

# Dynamics of Random Boolean Networks under Fully Asynchronous Stochastic Update Based on Linear Representation

Chao Luo, Xingyuan Wang\*

Faculty of Electronic Information and Electrical Engineering, Dalian University of Technology, Dalian, China

## Abstract

A novel algebraic approach is proposed to study dynamics of asynchronous random Boolean networks where a random number of nodes can be updated at each time step (ARBNs). In this article, the logical equations of ARBNs are converted into the discrete-time linear representation and dynamical behaviors of systems are investigated. We provide a general formula of network transition matrices of ARBNs as well as a necessary and sufficient algebraic criterion to determine whether a group of given states compose an attractor of lengths in ARBNs. Consequently, algorithms are achieved to find all of the attractors and basins in ARBNs. Examples are showed to demonstrate the feasibility of the proposed scheme.

**Citation:** Luo C, Wang X (2013) Dynamics of Random Boolean Networks under Fully Asynchronous Stochastic Update Based on Linear Representation. PLoS ONE 8(6): e66491. doi:10.1371/journal.pone.0066491

**Editor:** Peter Csermely, Semmelweis University, Hungary

**Received:** March 18, 2013; **Accepted:** May 6, 2013; **Published:** June 13, 2013

**Copyright:** © 2013 Luo and Wang. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Funding:** This research is supported by the National Natural Science Foundation of China (Nos: 61173183, 60973152, and 60573172), the Superior University Doctor Subject Special Scientific Research Foundation of China (No: 20070141014), Program for Liaoning Excellent Talents in University (No: LR2012003), the National Natural Science Foundation of Liaoning province (No: 20082165) and the Fundamental Research Funds for the Central Universities (No: DUT12JB06). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing Interests:** The authors have declared that no competing interests exist.

\* E-mail: wangxy@dlut.edu.cn

## Introduction

The gene regulatory networks (GRNs) as complex systems present diverse dynamical behaviors. In 1969, random Boolean networks (RBNs) were first introduced by Stuart Kauffman to model gene regulatory networks [1]. Each gene is represented by a node in RBNs with two possible states, i.e. the logical 0 and 1. Generally, the “1” value represents the state “on” corresponding to a gene that is being transcribed and the “0” value represents “off” corresponding to a gene that is not being transcribed [2]. A directed edge from one node to another stands for the interaction between genes, the mutual regulation of which is described by an assigned Boolean function. The values of all nodes in a network are updated in parallel and all of them together constitute the state of the whole network. Since dynamics is deterministic and the state space is finite, the states of a particular RBN eventually converge into a series of periodically recurring states, which are called attractors including the fixed points and cycles, starting from an arbitrary initial condition. Attractors and their transient states compose the basins of attraction, which present the dynamical behaviors of systems. RBNs provide a way to formalize large scale complexity in a realistic manner by describing the gene expression with ON/OFF and reflect the statistical features of real living cells by adjusting the parameters of networks [3]. Recent years, along with the research of networks in molecular biology, chemistry, neurobiology, and economy etc [4–16], RBNs have been extensively investigated and a wealth of results have been achieved [17–28].

As abstract models of physical and biological phenomena, a certain degree of simplification is necessary. Usually, synchrony idealization is adopted as update scheme in the classical RBNs, as

Ref. [22], “The simplest class of Boolean network is synchronous, which means that all elements update their activities at the same moment”. All of elements in RBNs evolve according to a global synchronized clock, which is based on a tacit assumption that update scheme would not affect the essential properties of dynamics. However, some doubts have been cast on the validity of the above assumption when more and more evidences showed the dynamic behaviors of systems under asynchronous update presented considerable divergence comparing with the counterparts under synchronous update [29–30]. ARBNs was firstly proposed in 1997 by Harvey et al. [31] where only one node was randomly chosen for update at each time step. Since then, many results on ARBNs have been presented and some modified versions of ARBNs were also proposed [32–38]. We think, if “synchronous update” is considered as “idealization” [31], Harvey’s model as the extreme opposite of synchronous models should be another form of “idealization”. Motivated biologically, since the expression of genes is not an instantaneous process, but the transcription of DNA and the transport of enzymes may take from milliseconds up to a few seconds [36]. So when modeling the above dynamics by means of a discrete form, at each time step, the condition to strictly limit only one node for update is too strong to describe the real situation.

Although RBNs have been much analyzed, but as logical systems, they are difficult to be investigated analytically. Recently, a new matrix theory called the semi-tensor product (STP) was proposed by Cheng et al. [39], whereby logical equations can be represented as a linear discrete-time dynamic equation

$$x(t+1) = L \times x(t) \quad (1)$$

where  $x(t) = \times_{i=1}^n x_i$  is the semi-tensor product of logical variables. In contrast with usual transition matrix expression, Eq. (1) contains complete information of the logical equations of RBNs, based on which dynamics of systems, such as attractors and basins, can be analyzed. Furthermore, this form can easily be extended to the Boolean control networks. Certain control problems, such as controllability [40,41], observability [42], and realization [43], etc., can be investigated. So far, the previous studies mainly focus on models under synchronous update. However, ARBNs as nondeterministic systems are considerable different from RBNs. For instance, in Ref. [39], Cheng et al. proposed a method by STP technique to find attractors of RBNs, but, which is based on a fact that RBNs are deterministic systems and attractors are circular. Obviously, this prerequisite can't be satisfied for ARBNs. So, we think it's interesting and meaningful to extend the related research into the field of asynchronous random Boolean networks.

Based on the above discussion, firstly, from the aspect of model selection, most of the previous results on ARBNs are based on the Harvey's model. As mentioned above, we think Boolean networks with a more general asynchronous update schedule are worth investigating. Secondly, from the aspect of methodology selection, despite of the overload of matrix operations, STP technique provides a new perspective to make the analytical analysis of logical dynamics. From the viewpoint of theoretical part, it's a valuable technique to be considered. Therefore, in this article, we focus on dynamics of ARBNs where a random number of nodes can be updated at each time step. Mainly, two parts of work are involved in this article. First of all, we complete an algebraic representation of the studied logical models. At the best of our knowledge, the related result is firstly presented. Network transition matrix takes the essential role for the linear representation of logical dynamics based on STP technique. Actually, in the previous works, Cheng et al. [39] have achieved results of RBNs. Here, we engage in a particular class of ARBNs, from the perspective of update schedule, this special update scheme can involve the case of synchronous update, which allow all of nodes are updated at a certain time step. As a result, a general formula of network transition matrices is achieved, based on which one can calculate all of network transition matrices of a particular ARBN including the counterparts under synchronous update and Harvey's model. Secondly, a necessary and sufficient algebraic criterion is discussed, which determine whether a given group of  $s$  states could compose a loose attractor of length  $s$  in a specific ARBN. Consequently, a novel approach to detect the attractors and basins of ARBNs is designed, which is completely based on set operations and different from the previous studies.

As we know, to search attractors in RBNs is a NP-hard problem. Recent years, many powerful algorithms [3,17,20,35,45] have been presented to improve the search efficiency and reduce the time complexity. Yet, it should be noted that our main interest in this article is to provide a complete quantitative method and a novel perspective to analyze the dynamics of ARBNs from the view point of theoretical part. And results would be taken as preparations for the further research, e.g. Boolean control network under asynchronous update.

This article is organized as follows. In the section of Methods, some concepts and properties of STP are introduced. In Results and discussion, the linear representation of ARBNs is discussed, based on which the dynamic properties of ARBNs are studied and algorithms to find attractors and basins of ARBNs are presented. Some examples are shown to illustrate the main results. Finally, a concluding remark is given.

## Methods

In this section, the semi-tensor product of matrices is briefly introduced. Some concepts and properties related to this article are presented.

### Definition 1 ([39])

- 1) Let  $X$  be a row vector of dimension  $np$ , and  $Y$  be a column vector of dimension  $p$ . Then we split  $X$  into  $p$  equal-size blocks as  $X^1, \dots, X^p$ , which are  $1 \times n$  rows. Define the STP, denoted by  $\times$ , as

$$\begin{cases} X \times Y = \sum_{i=1}^p X^i y_i \in \mathbb{R}^n, \\ Y^T \times X^T = \sum_{i=1}^p y_i (X^i)^T \in \mathbb{R}^n. \end{cases}$$

- 2) Let  $A \in M_{m \times n}$  and  $B \in M_{p \times q}$ . If either  $n$  is a factor of  $p$ , say  $nt = p$  and denote it as  $A \prec_t B$ , or  $p$  is a factor of  $n$ , say  $n = pt$  and denote it as  $A \succ_t B$ , then we define the STP of  $A$  and  $B$ , denoted by  $C = A \times B$ , as the following:  $C$  consists of  $m \times q$  blocks as  $C = (C^{ij})$  and each block is

$$C^{ij} = A^i \times B_j, \quad i = 1, \dots, m, \quad j = 1, \dots, q$$

where  $A^i$  is the  $i$ -th row of  $A$  and  $B_j$  is the  $j$ -th column of  $B$ .

It's easy to verify that for two column vectors  $X \in \mathbb{R}^m$  and  $Y \in \mathbb{R}^n$ ,  $X \times Y \in \mathbb{R}^{mn}$ .

Note that STP of matrix can be seen as a generalization of the conventional matrix product, so the properties of matrix product, such as distributive rule, associative rule, etc, still hold.

Some notations in this article are defined as follows

- 1)  $\delta_n^r$  denotes the  $r$ -th column of the  $n \times n$  identity matrix  $I_n$  and  $\Delta_n = \{\delta_n^r | 1 \leq r \leq n\}$ , which is the set of all  $n$  columns of  $I_n$ . An operation is defined as  $\Theta(\delta_n^r) = \{r\}$ , furthermore, when  $C = \{\delta_n^{i_1}, \delta_n^{i_2}, \dots, \delta_n^{i_p}\}$ ,  $\Theta(C) = \{i_1, i_2, \dots, i_p\}$ .
- 2) A matrix  $\Lambda \in M_{n \times m}$  can be called a logical matrix if  $\Lambda = [\delta_n^{i_1}, \delta_n^{i_2}, \dots, \delta_n^{i_m}]$ , which is briefly denoted by  $\Lambda = \delta_n[i_1, i_2, \dots, i_m]$ . And the set of  $n \times m$  logical matrices is denoted by  $\mathcal{L}_{n \times m}$ .
- 3) Let matrix  $A \in M_{n \times m}$ , all of elements in  $i$ -th column of matrix  $A$  compose a set denoted by  $\Gamma(A)_i$ , and when a set  $S \in \rho(\{1, 2, \dots, m\})$ ,  $\Gamma(A)_S = \bigcup_{i \in S} \Gamma(A)_i$ , where  $\rho(\bullet)$  represents the power set.

