確率ブーリアンネットワークのアトラクター検出とブーリアンネットワークとのアトラクターの関係

林田 守広[†] 田村 武幸[†] 阿久津 達也[†] 程 王韋王其[‡] 叢 陽[‡]

[†] 京都大学 化学研究所 バイオインフォマティクスセンター
[‡] 香港大学 数学科

概要 遺伝子制御ネットワークを数理的に表すモデルの一つに確率ブーリアンネットワーク (PBN) がある。本稿では PBN のアトラクターについて扱う。まず n を PBN の頂点数、L を各頂点に割当てられるブール関数の数とすると周期 1 のアトラクターの数の期待値は $(2-(\frac{1}{2})^{L-1})^n$ となることを示す。次に PBN のアトラクターを検出するアルゴリズムを提案し、平均時間計算量について理論的な解析を与えるとともに計算機実験を行うことで、2^n の全状態を探索する単純なアルゴリズムよりも遥かに効率が良いことを示す。さらにブーリアンネットワーク (BN) のアトラクター分布と BN から構成したある種の PBN の定常状態分布との関係についても示す。

Probabilistic Boolean Network Attractor Identification and a Relationship to Attractor Distribution in Boolean Networks

Morihiro Hayashida[†], Takeyuki Tamura[†], Tatsuya Akutsu[†],
Wai-Ki Ching[‡] and Yang Cong[‡]

[†]*Bioinformatics Center, Institute for Chemical Research, Kyoto University,*
Gokasho, Uji, Kyoto 611-0011, Japan.
[‡]*Advanced Modelling and Applied Computing Laboratory, Department of Mathematics,*
The University of Hong Kong, Pokfulam Road, Hong Kong.

Abstract Probabilistic Boolean Networks (PBNs) have been proposed as one of mathematical models for gene regulatory networks. In this work, we study attractors in PBNs. We study the expected number of singleton attractors in a PBN and show that it is $(2-(\frac{1}{2})^{L-1})^n$, where n is the number of nodes in a PBN and L is the number of Boolean functions assigned to each node. Then, we present an algorithm for identifying singleton and small attractors and perform both theoretical and computational analyses on their average case time complexities. The results of computational experiments suggest that this algorithm is much more efficient than the naive algorithm that examines all possible 2^n states. We also show a simple and interesting relation between the distribution of attractors in a BN and the steady-state distribution in a corresponding variant of a PBN.

1 Introduction

To understand the mechanism of gene regulatory networks is an important topic in computational systems biology. Many mathematical models of genetic networks have been proposed, which include Bayesian networks, Boolean networks and probabilistic Boolean networks, ordinary and partial differential equations, and qualitative differential equations. Among them, a lot of studies have been done on the Boolean network (BN in short). BN is a very simple model which combines genetic networks with Boolean algebra. In this model, each node (e.g., gene) takes either 0 (not expressed) or 1 (expressed) and the states of nodes change synchronously according to regulation rules given as Boolean functions. Though many aspects of Boolean networks have been studied, extensive studies have been done on the distribution of length and number of attractors for randomly generated BNs with an average indegree, where an attractor corresponds to a steady-state of a cell. However, exact results have not yet been obtained.

Although BN is a deterministic model, real genetic networks contain some uncertainty. In order to cope with this uncertainty, the probabilistic Boolean network (PBN in short) was proposed as a stochastic extension of BN. Unfortunately, it takes at least $O(2^n)$ computational time because the size of a vector representing the probability distribution is 2^n, where n is the number of nodes in a PBN. However, in many cases, it might be enough to know approximate probabilities of major states. Furthermore, it may be helpful to know attractors in PBNs because singleton or small attractors may correspond to major states

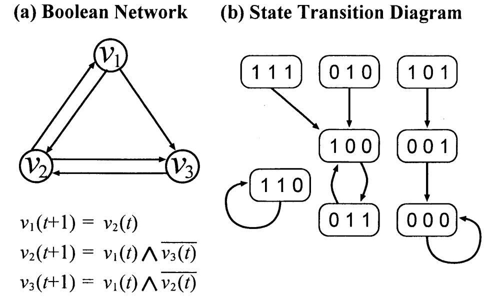

Figure 1: Example of Boolean network (a), and its state transition diagram (b).

in the steady-state probability distribution. Indeed, Brun et al. studied relations between attractors and steady-state probability distributions [1]. However, they did not provide efficient algorithms for computing attractors in BNs or PBNs.

