 s.

Table 4. Selection probability of the Boolean functions in the simulated network

f^1	f^2	f^3	f^4	f^5	f^6
0.7038	0.0093	0.0199	0.3280	0.8615	0.3421
0.2962	0.9907	0.9801	0.6720	0.1385	0.6579
f^7	f^8	f^9	f^{10}	f^{11}	f^{12}
0.0759	0.0331	0.9006	0.8244	0.6970	0.3603
0.9241	0.9669	0.0994	0.1756	0.3030	0.6397

The column corresponding to each f^i is the probability of choosing the corresponding Boolean function f^i_j, where j is the row number of the probabilities in the table.

or

$$k > \frac{\log\beta - n\log(1-c)}{\log c - \log(1-c)}$$

Here, k is the number of first Boolean functions being chosen in the BNs. Let k^* be the minimum integer such that the above inequality holds. Then the number of BNs being removed will be given by

$$\sum_{j=k^*}^{n} \frac{n!}{j!(n-j)!}.$$

Therefore, the proportion of BNs being removed is given by

$$\frac{1}{2^n}\sum_{j=k^*}^{n} \frac{n!}{j!(n-j)!}.$$

We then consider the following. We derive the probability that a randomly chosen BN will be removed during the construction of the transition matrix in the approximate computation given that for any gene i, the probability of its first Boolean function being chosen follows the uniform distribution $U(0, 1)$. We begin with the following lemma.

LEMMA 1. (Ross, 1997) *If u follows the uniform distribution $U[0, 1]$ then $1-u$ also follows the uniform distribution and the random variables*

$$\frac{-1}{\lambda}\ln(u) \qquad \text{and} \qquad \frac{-1}{\lambda}\ln(1-u)$$

follow the exponential distribution $\lambda e^{-\lambda x}$.

LEMMA 2. *If $X_1, X_2, \ldots, X_m$ are independent and identically distributed, and follow the exponential distribution with $\lambda = 1$, then*

$$\xi = X_1 + X_2 + \cdots + X_m$$

has the following Erlangian distribution of m phases

$$\text{Prob}(\xi < y) = 1 - \sum_{k=0}^{m-1} \frac{y^k}{k!} e^{-y}$$

Proof: We will prove this by using mathematical induction. When $\xi = X_1$, $\text{Prob}(\xi < y) = 1 - e^{-y}$, the statement holds. Suppose that for $\xi = X_1 + X_2 + \cdots + X_m$, we have

$$\text{Prob}(\xi < y) = 1 - \sum_{k=0}^{m-1} \frac{y^k}{k!} e^{-y}.$$

Then

$$\begin{aligned}
&\text{Prob}(X_1 + X_2 + \cdots + X_m + X_{m+1} < y) \\
&= \int_0^y e^{-X_{m+1}} \text{Prob}(X_1 + X_2 \cdots + X_m < y - X_{m+1}) dX_{m+1} \\
&= \int_0^y e^{-X_{m+1}} \left(1 - \sum_{k=0}^{m-1} \frac{(y - X_{m+1})^k}{k!} e^{-(y - X_{m+1})}\right) dX_{m+1} \\
&= \int_0^y e^{-X_{m+1}} dX_{m+1} - \int_0^y \sum_{k=0}^{m-1} \frac{(y - X_{m+1})^k}{k!} e^{-y} dX_{m+1} \\
&= 1 - e^{-y} - \sum_{k=1}^{m} \frac{y^k}{k!} e^{-y} = 1 - \sum_{k=0}^{m} \frac{y^k}{k!} e^{-y}.
\end{aligned}$$

Here the proposition follows.