Next, we define the swap matrix  $W_{[m,n]}$ , let  $X \in \mathbb{R}^m$  and  $Y \in \mathbb{R}^n$  be two column vectors

$$W_{[m,n]}XY = YX,$$

where  $W_{[m,n]}$  is a  $mn \times mn$  matrix labeled columns by  $(11, 12, \dots, 1n, \dots, m1, m2, \dots, mn)$  and rows by  $(11, 21, \dots, m1, \dots, 1n, 2n, \dots, mn)$ , the elements in position  $((I, J), (i, j))$  is

$$w_{(I,J),(i,j)} = \begin{cases} 1, & I=i \text{ and } J=j \\ 0, & \text{otherwise} \end{cases}$$

$W_{[m,m]}$  is briefly denoted by  $W_{[m]}$ .

In order to get the matrix expression of logic, the Boolean values should be denoted as vector forms  $T = 1 \sim \delta_2^1$  and  $F = 0 \sim \delta_2^2$ . For a logical function  $L(P_1, \dots, P_r)$  with arguments  $P_1, \dots, P_r$ , it has been proved in Ref. [39] that  $L(P_1, \dots, P_r)$  can be represented as  $L(P_1, \dots, P_r) = M_L P_1 P_2 \dots P_r$ , where matrix  $M_L \in M_{2 \times 2^r}$  is unique, which is called the structure matrix of logical function  $L$ .

More details on STP can be found in Ref. [39]. In this article, the matrix products are assumed to be STP and the symbol is omitted.

## Results and Discussion

### Model

In this section, the algebraic representation of asynchronous random Boolean networks is discussed. Without special note, ‘‘asynchrony’’ in this article implies  $m_t$  ( $m_t \in \{1, \dots, N\}$ ) nodes are randomly selected to be updated at time step  $t$ . Here, we don't consider the case  $m_t = 0$  i.e. none of nodes are updated.

ARBNs with  $N$  nodes can be described as follows:

$$\begin{aligned} x_i(t+1) &= f_i(x_{i_1}(t), \dots, x_{i_p}(t)), & i \in H_t \\ x_j(t+1) &= x_j(t), & j \in \{1, 2, \dots, N\} - H_t \end{aligned} \quad (2)$$

Where  $x_i(t)$  represents the state of node  $i$  at time  $t$ , which receives inputs from  $p$  distinct neighbors and  $x_{i_j}(t)$  is the  $j'$ th input. The nodes in ARBNs take values from the set  $\{0, 1\}$  corresponding to two levels of gene expressions.  $f_i$  is a Boolean logic from  $\{0, 1\}^p \rightarrow \{0, 1\}$ , which is assigned to node  $i$ . At time  $t$ , there are  $m_t$  ( $m_t \in \{1, \dots, N\}$ ) nodes which are selected at random for update, and all of updated elements are included in set  $H_t = \{i^1, \dots, i^{m_t}\}$ , without the loss of generality, we assume that  $i^{z_1} < i^{z_2}$  when  $\alpha_1 < \alpha_2$ . The updated value of selected node  $i$  is determined by all of the input values of  $x_{i_j}(t)$ ,  $j' \in \{1, 2, \dots, p\}$  and the logic  $f_i$ . At the same time, the remaining nodes keep the values at time step  $t$ .

Firstly, with the structure matrix  $M_i$  of each Boolean logic  $f_i$ , Eq. (2) can be converted into

$$\begin{aligned} x_i(t+1) &= M_i x(t), & i \in H_t \\ x_j(t+1) &= x_j(t), & j \in \{1, 2, \dots, N\} - H_t \end{aligned} \quad (3)$$

When the in-degree of node  $i$  is less than  $N$ , a dummy matrix  $E_d = \delta_2[1, 2, 1, 2]$  should be applied. It's easy to verify that for any Boolean variable  $u$ , one can get  $E_d u = I_2$ . Assume there was no input from node  $q$  to  $i$ ,  $x_q(t)$  can be substituted by  $E_d x_q(t)$ .

Multiplying all the equations in (3) as

$$\begin{aligned} x(t+1) &= x_1(t), \dots, x_{i^1-1}(t) M_{i^1} x(t) x_{i^1+1}(t), \dots, \\ & x_{i^{m_t}-1}(t) M_{i^{m_t}} x(t) x_{i^{m_t}+1}(t), \dots, x_N(t), \end{aligned} \quad (4)$$

According to Ref [39], when  $X_N(t) = \times_{i=1}^N x_i(t)$  and  $x_i(t) \in \Delta_2$ , one can obtain  $X_N^2(t) = \Phi_N X_N(t)$ , where  $\Phi_N = \prod_{i=1}^N I_{2^{i-1}} \otimes [(I_2 \otimes W_{[2, 2^{N-i}]} M_r)]$  and  $\otimes$  refers to the Kronecker product. Here,

$M_r = \delta_4[1, 4]$ , which is power-reducing matrix and it can be verified that  $P^2 = P M_r, \forall P \in \Delta_2$ .

Then, the following result can be achieved.

**Theorem 1.** Equation (4) can be represented as

$$x(t+1) = L_{H_t} x(t), \quad H_t \in \rho(\{1, 2, \dots, N\}), H_t \neq \Phi \quad (5)$$

$$\begin{aligned} \text{where } L_{H_t} &= \prod_{u=1}^{m_t} (I_{2^{m_t-u+(u-1)N}} \otimes M_{i^u}) \prod_{v=m_t}^1 (I_{2^{(i^v-1)+(v-1)(N-1)}} \\ & \otimes W_{[2^{N-i^v-(m_t-v), 2^N]}) \prod_{w=0}^{m_t-1} (I_{2^w} \otimes E_d) \prod_{y=1}^{m_t} W_{[2^{(m_t-y)+(i^y-1), 2}] \Phi_N^{m_t}. \end{aligned}$$

### Proof

$$\begin{aligned} x(t+1) &= x_1(t), \dots, x_{i^1-1}(t) M_{i^1} x(t) x_{i^1+1}(t), \dots, \\ & x_{i^{m_t}-1}(t) M_{i^{m_t}} x(t) x_{i^{m_t}+1}(t), \dots, x_N(t) \\ &= (I_{2^{i^1-1}} \otimes M_{i^1}) x_1(t), \dots, x_{i^2-1}(t) M_{i^2} x(t) x_{i^2+1}(t), \dots, \\ & x_{i^{m_t}-1}(t) M_{i^{m_t}} x(t) x_{i^{m_t}+1}(t), \dots, x_N(t) \\ &= \prod_{u=1}^{m_t} (I_{2^{m_t-u+(u-1)N}} \otimes M_{i^u}) x_1(t), \dots, \\ & x_{i^1-1}(t) x(t) x_{i^1+1}(t), \dots, \\ & x_{i^{m_t}-1}(t) x(t) x_{i^{m_t}+1}(t), \dots, x_N(t) \\ &= \prod_{u=1}^{m_t} (I_{2^{m_t-u+(u-1)N}} \otimes M_{i^u}) (I_{2^{(m_t-1)+(m_t-1)(N-1)}} \otimes \\ & W_{[2^{N-m_t}, 2^N]}) x_1(t), \dots, x_{i^1-1}(t) x(t) x_{i^1+1}(t), \dots, x_{i^{m_t}-1} \\ & (t) x_{i^{m_t}+1}(t), \dots, x_N(t) x(t) \\ &= \prod_{u=1}^{m_t} (I_{2^{m_t-u+(u-1)N}} \otimes M_{i^u}) \prod_{v=m_t}^1 (I_{2^{(i^v-1)+(v-1)(N-1)}} \\ & \otimes W_{[2^{N-i^v-(m_t-v), 2^N]}) x_1(t), \dots, \\ & x_{i^1-1}(t) x_{i^1+1}(t), \dots, x_{i^{m_t}-1}(t) x_{i^{m_t}+1}(t), \dots, x_N(t) x^{m_t}(t) \\ &= \prod_{u=1}^{m_t} (I_{2^{m_t-u+(u-1)N}} \otimes M_{i^u}) \prod_{v=m_t}^1 (I_{2^{(i^v-1)+(v-1)(N-1)}} \otimes \\ & W_{[2^{N-i^v-(m_t-v), 2^N]}) E_d (I_2 \otimes E_d), \dots, \\ & (I_{2^{m_t-1}} \otimes E_d) x_{i^1}(t) x_{i^2}(t), \dots, x_{i^{m_t}}(t) x_1(t), \dots, \\ & x_{i^1-1}(t) x_{i^1+1}(t), \dots, x_{i^{m_t}-1}(t) x_{i^{m_t}+1}(t), \dots, x_N(t) x^{m_t}(t) \\ &= \prod_{u=1}^{m_t} (I_{2^{m_t-u+(u-1)N}} \otimes M_{i^u}) \prod_{v=m_t}^1 (I_{2^{(i^v-1)+(v-1)(N-1)}} \otimes \\ & W_{[2^{N-i^v-(m_t-v), 2^N]}) \prod_{w=0}^{m_t-1} (I_{2^w} \otimes E_d) \\ & W_{[2^{(m_t-1)+(i^1-1), 2}] W_{[2^{(m_t-2)+(i^2-1), 2}], \dots, W_{[2^{(m_t-1), 2}] x^{m_t+1}(t) \\ &= \prod_{u=1}^{m_t} (I_{2^{m_t-u+(u-1)N}} \otimes M_{i^u}) \prod_{v=m_t}^1 (I_{2^{(i^v-1)+(v-1)(N-1)}} \otimes \\ & W_{[2^{N-i^v-(m_t-v), 2^N]}) \prod_{w=0}^{m_t-1} (I_{2^w} \otimes E_d) \\ & \prod_{y=1}^{m_t} W_{[2^{(m_t-y)+(i^y-1), 2}] \Phi_N^{m_t} x(t) \end{aligned}$$

This completes the proof.

One can prove that the mapping  $\times_{i=1}^N : \Delta_2^N \rightarrow \Delta_{2^n}$  is a bijective mapping. So Eq. (5) can adequately describe the dynamics of Eq. (3) at time  $t$ , say,  $L_{H_t}$  involves all of transition information of system from time  $t$  to  $t+1$  when set  $H_t$  is chosen for update, which is called network transition matrix under update elements  $H_t$ .

Whether RBNs or ARBNs, Network transition matrix is an essential concept for linear representation of logical dynamics based on STP technique. As deterministic system, there is only one network transition matrix for a particular RBN. However, multiple matrices exist in ARBNs depending on different asynchronous update scheme. For models discussed in this article,

there are total  $2^N - 1$  network transition matrices. By means of Theorem 1, when getting the structure matrix  $M_i$  of each Boolean logic  $f_i$  for any specific ARBN, all of network transition matrices could be calculated.