In this work, we study attractors in PBNs, where attractors do not correspond to steady-states but correspond to attractors in BNs [2]. First, we study the expected number of singleton attractors in PBNs. In particular, we show that this number is $(2-(\frac{1}{2})^{L-1})^n$ for a PBN, where L is the number of Boolean functions which are assigned to each node. Next, we present an algorithm for computing singleton and small attractors by extending the techniques proposed in [3]. We show that the average case time complexity is $o(2^n)$ in many cases, which suggests that computation of singleton attractors in PBNs is easier than computation of steady-state distributions in PBNs. We also perform computational experiments on randomly generated networks in order to verify the theoretically derived time complexities. Then, we show a simple relation between attractors and steady-state probability distributions for a corresponding variant of PBN.

2 BN, PBN and Attractors

A BN $G(V,F)$ consists of a set $V = \{v_1,\dots,v_n\}$ of *nodes* and a list $F = (f_1,\dots,f_n)$ of *Boolean functions* (see Fig. 1(a)). Then, the state of node v_i at time $t+1$, $v_i(t+1)$, is determined by $v_i(t+1) = f_i(v_{i_1}(t),\dots,v_{i_{|IN(v_i)|}}(t))$, where $IN(v_i)$ denotes the set of input nodes to v_i. $|IN(v_i)|$ is called the *indegree* of v_i. We use K to denote the *maximum indegree* of a BN. Here we let $\boldsymbol{v}(t) = [v_1(t),\dots,v_n(t)]$, which is called a *Gene Activity Profile* (GAP) or a (global) state of BN at time t. The dynamics of a BN can be well-described by a *state transition diagram* shown in Fig. 1(b). From this diagram, it can be seen that if $\boldsymbol{v}(0) = [1,0,1]$, GAP changes as $[1,0,1] \Rightarrow [0,0,1] \Rightarrow [0,0,0] \Rightarrow [0,0,0] \Rightarrow \cdots$ and the same GAP $[0,0,0]$ is repeated after $t=1$. These kinds of sets of repeating states are called *attractors*, each of which corresponds to a directed cycle in a state transition diagram. The number of elements in an attractor is called the *period* of the attractor. An attractor with period 1 (greater than 1) is called a *singleton* (*cyclic*) attractor.

PBN is an extension of BN. The difference between BN and PBN is only that in a PBN, for each vertex v_i, instead of having only one Boolean function, there are a number of Boolean functions (predictor functions) $f_j^{(i)} (j = 1,\dots,l(i))$ to be chosen for determining the state of gene v_i. The probability of choosing $f_j^{(i)}$ is $c_j^{(i)}$ ($0 \le c_j^{(i)} \le 1$ and $\sum_{j=1}^{l(i)} c_j^{(i)} = 1$ for $i = 1,\dots,n$). The dynamics of a PBN can be well described by the state transition probabilities. Let $\boldsymbol{u}$ be a GAP of a PBN at time t. Then, we can calculate the probability that the PBN takes a GAP $\boldsymbol{w}$ at time $t+1$ as $Prob(\boldsymbol{v}(t+1) = \boldsymbol{w} \mid \boldsymbol{v}(t) = \boldsymbol{u})$. Since there are 2^n GAPs, these probabilities can be represented by a $2^n \times 2^n$ matrix called the transition probability matrix. In a PBN, we also call a set of GAPs $\{\boldsymbol{v}_1,\dots,\boldsymbol{v}_p\}$ an *attractor of period* p if $Prob(\boldsymbol{v}(t+1) = \boldsymbol{v}_{i+1} | \boldsymbol{v}(t) = \boldsymbol{v}_i) \neq 0$ holds for all $i = 1,\dots,p$, where we identify $p+1$ and 1. As in BN, we also call an attractor with period 1 a *singleton attractor*.