PROPOSITION 1. *Given that the threshold level is β and there are n genes, the probability distribution function of all the BNs is given by*

$$f(\beta) = \frac{(-\ln\beta)^{n-1}}{(n-1)!} \quad \text{for} \quad \beta \in (0, 1).$$

Proof: Assume there are n genes and for each gene i there correspondingly $l_i = 2$ Boolean functions: $f_1^{(i)}, f_2^{(i)}, i = 1, 2, \ldots, n$. Let the probability of choosing $f_j^{(i)}$ be $c_j^{(i)}$, $c_j^{(i)} \sim U[0, 1]$. Suppose that P_j is the probability of choosing the j-th BN,

$$P_j = \prod_{i=1}^{n} c_{j_i}^{(i)}, \tag{8}$$

Given a constant β,

$$\begin{aligned}
\text{Prob}(P_j < \beta) &= \text{Prob}\left(\prod_{i=1}^{n} c_{j_i}^{(i)} < \beta\right) \\
&= \text{Prob}\left(-\sum_{i=1}^{n} \ln c_{j_i}^{(i)} > -\ln\beta\right) \\
&= 1 - \text{Prob}\left(-\sum_{i=1}^{n} \ln c_{j_i}^{(i)} \le -\ln\beta\right).
\end{aligned}$$

Since $c_j^{(i)} \sim U[0, 1]$, $-\ln c_j^{(i)}$ follows the exponential distribution. According to the above lemmas,

$$\text{Prob}(P_j < \beta) = \sum_{j=0}^{n-1} \frac{\beta(-\ln\beta)^j}{j!}. \tag{9}$$

Then by differentiating (9) with respect to β, the probability density function is got as

$$f(\beta) = \frac{(-\ln\beta)^{(n-1)}}{(n-1)!} \qquad \text{for} \quad \beta \in (0, 1).$$

We use a simple randomly generated network to illustrate the results. There are 12 genes in the network, each gene has 2 Boolean functions, and therefore there are totally 4096 BNs. For each gene there are 4 input genes, which are the input of one of the two Boolean functions. Figure 2 shows the distribution of all BNs. Figure 3 gives a close picture to the distribution of the first 500 BNs with highest probability. Table 4 shows the selection probabilities of all the Boolean

functions. Tables 5 and 6 show the details of this experiment for both the instantaneously random PBN model and the context- sensitive PBN model. The notations are same as those in the previous section. The perturbation probability is 0.03 here. In the context-sensitive PBN, the transition probability of a BN to other BNs is set to be 0.5. From the errors, it can be seen that the approximate method can give a reasonable explanation of the original system after dropping some BNs. The total probability of the first 500 states with highest probability is about 0.88. We note that almost all of these states appear in the approximate solution depending on the requirement of the error. The computational time for constructing the transition matrix are recorded in the tables. It decreased much in our approximation method depending on the requirement of the error tolerance.

4 CONCLUSION

In this article, we presented a matrix-based approximation method for computing the steady-state probability distribution of PBNs. This method works in the construction of the transition probability matrix, which is of complexity $O(nN2^n)$ in the original system, where N is the total number of BNs and n is the number of genes. In our method, N can be smaller by neglecting some BNs with little probability based on the

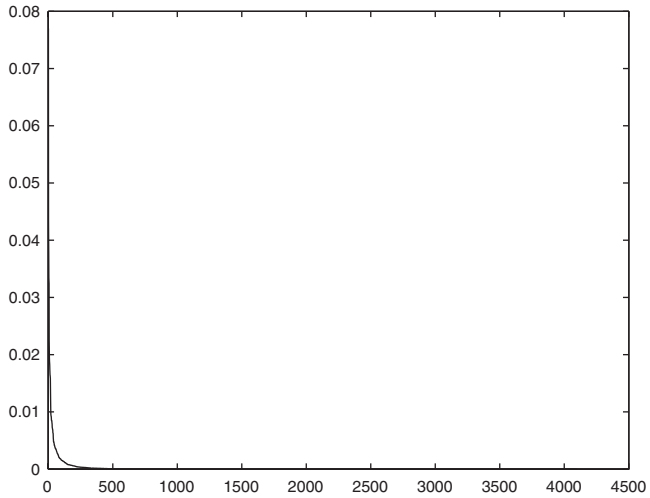

Fig. 2. Distribution of all the BNs in the randomly generated network with 12 genes, totally 4096 BNs.