**Example 1.** A logical dynamics is described as

$$\begin{cases} x_1(t+1) = x_2(t) \vee x_3(t) \\ x_2(t+1) = x_1(t) \wedge x_3(t) \\ x_3(t+1) = x_2(t) \end{cases} \quad (6)$$

where  $x_i(t)$  represents the state of node  $i$  at time  $t$ , and the symbols “ $\wedge$ ”, “ $\vee$ ” represent conjunction and disjunction, respectively. By truth tables, one can get  $\vee \sim M_d = \delta_2[1,1,1,2]$  and  $\wedge \sim M_c = \delta_2[1,2,2,2]$ .

The algebraic form of Eq. (6) as

$$\begin{cases} x_1(t+1) = M_d x_2(t) x_3(t) \\ x_2(t+1) = M_c x_1(t) x_3(t) \\ x_3(t+1) = x_2(t) \end{cases} \quad (7)$$

Case 1: at time  $t$ , when only node 1 is selected for update,

$$\begin{aligned} x(t+1) &= x_1(t+1)x_2(t)x_3(t) \\ &= M_d x_2(t)x_3(t)x_2(t)x_3(t) \\ &= M_d W_{[2,4]} x_2(t)x_2(t)x_3(t)x_3(t) \\ &= M_d W_{[2,4]} M_r (I_2 \otimes M_r) x_2(t)x_3(t) \\ &= M_d W_{[2,4]} M_r (I_2 \otimes M_r) E_d x_1(t)x_2(t)x_3(t) \end{aligned}$$

Case 2: at time  $t$ , when all of three nodes are selected for update,

$$\begin{aligned} x(t+1) &= x_1(t+1)x_2(t+1)x_3(t+1) \\ &= M_d x_2(t)x_3(t) M_c x_1(t)x_3(t)x_2(t) \\ &= M_d (I_4 \otimes M_c) x_2(t)x_3(t)x_1(t)x_3(t)x_2(t) \\ &= M_d (I_4 \otimes M_c) W_{[2,4]} x_1(t)x_2(t) M_r x_3(t)x_2(t) \\ &= M_d (I_4 \otimes M_c) W_{[2,4]} (I_4 \otimes M_r) x_1(t)x_2(t)x_3(t)x_2(t) \\ &= M_d (I_4 \otimes M_c) W_{[2,4]} (I_4 \otimes M_r) (I_4 \otimes W_{[2]}) (I_2 \otimes M_r) \\ &\quad x_1(t)x_2(t)x_3(t) \end{aligned}$$

Here, we just show the calculations of network transition matrices under the above two cases. Similarly, all of the network transition matrices can be calculated and results are as follows.

$$\begin{cases} L_1 = M_d W_{[2,4]} M_r (I_2 \otimes M_r) E_d = \delta_8[1,2,3,8,1,2,3,8] \\ L_2 = (I_2 \otimes M_c) M_r (I_2 \otimes M_r) E_d W_{[2]} = \delta_8[1,4,1,4,7,8,7,8] \\ L_3 = (I_2 \otimes M_r) E_d W_{[4,2]} = \delta_8[1,4,1,4,5,8,5,8] \\ L_{1,2} = M_d (I_4 \otimes M_c) W_{[2,4]} (I_4 \otimes M_r^2) = \delta_8[1,4,1,8,3,4,3,8] \\ L_{1,3} = M_d W_{[4]} M_r^2 E_d = \delta_8[1,1,4,8,1,1,4,8] \\ L_{2,3} = (I_2 \otimes M_c) M_r (I_2 \otimes W_{[2]}) = \delta_8[1,3,2,4,7,7,8,8] \\ L_{1,2,3} = M_d (I_4 \otimes M_c) W_{[2,4]} (I_4 \otimes M_r) (I_4 \otimes W_{[2]}) (I_2 \otimes M_r) \\ = \delta_8[1,3,2,8,3,3,4,8] \end{cases}$$

And all of network transition matrices compose a set  $T = \{L_1, L_2, L_3, L_{1,2}, L_{1,3}, L_{2,3}, L_{1,2,3}\}$ .

It should be noted that matrix operations are involved in quantity during the above calculations. Fortunately, the related matrices are typically sparse and many efficient algorithms can be applied to optimize the calculation procedure.

### The asynchronous network transition table

In order to present the algorithm to find the attractors and basins of ARBNs, we first provide the following definition.

**Definition 2.** For a random Boolean network with  $N$  nodes under asynchronous stochastic update,  $T = \{A_1, A_2, \dots, A_{2^N-1}\}$  is the set of network transition matrices, where  $A_i = \delta_{2^N} [a_1^i, a_2^i, \dots, a_{2^N}^i]$ ,  $i \in \{1, \dots, 2^N - 1\}$ . The asynchronous network transition table, briefly called transition table, is defined as

$$I = \begin{pmatrix} a_1^1 & a_2^1 & \dots & a_{2^N}^1 \\ a_1^2 & a_2^2 & \dots & a_{2^N}^2 \\ \vdots & \vdots & \ddots & \vdots \\ a_1^{2^N-1} & a_2^{2^N-1} & \dots & a_{2^N}^{2^N-1} \end{pmatrix}. \quad (8)$$

**Example 2.** Recall Example 1. The transition table of system (7) is as follows

$$I = \begin{pmatrix} 1 & 2 & 3 & 8 & 1 & 2 & 3 & 8 \\ 1 & 4 & 1 & 4 & 7 & 8 & 7 & 8 \\ 1 & 4 & 1 & 4 & 5 & 8 & 5 & 8 \\ 1 & 4 & 1 & 8 & 3 & 4 & 3 & 8 \\ 1 & 1 & 4 & 8 & 1 & 1 & 4 & 8 \\ 1 & 3 & 2 & 4 & 7 & 7 & 8 & 8 \\ 1 & 3 & 2 & 8 & 3 & 3 & 4 & 8 \end{pmatrix}. \quad (9)$$

It should be noted that the asynchronous network transition table is specially designed for ARBNs. As mentioned above, more than one network transition matrix exists in ARBNs, which is different from RBNs. Therefore, there is a question: how to organize these network transition matrices can provide an efficient way for applications. For the above consideration, the asynchronous network transition table is presented, by means of which algorithms to find attractors and basins of ARBNs can be achieved completely based on set operations, which are shown later in Algorithms 1 and 2.

### Fixed Points (point attractors)

**Proposition 1.** For a given ARBN with  $N$  nodes, when  $\Gamma(I)_i = \{i\}$ , the state  $x = \delta_{2^N}^i$  is a fixed point, where  $I$  is the asynchronous network transition table.

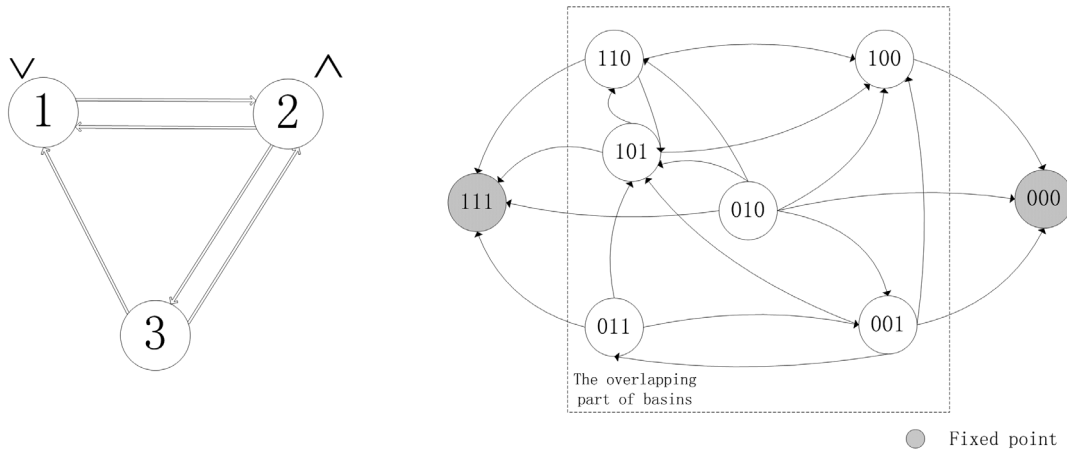
**Proof.** Assume  $T$  is the set of network transition matrices. When  $\Gamma(I)_i = \{i\}$ , we can obtain  $\Theta(Col(L)_i) = \{i\}, \forall L \in T$ , i.e. the  $i$ -th column of  $L$  is equal to  $\delta_{2^N}^i$ , which holds for any network transition matrices in  $T$ . And, it's easily verified that  $L \delta_{2^N}^i = \delta_{2^N}^i, \forall L \in T$ , which completes the proof.

Note that  $Col(\bullet)_i$  represents the  $i$ -th column of a matrix.

**Example 3.** Recall Example 1. We get  $\Gamma(I)_1 = \{1\}$  and  $\Gamma(I)_8 = \{8\}$ . So, two fixed points are  $\delta_8^1 \sim 111$  and  $\delta_8^8 \sim 000$ , which can be checked in the state space graph of Fig. 1.

### Attractors (Loose attractors)

Fixed points can be seen as point attractors. In the following, we focus on loose attractors in ARBNs.



**Figure 1. An asynchronous Boolean network with 3 nodes and its state space graph.**  
doi:10.1371/journal.pone.0066491.g001

**Theorem 2.** For a given ARBN with  $N$  nodes and  $I$  is the asynchronous network transition table, a state set of  $C = \{\delta_{2^N}^i, \delta_{2^N}^j, \dots, \delta_{2^N}^s\}$  ( $s \geq 2$ ) is an attractor of length  $s$ , iff

$$\Theta(C) = \Gamma(I)_{\Theta(C)}, \tag{10}$$

and

$$\Gamma(I)_{\Theta(C-C') \cap \Theta(C')} \neq \Phi, \quad \forall C' \subset C, \tag{11}$$

**Proof.** (Necessity) Assume  $T$  is the set of network transition matrices. When  $C = \{\delta_{2^N}^i, \delta_{2^N}^j, \dots, \delta_{2^N}^s\}$  is an attractor of length  $s$ , one can obtain  $L\delta_{2^N}^j \in C, \forall L \in T, \forall j \in \{1, 2, \dots, s\}$ , which implies  $\Gamma(I)_{\Theta(C)} \subseteq \Theta(C)$ . Since all of states in  $C$  can be reached, that is, for each element  $p$  in set  $\Theta(C)$ , one can find  $q \in \Theta(C)$  and  $L' \in T$  to satisfy  $L'\delta_{2^N}^q = \delta_{2^N}^p$ , i.e.  $p \in \Gamma(I)_{\Theta(C)}$ , which implies  $\Theta(C) \subseteq \Gamma(I)_{\Theta(C)}$ . So, one can obtain  $\Theta(C) = \Gamma(I)_{\Theta(C)}$  and Eq. (10) holds.