3 Expected Number of Singleton Attractors in PBN

In this section, we show how many singleton attractors there are in a PBN in average. Recall that $\boldsymbol{u}$ is called a singleton attractor of a PBN if $f_j(\boldsymbol{u}) = \boldsymbol{u}$ holds for some j $(1 \le j \le N)$. The following theorem shows that the expected number of singleton attractors in a PBN is quite large when compared to that in a BN, which is known to be 1.

Theorem 1 *Suppose that f_i and $|IN(v_i)|$ are randomly assigned for each v_i. When $l(i) = L$ holds for any i, the expected number of singleton attractors in a PBN is* $\left\{2-\left(\frac{1}{2}\right)^{L-1}\right\}^n$.

Proof. For each v_i, the probability that $f_j^{(i)}(v_i(t)) \neq a)$ holds for any j is $(\frac{1}{2})^L$ where $a \in \{0,1\}$. Therefore, since $Prob(\boldsymbol{v}(t+1) = \boldsymbol{u} \mid \boldsymbol{v}(t) = \boldsymbol{u})$ is $\{1-(\frac{1}{2})^L\}^n$, the expected number of singleton attractors is $2^n \times \left\{1-\left(\frac{1}{2}\right)^L\right\}^n = \left\{2-\left(\frac{1}{2}\right)^{L-1}\right\}^n$. □

4 Algorithm for Computing Attractors in a PBN

In this section, we present an algorithm for finding all attractors of a PBN by extending branch-and-bound type algorithms proposed in [3].

We call *type-1 problem* a problem for finding attractors with period $x \le p$ of a PBN. The overall computational time for type-1 problem can be represented by $time(x=1)+\cdots+time(x=p)$, where each term will be an exponential of n as shown below. Therefore, when p is small, the overall average time complexity will only be affected by the largest term in the above formula. Thus, the order of the time complexity for a problem of finding attractors with period p, which we call *type-2 problem*, is theoretically the same as that of type-1 problem. Therefore, we consider type-2 problem instead of type-1 problem in the following. The pseudo code of the proposed algorithm is given below:

```
Pseudo code
Input: a PBN
Output: all attractors with period $p$
Begin
  define $x$-ancestor($v_i$): nodes initializing paths
  to $v_i$ whose lengths are less than or equal to $x$.
  initialize $m = 1$;
  Procedure $PBNAttractor(\boldsymbol{v}, m)$
   if $m = n + 1$ then output $\boldsymbol{v}(t)$ return;
   for $b = 0$ to $1$ do $v_m(t) = b$;
    $flag = 0$; $i = 1$;
    while $flag = 0$ and $i \le m$
     if every $p$-ancestor($v_i$) is assigned
      $r = 0$;
      while $flag = 0$ and $r \le p - 1$
       if it is found that
         $f_j^{(i)}(v_i(t+r)) \neq v_i(t+r+1)$
         holds for each $j$
       then $flag = 1$;
       $r = r + 1$;
     $i = i + 1$;
    if $flag = 1$ then continue;
    else $PBNAttractor(\boldsymbol{v}, m + 1)$;
   return;
End
```

Now, we perform theoretical analysis of average computational time of the above algorithm. We assume that $l(i) = L$ and $c_1^{(i)} = \cdots = c_{l(i)}^{(i)}$ hold for any i. Assume that m of n nodes have already been examined. Since $|p - ancestor(v_i)|$ is approximately $\sum_{x=1}^{p} |IN(v_i)|^x$, the probability that $v_i(t) \neq v_i(t+p)$ holds for some i is approximately

$$\left\{ 0.5 \times \frac{\binom{m}{\sum_{x=1}^{p} |IN(v_i)|^x}}{\binom{n}{\sum_{x=1}^{p} |IN(v_i)|^x}} \right\}^L \approx \left\{ 0.5 \cdot \left(\frac{m}{n}\right)^{\sum_{x=1}^{p} K^x} \right\}^L .$$

The probability that the algorithm examines the $(m+1)$-th gene is no more than $\left\{1 - 0.5^L \left(\frac{m}{n}\right)^{L\sum_{x=1}^{p} K^x}\right\}^m$. The number of recursive calls executed for the first m genes is, by setting $s = \frac{m}{n}$, at most

$$2^m \times \left\{1 - 0.5^L s^{L\sum_{x=1}^{p} K^x}\right\}^m = \{(2 - 0.5^{L-1} s^{L \times \frac{K(1-K^p)}{1-K}})^s\}^n .$$

We can confirm that some maximum values of this formula are as shown in Table 1.