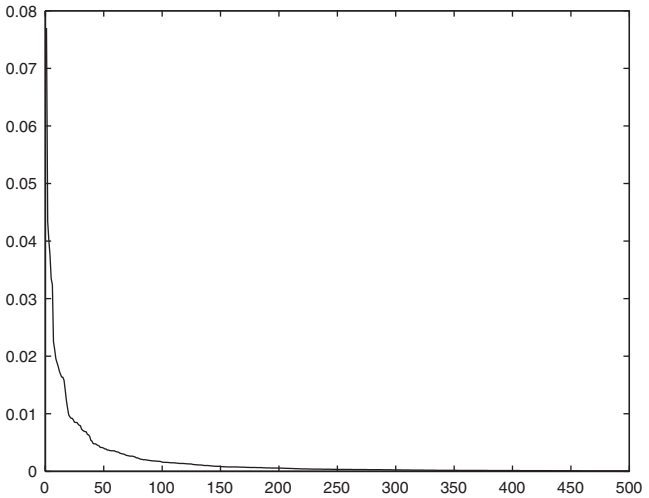

Fig. 3. Distribution of the first 500 BNs with highest probability in the randomly generated network with 12 genes.

Table 5. Numerical results of the approximation method for the PBNs with number of Boolean functions equal to two for the instantaneously random PBN with 12 genes

Lower bound of p	$Error_{\|\cdot\|_1}$	$Error_{\|\cdot\|_2}$	$Error_{\|\cdot\|_\infty}$	n_0	Time (s)	n_{match} out of 500
10^{-4}	9.21×10^{-1}	6.54×10^{-2}	2.91×10^{-2}	3646	326	495
10^{-5}	1.55×10^{-1}	1.04×10^{-2}	4.50×10^{-3}	3052	663	498
10^{-6}	2.05×10^{-2}	1.30×10^{-3}	5.36×10^{-4}	2261	1115	500

Explanations of the table is similar to that for Table 2. The random gene perturbation probability is 0.03. The total computational time for the construction of the true transition probability matrix without perturbation is 2395 s.

Table 6. Numerical results of the approximation method for the PBNs with number of Boolean functions equal to 2 for the context-sensitive PBN with 12 genes

Lower bound of p	$Error_{\|\cdot\|_1}$	$Error_{\|\cdot\|_2}$	$Error_{\|\cdot\|_\infty}$	n_0	Time (s)	n_{match} out of 500
10^{-4}	9.27×10^{-1}	6.55×10^{-2}	2.91×10^{-2}	3646	750	495
10^{-5}	1.57×10^{-1}	1.04×10^{-2}	4.50×10^{-3}	3052	1060	500
10^{-6}	2.07×10^{-2}	1.30×10^{-3}	5.37×10^{-4}	2261	1475	500

Explanations of the table is similar to that for Table 2. The transition probability of one BN to other BNs is 0.5. The random gene perturbation probability is 0.03. The total computational time for the construction of the true PBN transition matrix without perturbation is 2620 s.

probability distribution of all BNs and error evaluation. Furthermore, we gave the theoretical results on the probability distribution of number (proportion) of BNs dropped given a threshold level with at most two Boolean functions for one gene. Numerical experiments are given to demonstrate both the efficiency and effectiveness of the proposed method. Since the total number of states increases exponentially with respect to the number of genes, it is still a problem to compute the steady-state probability distribution for a general large network.

ACKNOWLEDGEMENTS

W.-K.C. is supported in part by RGC Grant 7017/07P, HKU Strategic Research Theme Fund on Computational Physics and Numerical Methods, Hung Hing Ying Physical Research Fund, HKU GRCC Grants Nos. 10206647, 10206483 and 10206147.
M.K.N. is supported in part by RGC 7046/03P, 7035/04P, 7035/05P and HKBU FRGs.
T.A. is partially supported by Grant-in-Aid Systems Genomics from MEXT, Japan.

Conflict of Interest: none declared.