Assume there exists a set  $C' \subset C$ , which satisfy  $\Gamma(I)_{\Theta(C-C') \cap \Theta(C')} = \Phi$ . For  $\forall x \in C'$  and  $\forall x' \in C - C'$ , we can't find a path from  $x'$  to  $x$ , which is contradiction to the definition of attractor. So Eq. (11) holds.

(Sufficiency) Firstly, we should prove that each state in set  $C$  can find a way to reach all of the other states. Assume there exists a state  $\delta_{2^N}^p \in C$  and all of states that  $\delta_{2^N}^p$  can't reach compose a non-empty set  $C'$ . Since  $\delta_{2^N}^p$  can reach each state in set  $C - C'$ , so any states in set  $C - C'$  also can't reach states in  $C'$ , i.e.  $\Gamma(I)_{\Theta(C-C') \cap \Theta(C')} = \Phi$ , which is a contradiction to Eq. (11). So, each state in  $C$  can reach the other states.

Secondly, we should prove each state in set  $C$  can't reach any states out of set  $C$ .

We assume there exist  $\delta_{2^N}^q \in C$  and  $L' \in T$  which satisfy  $L'\delta_{2^N}^q = \delta_{2^N}^z \notin C$ . Then  $\alpha \in \Gamma(I)_{\Theta(C)}$  and  $\alpha \notin \Theta(C)$ , which is a contradiction to Eq. (10).

The proof is complete.

As a consequence, we develop a procedure to find all of attractors of ARBNs based on the above result, that is shown later in Algorithm 1.

### Basins of attractors

**Definition 3.** For ARBNs with  $N$  nodes,  $T$  and  $I$  are the set of network transition matrices and transition table, respectively. Let  $x, x' \in \Delta_{2^N}$  and  $L \in T$ , when  $Lx' = x$ ,  $x'$  is called the parent state of  $x$ . All of parent states of  $x$  are denoted by the set  $L^{-1}(x)$ . And, for a

set  $S \subseteq \Delta_{2^N}$ ,  $L^{-1}(S) = \bigcup_{x \in S} L^{-1}(x)$ . For simplicity,  $\underbrace{L^{-1}L^{-1} \dots L^{-1}}_t(x)$  is denoted by  $L^{-t}(x)$ .

For nondeterministic systems, a partition can't be found to separate the whole state space of ARBNs into several disjointed sets. As mentioned in Ref. [31], basins of attractors in asynchronous random Boolean networks involve two kinds: the definite basins and possible basins. Then, the overlapping parts of basins of different attractors as common phenomena exist the state space of ARBNs, which mean certain transient states can jump from one basin of attractors into another. More studies related to basins in ARBNs can refer to Refs. [32–34].

Generally, by a backstepping manner, one can find the basin of a given attractor  $C$ . Starting from the state set of attractor  $C$ , we can determine parent states of each state in  $C$ , which compose a new set  $C'$ . Repeat to find the parent states of each state in  $C'$  and add them into  $C'$  until there aren't any new states involved. The set  $C' \cup C$  is the basin of attractor  $C$ . The above procedure is shown in Algorithm 2.

**Example 4.** Recall Examples 2. The attractors of system (7) include the fixed points 000 and 111. For the fixed point  $\delta_8^1$ , by checking the transition table  $I$  in (9), we can get  $L^{-1}(\delta_8^1) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^5, \delta_8^6\}$ ,  $L^{-1}(\delta_8^1) \cup L^{-2}(\delta_8^1) = \{\delta_8^1, \delta_8^2, \delta_8^3, \delta_8^5, \delta_8^6, \delta_8^7\}$ . Since  $\bigcup_{i=1}^2 L^{-i}(\delta_8^1) = \bigcup_{i=1}^3 L^{-i}(\delta_8^1)$ , the basin of the fixed point  $\delta_8^1$  is composed of six states  $\delta_8^1 \sim 111, \delta_8^2 \sim 110, \delta_8^3 \sim 101, \delta_8^5 \sim 011, \delta_8^6 \sim 010, \delta_8^7 \sim 001$ . Similarly, for the fixed point  $\delta_8^8$ , one can get the basin  $\{\delta_8^2, \delta_8^3, \delta_8^4, \delta_8^5, \delta_8^6, \delta_8^7, \delta_8^8\}$ . The above results can be verified in Fig. 1.

In the above example, except state  $\delta_8^4$  is the definite basin of fixed point  $\delta_8^8$ , all the rest transient states are the overlapping part between the basins of two fixed points.

### Algorithms for detecting the attractors and basins of ARBNs

Given a specific logical dynamics under asynchronous update, all of the network transition matrices could be calculated based on Theorem 1. Consequently, the asynchronous network transition table can also be achieved. By means of the transition table, we formulate the following algorithms to find attractors and basins of ARBNs.

Algorithm 1: An algorithm for finding attractors in ARBNs with  $n$  nodes and the asynchronous network transition table  $I$ .

Begin: Algorithm 1

$\mathcal{F} = \{\delta_{2^n}^1, \dots, \delta_{2^n}^{2^n}\}, s = 1$  /\*  $\mathcal{F}$  is a feasible state set and  $s$  is the length of attractor. \*/

**for** ( $i = 1; i \leq 2^n; i++$ ) **do** /\*find the fixed points in ARBNs \*/  
**if**  $\Gamma(I)_i = \{i\}$  **then** /\*  $\delta_{2^n}^i$  is a fixed point \*/

    Find the basin of state  $\delta_{2^n}^i$ ; /\* the procedure to find basin of a particular attractor refers to Algorithm 2 \*/

    Remove the state  $\delta_{2^n}^i$  and its basin from set  $\mathcal{F}$ ;

**end if**

**end for**

**while** Nonempty ( $\mathcal{F}$ ) **do**

$s = s + 1$

$\mathcal{F}' = \{\delta_{2^n}^i \mid \text{card}(\Gamma(I)_i) \leq s \text{ and } \delta_{2^n}^i \in \mathcal{F}\}$  /\*  $\text{card}(\bullet)$  is the number of elements in set. \*/

    Update  $\mathcal{F}'$  by removing states whose successors aren't in  $\mathcal{F}'$ .

**while** Nonempty ( $\mathcal{F}'$ ) **do**

        Randomly pick up one state  $\delta_{2^n}^i$  from  $\mathcal{F}'$  and all reachable states from  $\delta_{2^n}^i$  compose a set  $C$ .

**if**  $\text{card}(C) = s$  and  $\Theta(C) = \Gamma(I)_{\Theta(C)}$  **then** /\*  $C$  is an attractor of length  $s$  \*/

            Find the basin of attractor  $C$  and remove the corresponding states from sets  $\mathcal{F}$  and  $\mathcal{F}'$ .

**else** remove state  $\delta_{2^n}^i$  from  $\mathcal{F}'$

**end if**

**end while**

**end while**

**End: Algorithm 1**

Since, according to the length, the attractors are found successively from small to large, so it isn't necessary to check the condition of Eq. (11) to determine attractors.

Algorithm 2: An algorithm for finding basin of a particular attractor  $C$  in ARBNs with  $n$  nodes.

Begin: Algorithm 2

$C' = C$  /\*  $C$  is the set of a particular attractor. \*/

**while**  $C' \neq L^{-1}(C')$  **do**

$$C' = C' \cup L^{-1}(C')$$

**end while**

**End: Algorithm 2**

In Ref. [39], Cheng's approach to find the attractors of length  $s$  in RBNs needs the computation of  $L^s$ , where  $L$  is the network transition matrix. Moreover, Cheng's algorithm is based on a prerequisite that RBNs are deterministic systems and a state on a cycle of length  $s$  will be back after  $s$  steps. However, this prerequisite can't hold for ARBNs. Therefore, we propose a novel way to detect attractors and basins of ARBNs by means of the transition table.

### Examples

In this section, some examples are presented to illustrate the main results of this article. Firstly, an idealized example is shown

as follows, which is a model of phosphorylation/dephosphorylation cycles considered by Gonze et. al [44] and re-investigated in [45].

**Example 5.** A logical dynamics is described as

$$\begin{cases} x_1(t+1) = x_2(t) \\ x_2(t+1) = x_3(t) \\ x_3(t+1) = x_4(t) \\ x_4(t+1) = x_1(t) \end{cases} \quad (12)$$

where  $x_i(t)$  represents the state of node  $i$  at time  $t$ . System (12) presents a model of protein-protein interactions. As shown in [45], there are three cycles of length four, one cycle of length two, and two fixed points under synchronous update. Here, we study the attractors and basins of system under asynchronous stochastic update.

The algebraic form of Eq. (12) as

$$\begin{cases} x_1(t+1) = E_d(I_2 \otimes E_d)(I_4 \otimes E_d)(I_2 \otimes W_{[2,4]})x(t) \\ x_2(t+1) = E_d(I_2 \otimes E_d)(I_4 \otimes E_d)(I_4 \otimes W_{[2]})x(t) \\ x_3(t+1) = E_d(I_2 \otimes E_d)(I_4 \otimes E_d)x(t) \\ x_4(t+1) = E_d(I_2 \otimes E_d)(I_4 \otimes E_d)W_{[2,8]}x(t) \end{cases} \quad (13)$$

where  $x(t) = \times_{i=1}^4 x_i(t)$ .

Then, the structure matrices of logical functions are as follows

$$\begin{cases} M_1 = E_d(I_2 \otimes E_d)(I_4 \otimes E_d)(I_2 \otimes W_{[2,4]}) = \\ \delta_2[1, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 2] \\ M_2 = E_d(I_2 \otimes E_d)(I_4 \otimes E_d)(I_4 \otimes W_{[2]}) = \\ \delta_2[1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2, 1, 1, 2, 2] \\ M_3 = E_d(I_2 \otimes E_d)(I_4 \otimes E_d) = \\ \delta_2[1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2, 1, 2] \\ M_4 = E_d(I_2 \otimes E_d)(I_4 \otimes E_d)W_{[2,8]} = \\ \delta_2[1, 1, 1, 1, 1, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2] \end{cases}$$

By Theorem 1, all of network transition matrices can be calculated and the asynchronous network transition table is constructed as follows.