Table 1: Theoretically estimated average case time complexity for $p = 1$, $K \in \{2,3,4\}$, $L \in \{2,3\}$.

	$L=2$	$L=3$
K=2	$O(1.601^n)$	$O(1.763^n)$
K=3	$O(1.658^n)$	$O(1.790^n)$
K=4	$O(1.698^n)$	$O(1.813^n)$

Table 2: Empirical numbers of singleton attractors in PBN for $K \in \{2,3,4\}$ and $L \in \{2,3\}$.

	$L=2$	$L=3$
$K=2$	1.499^n	1.751^n
$K=3$	1.500^n	1.750^n
$K=4$	1.500^n	1.749^n

5 Computational Experiments

We evaluated expected numbers of singleton attractors in PBN and time complexities of the proposed algorithm. For each $K \in \{2,3,4\}$, we randomly generated 100 PBNs with $n \in \{5,10,15,20,25,30\}$ nodes and $L \in \{2,3\}$ Boolean functions for each node, and took the average values. All of these computational experiments were done on a PC with Xeon X5460 3.16GHz CPUs under Linux (version 2.6).

Table 2 shows the empirical numbers of singleton attractors in PBN for each $L \in \{2,3\}$. We used a tool for GNUPLOT to fit the function a^n to the experimental results. This tool uses the nonlinear least-squares (NLLS) Marquardt-Levenberg algorithm. We can see from the table that the empirical numbers were almost the same as those obtained by theoretical analysis.

Table 3 shows the empirical time complexities of the proposed algorithm estimated by 100 randomly generated PBNs with up to 30 nodes for $p = 1$, $K \in \{2,3,4\}$ and $L \in \{2,3\}$. We fit the function $b \times a^n$ to the experimental results in order to obtain the empirical time complexities. The empirical time complexities were slightly larger than those derived from the theoretical analysis. However, it was still faster than the naive algorithm which examines all 2^n states.

Table 3: Empirical time complexities estimated using 100 randomly generated PBNs with up to 30 nodes for $p = 1$, $K \in \{2, 3, 4\}$ and $L \in \{2, 3\}$

	$L = 2$	$L = 3$
$K = 2$	$O(1.694^n)$	$O(1.855^n)$
$K = 3$	$O(1.758^n)$	$O(1.904^n)$
$K = 4$	$O(1.779^n)$	$O(1.920^n)$

6 Relation between Steady-State Probabilities and Attractors

In this section, we discuss a simple relation between the steady-state probability distribution in a variant of PBN and the structure of attractors in a BN.

Let $\boldsymbol{u}, \boldsymbol{w}$ be GAPs of a BN at time t and $t + 1$, respectively. Since GAP at time $t + 1$ is given deterministically from GAP at time t in BN, we can write

$$Prob(\boldsymbol{v}(t+1) = \boldsymbol{w} | \boldsymbol{v}(t) = \boldsymbol{u}) \;=\; 1.$$

For the other GAPs $\boldsymbol{w}'$, we can write

$$Prob(\boldsymbol{v}(t+1) = \boldsymbol{w}' | \boldsymbol{v}(t) = \boldsymbol{u}) \;=\; 0.$$

The dynamics of the BN can be represented by a $2^n \times 2^n$ matrix $\mathbf{P}$. Here, we construct a variant of PBN by introducing slight perturbation to this BN. If $\boldsymbol{w}$ is the next GAP to $\boldsymbol{u}$ in the above BN, we let

$$Prob(\boldsymbol{v}(t+1) = \boldsymbol{w} | \boldsymbol{v}(t) = \boldsymbol{u}) \;=\; 1 - \left(1 - \frac{1}{2^n}\right)\epsilon,$$

where $\epsilon \in (0, 1]$. For other $\boldsymbol{w}'$, we let

$$Prob(\boldsymbol{v}(t+1) = \boldsymbol{w}' | \boldsymbol{v}(t) = \boldsymbol{u}) \;=\; \frac{\epsilon}{2^n}.$$