REFERENCES

Brun,M. *et al.* (2005) Steady-state probabilities for attractors in probabilistic Boolean networks. *EURASIP J. Signal Processing*, **85**, 1993–2013.
Celis,J.E. *et al.* (2000) Gene expression profiling: monitoring transcription and translation products using DNA microarrays and proteomics. *FEBS Lett.*, **480**, 2–16.
Datta,A. *et al.* (2003) External control in Markovian genetic regulatory networks. *Mach. Learn.*, **52**, 169–191.
Datta,A. *et al.* (2004) External control in Markovian genetic regulatory networks: the imperfect information case. *Bioinformatics*, **20**, 924–930.
Guelzim,N. *et al.* (2002) Topological and causal structure of the yeast transcriptional regulatory network. *Nat. Genet.*, **31**, 60–63.
Huang,S. (1999) Gene expression profiling, genetic networks, and cellular states: an integrating concept for tumorigenesis and drug discovery. *J. Mol. Med.*, **77**, 469–480.
Hughes,T.R. *et al.* (2001) Expression profiling using microarrays fabricated by an ink-jet oligonucleotide synthesizer. *Nat. Biotechnol.*, **19**, 342–347.
Jong,H.D. (2002) Modeling and simulation of genetic regulatory systems: a literature review. *J. Comp. Biol.*, **9**, 67–103.
Kauffman,S.A. (1993) *The Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press, New York.
Lähdesmäki,H. *et al.* (2006) Relationships between probabilistic Boolean networks and dynamic Bayesian networks as models of gene regulatory networks. *Signal Processing*, **86**, 814–834.
Lipshutz,R.J. *et al.* (1999) High density synthetic oligonucleotide arrays. *Nat. Genet.*, **21**, 20–24.
Lockhart,D.J. and Winzeler,E.A. (2000) Genomics, gene expression and DNA arrays. *Nature*, **405**, 827–836.
Motwani,R. and Raghavan,P. (1995) *Randomized Algorithms*. Cambridge University Press, New York.
Ng,M.K. *et al.* (2006) A control model for Markovian genetic regulatory networks. *Trans. Comput. Syst. Biol.*, **4070**, 36–48.
Pal,R. *et al.* (2005) Intervention in context-sensitive probabilistic Boolean networks. *Bioinformatics*, **21**, 1211–1218.
Rosenthal,J.S. (1995) Minorization conditions and convergence rates for Markov chain Monte Carlo. *J. Am. Stat. Assoc.*, **90**, 558–566.
Ross,S.M. (1997) *Introduction to Probability Model*. 7th edn. Academic Press, San Diego, CA.
Schena,M. *et al.* (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, **270**, 467–470.
Shmulevich,I. and Zhang,W. (2002) Binary analysis and optimization-based normalization of gene expression data. *Bioinformatics*, **18**, 555–565.
Shmulevich,I. *et al.* (2002a) From Boolean to probabilistic Boolean networks as models of genetic regulatory networks. *Proc. IEEE*, **90**, 1778–1792.
Shmulevich,I. *et al.* (2002b) Gene perturbation and intervention in probabilistic Boolean networks. *Bioinformatics*, **18**, 1319–1331.
Shmulevich,I. *et al.* (2002c) Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, **18**, 261–274.
Shmulevich,I. *et al.* (2003) Steady-state analysis of genetic regulatory networks modeled by probabilistic Boolean networks. *Comp. Funct. Genomics*, **4**, 601–608.
Somogyi,R. and Sniegoski,C. (1996) Modeling the complexity of gene networks: understanding multigenic and pleiotropic regulation. *Complexity*, **1**, 45–63.
Szallasi,Z. and Liang,S. (1998) Modeling the normal and neoplastic cell cycle with 'realistic Boolean genetic networks': their application for understanding carcinogenesis and assessing therapeutic strategies. *Proc. Pac. Symp. Biocomput.*, **3**, 66–76.
Wuensche,A. (1998) Genomic regulation modeled as a network with basins of attraction. *Proc. Pac. Symp. Biocomput.*, **3**, 89–102.
Zhang,S. *et al.* (2007) Simulation study in probabilistic Boolean network models for genetic regulatory networks. *Int. J. Data Min. Bioinformatics*, **1**, 217–240.