I =	1	2	3	4	13	14	15	16	1	2	3	4	13	14	15	16
	1	2	7	8	1	2	7	8	9	10	15	16	9	10	15	16
	1	4	1	4	5	8	5	8	9	12	9	12	13	16	13	16
	1	1	3	3	5	5	7	7	10	10	12	12	14	14	16	16
	1	2	7	8	9	10	15	16	1	2	7	8	9	10	15	16
	1	4	1	4	13	16	13	16	1	4	1	4	13	16	13	16
	1	1	3	3	13	13	15	15	2	2	4	4	14	14	16	16
	1	4	5	8	1	4	5	8	9	12	13	16	9	12	13	16
	1	1	7	7	1	1	7	7	10	10	16	16	10	10	16	16
	1	3	1	3	5	7	5	7	10	12	10	12	14	16	14	16
	1	4	5	8	9	12	13	16	1	4	5	8	9	12	13	16
	1	1	7	7	9	9	15	15	2	2	8	8	10	10	16	16
	1	3	1	3	13	15	13	15	2	4	2	4	14	16	14	16
	1	3	5	7	1	3	5	7	10	12	14	16	10	12	14	16
	1	3	5	7	9	11	13	15	2	4	6	8	10	12	14	16

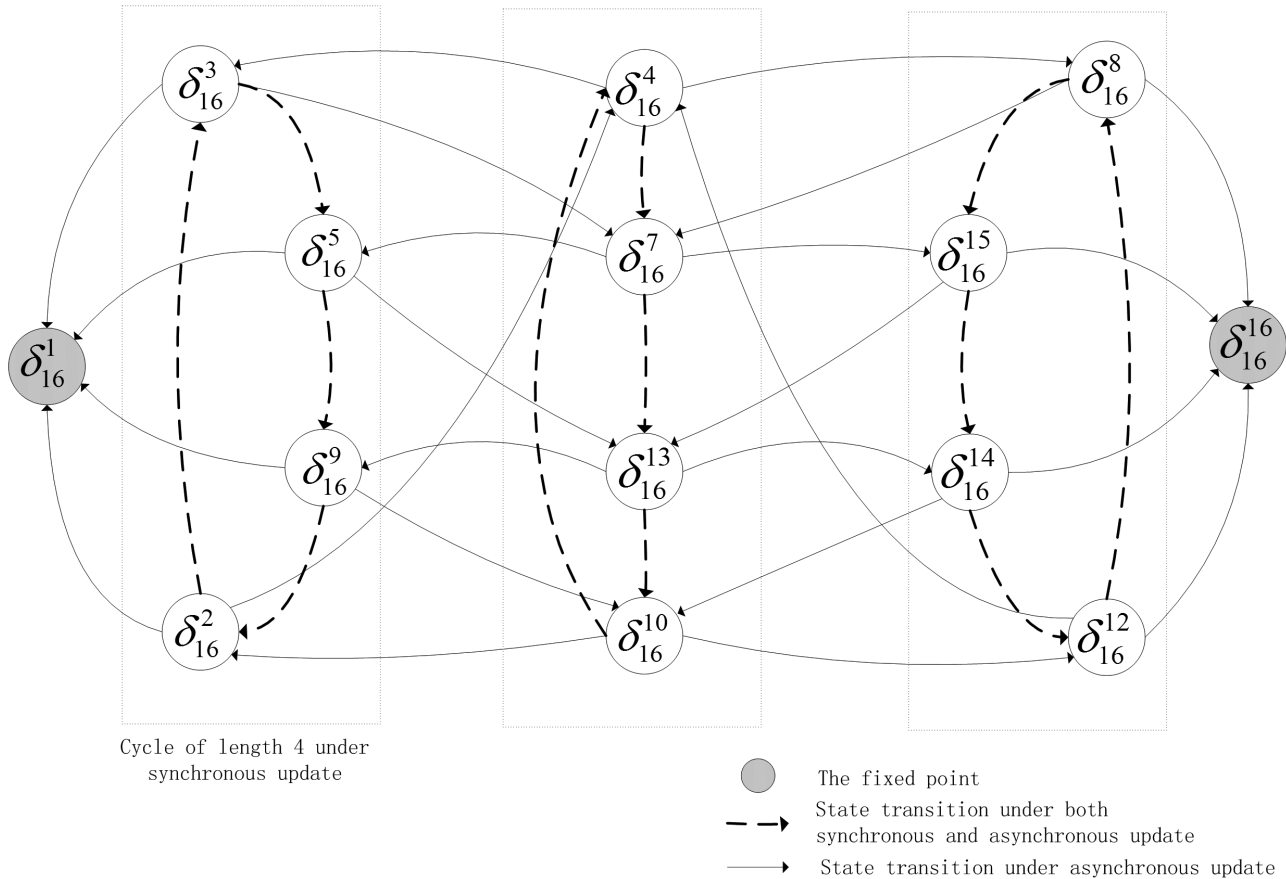
By means of Algorithm 1, since  $\Gamma(I)_1 = \{1\}$  and  $\Gamma(I)_{16} = \{16\}$  in the transition table I, it's easy to get two fixed points  $\{\delta_{16}^1 \sim 1111\}$  and  $\{\delta_{16}^{16} \sim 0000\}$ . Before finding the attractors of system, we first determine the basins of fixed points by Algorithm 2 to reduce the search scope in state space. For the fixed point  $C = \{\delta_{16}^1\}$ , one can get  $L^{-1}(C) = \{\delta_{16}^1, \delta_{16}^2, \delta_{16}^3, \delta_{16}^4, \delta_{16}^5, \delta_{16}^6, \delta_{16}^9, \delta_{16}^{11}\}$  and  $L^{-1}(C) \cup L^{-2}(C) = \{\delta_{16}^i | i = 1, \dots, 15\}$ . So far, we can know the basin of the fixed point  $\delta_{16}^1$  includes all of the states in state space except for the other fixed point  $\delta_{16}^{16}$ . Therefore, we can conclude that there aren't any attractors in this system. Similarly, the basin of the fixed point  $\delta_{16}^{16}$  involves all of the states except for  $\delta_{16}^1$ . A big overlapping part exists between the basins of two fixed points.

One can verify, under synchronous update, System (12) has two fixed points as  $\{\delta_{16}^1\}$  and  $\{\delta_{16}^{16}\}$ , three cycles of length four as  $\{\delta_{16}^2, \delta_{16}^3, \delta_{16}^5, \delta_{16}^9\}$ ,  $\{\delta_{16}^4, \delta_{16}^7, \delta_{16}^{13}, \delta_{16}^{10}\}$  and  $\{\delta_{16}^6, \delta_{16}^{11}\}$ . Without ambiguity, we call these cycles under synchronous update as "former cycles". When system (12) is under asynchronous update, all of former cycles turn into the overlapping part of the basins of two fixed points, say, they can jump from one basin of attractor into another. Furthermore, we also can observe some interesting phenomena. Each state in former cycle  $\{\delta_{16}^2, \delta_{16}^3, \delta_{16}^5, \delta_{16}^9\}$  has a direct link pointing to the fixed point  $\delta_{16}^1$ , at the same time, all of in-edges of the fixed point

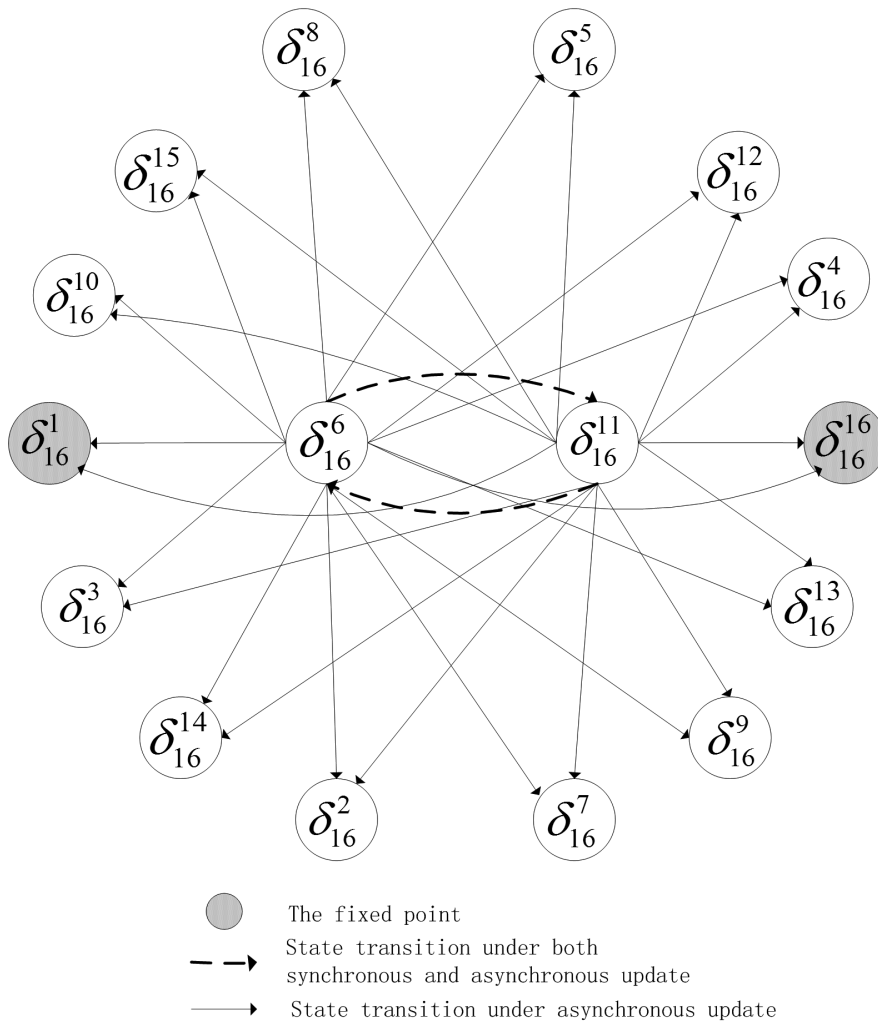
$\delta_{16}^1$  come from this former cycle. The same case occurs between the former cycle  $\{\delta_{16}^8, \delta_{16}^{15}, \delta_{16}^{14}, \delta_{16}^{12}\}$  and the fixed point  $\delta_{16}^{16}$ . As to the third former cycle  $\{\delta_{16}^4, \delta_{16}^7, \delta_{16}^{13}, \delta_{16}^{10}\}$ , it plays a role of bridge between the rest former cycles of length four, but it hasn't any direct links pointing to either of fixed points. The related state transfers are depicted in Fig. 2. Former cycle  $\{\delta_{16}^6, \delta_{16}^{11}\}$  is more special, each state of which has links pointing to all of states in state space except for itself, which is shown in Fig. 3.

Under synchronous update, all of attractors present similar dynamics, e.g. stability, and it's difficult to determine whether there is difference among them. However, some divergences among these attractors can be observed under asynchronous update. Such as the above example in Fig. 2, two former cycles of length four are close to the fixed points, which have greater possibility to flow into the corresponding fixed points. So, we can reasonably consider that the states of the two cycles would be faster into stable states. Yet, the remaining period-4 cycle is in the intermediate position, statistically, whose location determines it would take more time to converge into stability. As to the former period-2 cycle in Fig. 3, to some extent, its characteristic is more unstable under asynchronous update.