We consider the case when ϵ is close to 0. Let $\mathbf{P}_\epsilon$ be the $2^n \times 2^n$ matrix corresponding to these transition probabilities. Then we have

$$\mathbf{P}_\epsilon = (1 - \epsilon)\mathbf{P} + \frac{\epsilon}{2^n}\mathbf{1}\mathbf{1}^t,$$

where $\mathbf{1} = (1\ 1\ \cdots\ 1)^t$ of size 2^n and $\mathbf{H}^t$ denotes the transposed matrix of $\mathbf{H}$.

Let $\boldsymbol{x}_\epsilon$ be the corresponding steady-state probability vector of size 2^n (i.e., $\boldsymbol{x}_\epsilon = \mathbf{P}_\epsilon \boldsymbol{x}_\epsilon$). Then, $\boldsymbol{x}_\epsilon$ is uniquely determined as the following lemma.

Lemma 1 *The vector* $\boldsymbol{x}_\epsilon = \frac{\epsilon}{2^n}(\mathbf{I} - (1 - \epsilon)\mathbf{P})^{-1}\mathbf{1}$ *is the steady-state probability distribution of the PBN, where* $\mathbf{P}$ *is the transition matrix of the original BN, and* $\mathbf{I}$ *is the identity matrix of size* 2^n.

Let A be the set of GAPs in an attractor with period p in a BN (i.e., $p = |A|$). Let B be the *basin* of the attractor A. That is, B is the set of GAPs that eventually fall into the attractor A (i.e., $A \subseteq B$).

We use $\pi(B)$ to denote the sum of probabilities of the GAPs that fall into the basin B. Then, we have $\pi(B) = \sum_{i \in B} [\boldsymbol{x}_\epsilon]_i$ and the following result.

Theorem 2 *In the steady state, the probability that the distribution state falls into the attractor basin set* B *is* $\pi(B) = \frac{|B|}{2^n}$.

Then, as shown in [1], we have

$$[\boldsymbol{x}_\epsilon]_i \approx \begin{cases} \frac{1}{|A|} \times \pi(B) = \frac{|B|}{|A| \times 2^n}, & \text{if } i \in A, \\ 0, & \text{if } i \in B - A. \end{cases}$$

This result is interesting because it relates steady-state probabilities with the sizes of attractors and their basins.

7 Conclusion

In this work, we have studied attractors in PBNs. We showed theoretical and experimental results on the number of singleton attractors in PBNs. It is suggested that different attractors correspond to different cell types. Since 1 is too small and 1.5^n is too large, these results suggest that many of genes may have only one control rule (per gene) and several or more genes may have multiple control rules. We presented an algorithm for finding singleton and small attractors in PBNs and performed theoretical and empirical analyses of their average case time complexities. Though the proposed algorithm is much faster than the naive algorithm that examines all 2^n states, these cannot yet handle very large PBNs. In particular, it is very difficult for the case of finding cyclic attractors. Therefore, improvement of the algorithm for finding attractors in PBNs, especially for finding cyclic attractors, is important future work. In order to relate attractors in BNs with steady-state probability distributions in PBNs, we derived a simple and interesting relation. However, this relation holds only for a special variant of PBN, which is considered as a BN with very small noise. Since currently available algorithms can not yet be applied to large networks, development of faster algorithms for finding or approximating steady-state probability distributions is also left as future work.

References

[1] M. Brun, E.R. Dougherty, I. Shmulevich. *Signal Processing*, 85, 1993–2013, 2005.

[2] M. Hayashida, T. Tamura, T. Akutsu, W. Ching. *Lecture Notes in Operations Research*, 9, 91–100, 2008.

[3] S. Zhang, M. Hayashida, T. Akutsu, W. Ching, M. Ng. *EURASIP Journal on Bioinformatics and Systems Biology*, Article ID 20180, 2007.