**Example 6.** Consider an ARBN with  $N=5, K=2$ , where a random number of nodes can be updated at each time step described as



**Figure 2. State transfers (Including two fixed points and the former cycles of length four).**  
doi:10.1371/journal.pone.0066491.g002



**Figure 3. State transfers (The basin of the former cycle of length two).**  
 doi:10.1371/journal.pone.0066491.g003

$$\begin{cases} x_1(t+1) = x_1(t) \wedge x_3(t) \\ x_2(t+1) = x_2(t) \rightarrow x_4(t) \\ x_3(t+1) = \neg(x_2(t) \wedge x_5(t)) \\ x_4(t+1) = x_3(t) \rightarrow x_4(t) \\ x_5(t+1) = x_1(t) \leftrightarrow x_5(t) \end{cases} \quad (14)$$

where  $\neg$ ,  $\rightarrow$  and  $\leftrightarrow$  represent the logical functions of negation, implication and equivalence, respectively. One can obtain  $\neg \sim M_n = \delta_2[1,2]$ ,  $\rightarrow \sim M_i = \delta_2[1,2,1,1]$  and  $\leftrightarrow \sim M_e = \delta_2[1,2,2,1]$ .

The algebraic form of Eq. (14) as

$$\begin{cases} x_1(t+1) = M_c E_d W_{[4,2]} E_d W_{[4,2]} E_d W_{[2]} x(t) \\ x_2(t+1) = M_i E_d W_{[4,2]} E_d W_{[2]} E_d x(t) \\ x_3(t+1) = M_n M_c E_d W_{[2]} E_d W_{[2]} E_d x(t) \\ x_4(t+1) = M_i E_d W_{[4,2]} E_d E_d x(t) \\ x_5(t+1) = M_i E_d W_{[2]} E_d W_{[2]} E_d W_{[2]} x(t) \end{cases} \quad (15)$$

where  $x(t) = \sum_{i=1}^5 x_i(t)$ .

Then, the structure matrices of logical functions are as follows.

$$\begin{cases} M_1 = M_c E_d W_{[4,2]} E_d W_{[4,2]} E_d W_{[2]} = \delta_2[1,1,1,1,2,2,2,2,1,1,1,1, \\ 2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2] \\ M_2 = M_i E_d W_{[4,2]} E_d W_{[2]} E_d = \delta_2[1,1,2,2,1,1,2,2,1,1,1,1,1,1,1, \\ 1,1,1,2,2,1,1,2,2,1,1,1,1,1,1,1] \\ M_3 = M_n M_c E_d W_{[2]} E_d W_{[2]} E_d = \delta_2[2,1,2,1,2,1,2,1,1,1,1,1,1,1, \\ 1,1,2,1,2,1,2,1,2,1,1,1,1,1,1,1] \\ M_4 = M_i E_d W_{[4,2]} E_d E_d = \delta_2[1,1,2,2,1,1,1,1,1,1,2,2,1,1,1,1, \\ 1,2,2,1,1,1,1,1,2,2,1,1,1,1] \\ M_5 = M_i E_d W_{[2]} E_d W_{[2]} E_d W_{[2]} = \delta_2[1,2,1,2,1,2,1,2,1,2,1,2,1,2, \\ 1,2,2,1,2,1,2,1,2,1,2,1,2,1,2,1] \end{cases}$$

By Theorem 1, all of network transition networks can be calculated and the asynchronous network transition table is as follows.



Construct the feasible set  $\mathcal{F} = \{\delta_{32}^i | i = 1, 2, \dots, 32\}$ . According to Algorithm 1, since  $\Gamma(\mathbf{I})_2 = \{2\}$ , the fixed point  $\delta_{32}^2$  can firstly be found. Consequently, the basin of this fixed point can be detected as  $\{\delta_{32}^6, \delta_{32}^8, \delta_{32}^{10}, \delta_{32}^{14}, \delta_{32}^{16}\}$  by Algorithm 2. Remove the fixed point and its basin from feasible set  $\mathcal{F}$ . Next, to find the attractors of length  $s=2$ , one could obtain  $\mathcal{F}' = \{\delta_{32}^1, \delta_{32}^4, \delta_{32}^5, \delta_{32}^9, \delta_{32}^{11}, \delta_{32}^{12}, \delta_{32}^{18}, \delta_{32}^{21}\}$ . Since  $\delta_{32}^{11} \rightarrow \delta_{32}^3, \delta_{32}^{18} \rightarrow \delta_{32}^{17}$  and  $\delta_{32}^9 \rightarrow \delta_{32}^1, \delta_{32}^{12} \rightarrow \delta_{32}^{21} \rightarrow \delta_{32}^{22}$ , these states should be removed from  $\mathcal{F}'$  because their successors aren't involved in set  $\mathcal{F}'$ . After updating, set  $\mathcal{F}' = \{\delta_{32}^4, \delta_{32}^{12}\}$ . Pick up state  $\delta_{32}^4$  and form the reachable set  $C_2 = \{\delta_{32}^4, \delta_{32}^{12}\}$ . It's easy to check  $\text{card}(C_2) = 2$  and  $\Theta(C_2) = \Gamma(\mathbf{I})_{\Theta(C_2)} = \{4, 12\}$ , so  $C_2 = \{\delta_{32}^4, \delta_{32}^{12}\}$  is an attractor of length 2. And, we can find the basin of  $C_2$  is  $\{\delta_{32}^8, \delta_{32}^{16}\}$ . Remove  $C_2$  and its basins from  $\mathcal{F}, \mathcal{F}'$  and continue to the above procedure. Similarly, we can find a loose attractor of length  $s=4$  as  $C_4 = \{\delta_{32}^{17}, \delta_{32}^{18}, \delta_{32}^{21}, \delta_{32}^{22}\}$  and its basin is  $\{\delta_{32}^i | i \in \{1, 3, 5, 6, 7, 8, 9, 11, 13, 14, 15, 16, 19, 20, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32\}\}$ . After removing the states  $C_4$  and its basin from  $\mathcal{F}$ , the feasible set  $\mathcal{F}$  is empty and the procedure is terminated. The state

as definite basin and four states  $\{\delta_{32}^6, \delta_{32}^8, \delta_{32}^{14}, \delta_{32}^{16}\}$  as possible basin. For attractor  $\{\delta_{32}^4, \delta_{32}^{12}\}$ , there are total two states  $\{\delta_{32}^8, \delta_{32}^{16}\}$  in its basin. Similarly, as to attractor  $\{\delta_{32}^{17}, \delta_{32}^{18}, \delta_{32}^{21}, \delta_{32}^{22}\}$ , one can find a definite basin  $\{\delta_{32}^1, \delta_{32}^5, \delta_{32}^9\}$  and the rest of transient states in its basin form the possible basin. For simplicity, here, we just show part of state transfers of states in attractors' possible basins.

**Example 7.** A Boolean model of cAMP signaling in Dictyostelium [45–47] as follows

$$\begin{cases} A(t+1) = 1 + C(t) \\ B(t+1) = 1 + A(t) \\ C(t+1) = H(t) + B(t) \cdot H(t) \\ D(t+1) = C(t) \\ E(t+1) = D(t) \\ F(t+1) = 1 + E(t) \\ G(t+1) = D(t) + E(t) + D(t) \cdot E(t) \\ H(t+1) = G(t) + F(t) \cdot G(t) \end{cases} \quad (16)$$

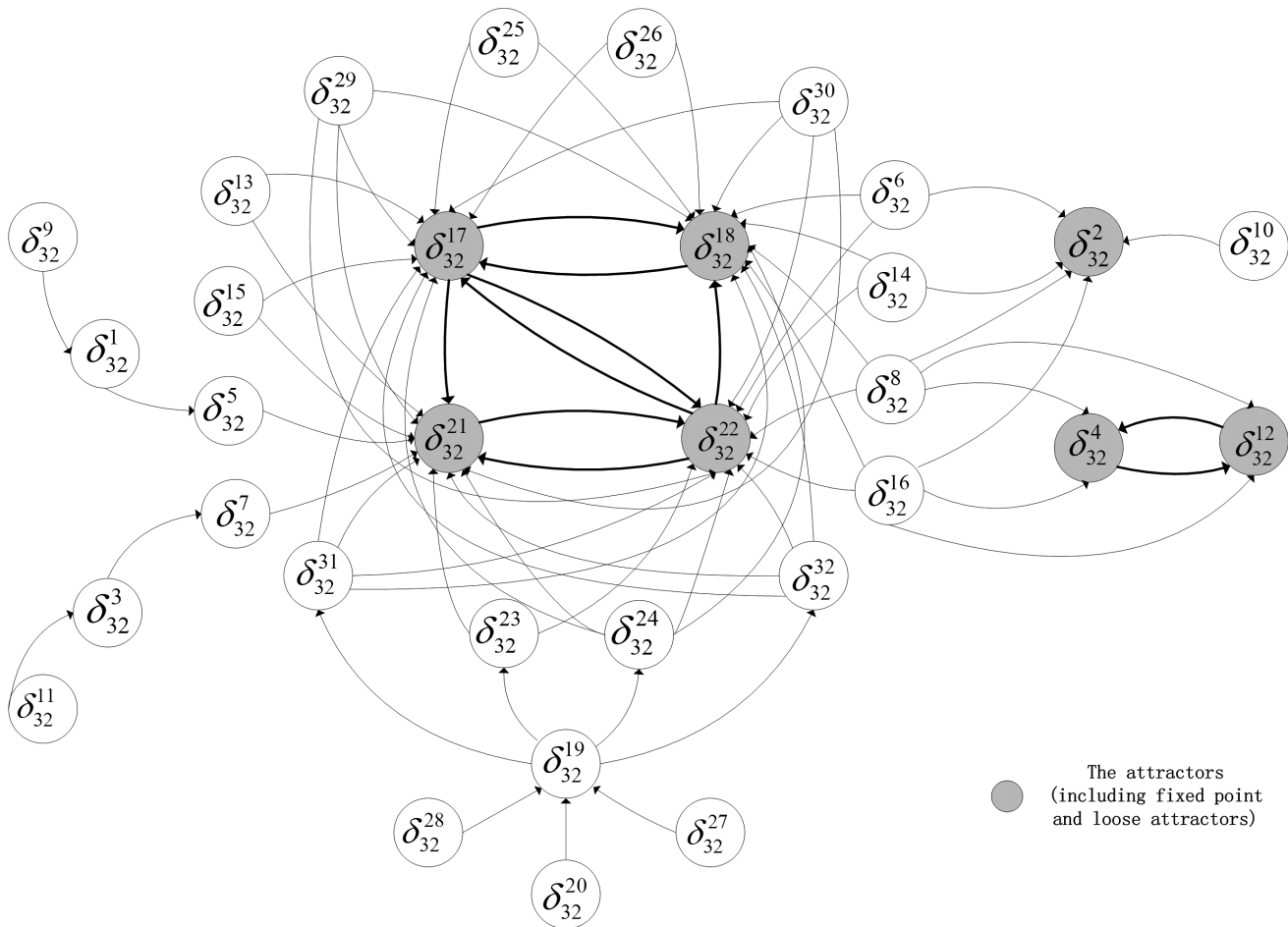
Where the symbols  $A, B, \dots, H$  represent PDI, PDE, cAMP(out-

I =	1	2	3	4	21	22	23	24	9	10	11	12	29	30	31	32	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
	1	2	11	12	5	6	15	16	1	2	3	4	5	6	7	8	17	18	27	28	21	22	31	32	17	18	19	20	21	22	23	24
	5	2	7	4	5	2	7	4	9	10	11	12	9	10	11	12	21	18	23	20	21	18	23	20	25	26	27	28	25	26	27	28
	1	2	3	4	5	6	5	6	9	10	11	12	13	14	13	14	17	18	19	20	21	22	21	22	25	26	27	28	29	30	29	30
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	18	17	20	19	22	21	24	23	26	25	28	27	30	29	32	31
	1	2	11	12	21	22	31	32	1	2	3	4	21	22	23	24	17	18	27	28	21	22	31	32	17	18	19	20	21	22	23	24
	5	2	7	4	21	18	23	20	9	10	11	12	25	26	27	28	21	18	23	20	21	18	23	20	25	26	27	28	25	26	27	28
	1	2	3	4	21	22	21	22	9	10	11	12	29	30	29	30	17	18	19	20	21	22	21	22	25	26	27	28	29	30	29	30
	1	2	3	4	21	22	23	24	9	10	11	12	29	30	31	32	18	17	20	19	22	21	24	23	26	25	28	27	30	29	32	31
	5	2	15	12	5	2	15	12	1	2	3	4	1	2	3	4	21	18	31	28	21	18	31	28	17	18	19	20	17	18	19	20
	1	2	11	12	5	6	13	14	1	2	3	4	5	6	5	6	17	18	27	28	21	22	29	30	17	18	19	20	21	22	21	22
	1	2	11	12	5	6	15	16	1	2	3	4	5	6	7	8	18	17	28	27	22	21	32	31	18	17	20	19	22	21	24	23
	5	2	7	4	5	2	5	2	9	10	11	12	9	10	9	10	21	18	23	20	21	18	21	18	25	26	27	28	25	26	25	26
	5	2	7	4	5	2	7	4	9	10	11	12	9	10	11	12	22	17	24	19	22	17	24	19	26	25	28	27	26	25	28	27
	1	2	3	4	5	6	5	6	9	10	11	12	13	14	13	14	18	17	20	19	22	21	22	21	26	25	28	27	30	29	30	29
	5	2	15	12	21	18	31	28	1	2	3	4	17	18	19	20	21	18	31	28	21	18	31	28	17	18	19	20	17	18	19	20
	1	2	11	12	21	22	29	30	1	2	3	4	21	22	21	22	17	18	27	28	21	22	29	30	17	18	19	20	21	22	21	22
	1	2	11	12	21	22	31	32	1	2	3	4	21	22	23	24	18	17	28	27	22	21	32	31	18	17	20	19	22	21	24	23
	5	2	7	4	21	18	21	18	9	10	11	12	25	26	25	26	21	18	23	20	21	18	21	18	25	26	27	28	25	26	25	26
	5	2	7	4	21	18	23	20	9	10	11	12	25	26	27	28	22	17	24	19	22	17	24	19	26	25	28	27	26	25	28	27
	1	2	3	4	21	22	21	22	9	10	11	12	29	30	29	30	18	17	20	19	22	21	22	21	26	25	28	27	30	29	30	29
	5	2	15	12	5	2	13	10	1	2	3	4	1	2	1	2	21	18	31	28	21	18	29	26	17	18	19	20	17	18	17	18
	5	2	15	12	5	2	15	12	1	2	3	4	1	2	3	4	22	17	32	27	22	17	32	27	18	17	20	19	18	17	20	19
	1	2	11	12	5	6	13	14	1	2	3	4	5	6	5	6	18	17	28	27	22	21	30	29	18	17	20	19	22	21	22	21
	5	2	7	4	5	2	5	2	9	10	11	12	9	10	9	10	22	17	24	19	22	17	22	17	26	25	28	27	26	25	26	25
	5	2	15	12	21	28	29	26	1	2	3	4	17	18	17	18	21	18	31	28	21	18	29	26	17	18	19	20	17	18	17	18
	5	2	15	12	21	18	31	28	1	2	3	4	17	18	19	20	22	17	32	27	22	17	32	27	18	17	20	19	18	17	20	19
	1	2	11	12	21	22	29	30	1	2	3	4	21	22	21	22	18	17	28	27	22	21	30	29	18	17	20	19	22	21	22	21
	5	2	7	4	21	18	21	18	9	10	11	12	25	26	25	26	22	17	24	19	22	17	22	17	26	25	28	27	26	25	26	25
	5	2	15	12	5	2	13	10	1	2	3	4	1	2	1	2	22	17	32	27	22	17	30	25	18	17	20	19	18	17	18	17
	5	2	15	12	21	18	29	26	1	2	3	4	17	18	17	18	22	17	32	27	22	17	30	25	18	17	20	19	18	17	18	17

transfer graph is shown in Fig. 4.

In Fig. 4, we can find one fixed point  $\delta_{32}^2$ , one attractor of length two  $C_2 = \{\delta_{32}^4, \delta_{32}^{12}\}$  and one attractor of length four  $C_4 = \{\delta_{32}^{17}, \delta_{32}^{18}, \delta_{32}^{21}, \delta_{32}^{22}\}$ . Correspondingly, the basin of each attractor is also depicted. As to fixed point  $\delta_{32}^2$ , there exist one state  $\delta_{32}^{10}$

side the cell), cAR1/3, Erk-2, RegA, ACA and cAMP (inside the cell), respectively.  $+ \sim M_p = \delta_2[2, 1, 1, 2]$  and  $\cdot \sim M_c = \delta_2[1, 2, 2, 2]$ . The above biochemical system is known for producing cyclic aggregation signals. In [45], researchers studied the attractors of system under synchronous update and found one fixed point and two cycles of length five. Here, we re-investigate this well-studied



**Figure 4. Attractors and basins of system (14).**  
doi:10.1371/journal.pone.0066491.g004

system under asynchronous update. Considering the interactions of the elements in the actual model, one element is randomly chosen for update at each time step.

The algebraic form of Eq. (16) is represented as

$$\begin{cases}
 A(t+1) = A_6(I_{128} \otimes M_n)(I_4 \otimes W_{[2,32]})\mathbf{K}(t) \\
 B(t+1) = A_6(I_{128} \otimes M_n)W_{[2,128]}\mathbf{K}(t) \\
 C(t+1) = A_5(I_{64} \otimes M_p)(I_{128} \otimes M_c)(I_2 \otimes W_{[2,64]})(I_{128} \otimes M_r)\mathbf{K}(t) \\
 D(t+1) = A_6(I_4 \otimes W_{[2,32]})\mathbf{K}(t) \\
 E(t+1) = A_6(I_8 \otimes W_{[2,16]})\mathbf{K}(t) \\
 F(t+1) = A_6(I_{128} \otimes M_n)(I_{16} \otimes W_{[2,8]})\mathbf{K}(t) \\
 G(t+1) = A_5(I_{64} \otimes M_p)(I_{128} \otimes M_p)(I_{256} \otimes M_c)(I_{128} \otimes W_{[2]} \\
 \quad (I_{64} \otimes M_r)(I_{128} \otimes M_r)(I_8 \otimes W_{[4,8]})\mathbf{K}(t) \\
 H(t+1) = A_5(I_{64} \otimes M_p)(I_{128} \otimes M_c)(I_{32} \otimes W_{[2,4]})(I_{128} \otimes M_r) \\
 \quad (I_{64} \otimes W_{[2]})\mathbf{K}(t)
 \end{cases} \tag{17}$$

where  $\Lambda_x = \left( \prod_{i=0}^x (I_{2^i} \otimes E_d) \right)$  and  $\mathbf{K}(t) = A(t)B(t)C(t)D(t)E(t)F(t)G(t)H(t)$ .

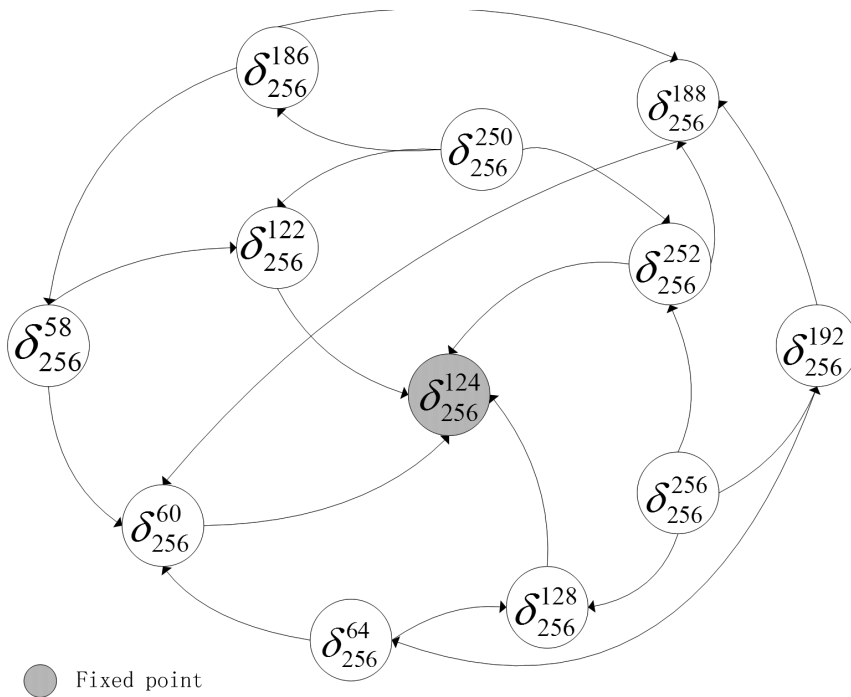
Based on the above discussions, one can calculate the asynchronous network transition table of system. The detailed computations are skipped and the results are presented as follows.

It can be verified that only one fixed point  $\delta_{256}^{124} \sim (10000100)$  exists in system (16) under asynchronous update, which is the same as the one under synchronous update. Fig. 5 and Fig. 6 show the basins of the fixed point under asynchronous and synchronous update, respectively.

Comparing Fig. 5 with Fig. 6, under asynchronous update, the basin of the fixed point contains fewer states but has more complex interconnections. There are 11 transient states in Fig. 5 and the farthest distance from one transient state to the fixed point is 3 steps, for instance,  $\delta_{256}^{186}(01000110) \rightarrow \delta_{256}^{58}(11000110) \rightarrow \delta_{256}^{60}(11000100) \rightarrow \delta_{256}^{124}(10000100)$ . However, in Fig. 6, there are more states directly flowing to the fixed point, furthermore, any states can reach the fixed point though at most 2 steps. There are no attractors in system (16) under asynchronous update, or from another perspective, the remaining states together construct a complicated and big loose attractor.

**Conclusions**

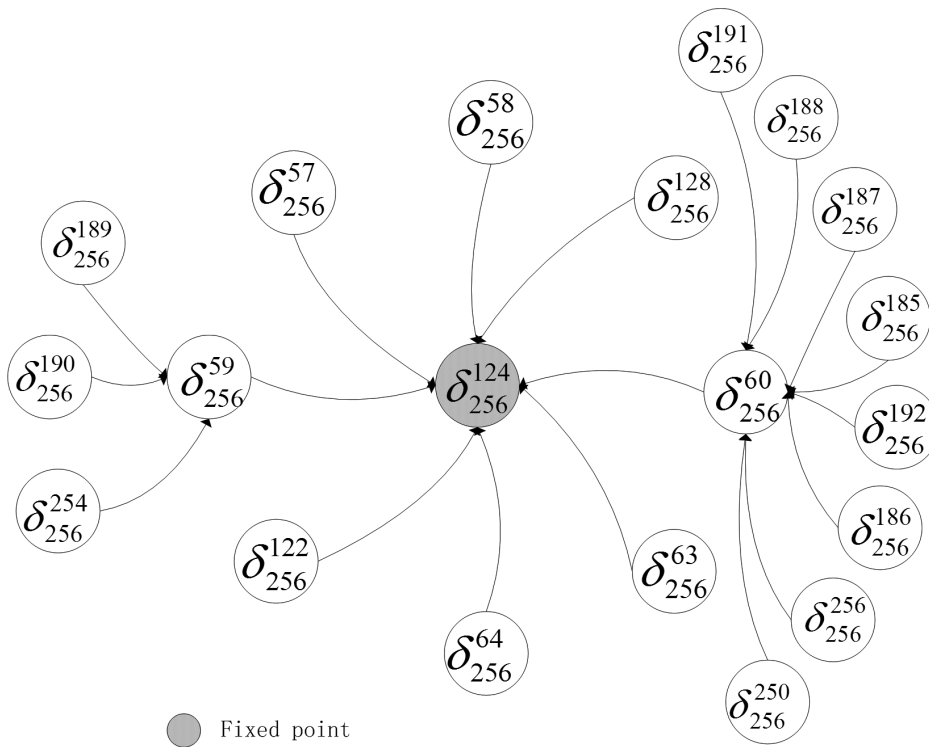
In this article, we have presented a new approach to study the dynamics of random Boolean networks under asynchronous stochastic update. By semi-tensor product of matrix, the logic of ARBN is converted into linear representation. A general formula



**Figure 5. The basin of the fixed point under asynchronous update.**  
doi:10.1371/journal.pone.0066491.g005

of network transition matrices is presented, by which one can obtain all of network transition matrices of a given logical dynamics whether under synchronous update or under asynchronous update. A necessary and sufficient algebraic criterion has

been proved to determine whether a given group of states compose attractor of length  $s$  on a particular ARBN. Asynchronous network transition table as a transformed set of network transition matrices is presented. Based on the above results, algorithms to detect the



**Figure 6. The basin of the fixed point under synchronous update.**  
doi:10.1371/journal.pone.0066491.g006

attractors and basins of ARBNs are achieved. Examples are shown to demonstrate the validity of results. As a preparation, results in this article can be applied in the investigation of Boolean control problems under asynchronous update, which would be the work in the future.

## References

- Kauffman SA (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. *J Theor Biol* 22:437–467.
- Drossel B (2008) Random Boolean Networks. In *Annual review of nonlinear dynamics and complexity*, vol. 1, New York: Wiley.
- Farrow C, Heidel J, Maloney J, Rogers J (2004) Scalar equations for synchronous Boolean networks with biological applications. *IEEE Trans. Neural Netw* 15:348–354.
- Steuer R (2007) Computational approaches to the topology, stability and dynamics of metabolic networks. *Phytochemistry* 68:2139–2151.
- Liu YJ, Chen CLP, Wen GX, Tong SC (2011) Adaptive neural output feedback tracking control for a class of uncertain discrete-time nonlinear systems. *IEEE Transactions on Neural Networks* 22:1162–1167.
- Zhang HG, Luo YH, Liu DR (2009) Neural-network-based near-optimal control for a class of discrete-time affine nonlinear systems with control constraints. *IEEE Transactions on Neural Networks* 20:1490–1503.
- Steuer R, Gross T, Selbig J, Blasius B (2006) Structural kinetic modeling of metabolic networks. *Proc Natl Acad Sci* 103:11868–11873.
- Fu J, Zhang HG, Ma TD, Zhang QL (2010) On passivity analysis for stochastic neural networks with interval time-varying delay. *Neurocomputing* 73:795–801.
- Sporns O, Chialvo DR, Kaiser M, Hilgetag CC (2004) Organization, development and function of complex brain networks. *Trends in Cognitive Sciences* 8:418–425.
- Liu YJ, Tong SC, Wang D, Li TS, Chen CLP (2011) Adaptive neural output feedback controller design with reduced-order observer for a class of uncertain nonlinear SISO systems. *IEEE Transactions on Neural Networks* 22:1328–1334.
- Chmiel AM, Sienkiewicz J, Suchecki K, Holyst JA (2007) Networks of companies and branches in Poland. *Physica A* 383: 134–138.
- Zhang HG, Wang YC (2008) Stability analysis of Markovian jumping stochastic Cohen-Grossberg neural networks with mixed time delays. *IEEE Transactions on Neural Networks* 19: 366–370.
- Liu YJ, Tong SC and Chen CLP (2013) Adaptive fuzzy control via observer design for uncertain nonlinear systems with unmodeled dynamics. *IEEE Transactions on Fuzzy Systems* 21:275–288.
- Colizza V, Barrat A, Barthelemy M, Vespignani A (2006) The role of the airline transportation network in the prediction and predictability of global epidemics. *Proc Natl Acad Sci* 103:2015–2020.
- Zhang HG, Ma TD, Huang GB, Wang ZL (2010) Robust global exponential synchronization of uncertain chaotic delayed neural networks via dual-stage impulsive control. *IEEE Transactions on System Man and Cybernetics Part B-Cybernetics* 40:831–844.
- Uchida S (2008) Stability and Structure of Model Food-webs with Adaptive Behavior. Ph.D. thesis, Technische Universität Darmstadt Dissertation, D17.
- Dubrova E, Teslenko M (2011) A SAT-Based Algorithm for Finding Attractors in Synchronous Boolean Networks. *IEEE-ACM Transactions on Computational Biology and Bioinformatics* 8:1393–1399.
- Dubrova E, Teslenko M (2005) Compositional properties of random Boolean networks. *Physical Review E* 71:056116.
- Lloyd-Price J, Gupta A, Ribeiro AS (2012) Robustness and information propagation in attractors of Random Boolean Networks. *PLOS ONE* 7:e42018.
- Akutsu T, Kosub S, Melkman AA (2012) Tamura T. Finding a Periodic Attractor of a Boolean Network. *IEEE-ACM Transactions on Computational Biology and Bioinformatics* 9:1410–1421.
- Yang LJ, Meng Y, Bao C, Liu WH, Ma CZ, et al. (2013) Robustness and Backbone Motif of a Cancer Network Regulated by miR-17-92 Cluster during the G1/S Transition. *PLOS ONE* 8:e57009.
- Kauffman SA (1993) *The Origins of Order: Self-Organization and Selection of Evolution*. Oxford: Oxford University Press.
- Klemm K, Bornholdt S (2005) Stable and unstable attractors in Boolean networks. *Physical Review E* 72:055101.
- Aracena J, Goles E, Moreira A (2009) On the robustness of update schedules in Boolean networks. *Biosystems* 97:1–8.
- Bassler KE, Lee C, Lee Y (2004) Evolution of Developmental Canalization in Networks of Competing Boolean Nodes. *Phys Rev Lett* 93:038101.
- Aldana M (2003) Boolean dynamics of networks with scale-free topology. *Physica D* 185:45–66.
- Paul U, Kaufman V, Drossel B (2006) Properties of attractors of canalizing random Boolean networks. *Physical Review E* 73: 026118.
- Krawitz P, Shmulevich I (2007) Basin Entropy in Boolean Network Ensembles. *Physical Review Letters* 98:158701.
- Hopfield JJ (1982) Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences* 79:2554–2558.
- Dellaert F, Beer R (1994) Toward an evolvable model of development for autonomous agent synthesis. *Proceedings of Alife 4*, pp.246–257, Cambridge MA, MIT Press.
- Harvey I, Bossomaier T (1997) Time out of joint: Attractors in asynchronous random boolean networks, in *Proceedings of the Fourth European Conference on Artificial Life (ECAL97*, MIT Press), pp. 67–75.
- Shreim A, Berdahl A, Greil F, Davidsen J, Paczuski M (2010) Attractor and basin entropies of random Boolean networks under asynchronous stochastic update. *Physical Review E* 82: 035102.
- Greil F, Drossel B (2005) The dynamics of critical Kauffman networks under asynchronous stochastic update. *Physical Review Letters* 95:048701.
- Deng XT, Geng HM, Matache MT (2007) Dynamics of asynchronous random Boolean networks with asynchrony generated by stochastic processes. *Biosystems* 88:16–34.
- Skodawessely T, Klemm K (2011) Finding attractors in asynchronous Boolean dynamics. *Advances in Complex Systems* 14:439–449.
- Greil F, Drossel B, Sattler J (2007) Critical Kauffman networks under deterministic asynchronous update. *New Journal of Physics* 9:373.
- Di Paolo EA (2001) Rhythmic and non-rhythmic attractors in asynchronous random Boolean networks. *BioSystems*, 59:185–195.
- Hallinan J, Wiles J (2004) Asynchronous dynamics of an artificial genetic regulatory network. In *Ninth International Conference on the Simulation and Synthesis of Living Systems*. ALife9, Boston.
- Cheng D, Qi H, Li Z (2011) Analysis and Control of Boolean Networks: A Semi-Tensor Product Approach. London, U.K.: Springer-Verlag.
- Cheng D, Qi H (2009) Controllability and observability of Boolean control networks. *Automatica* 45:1659–1667.
- Li F, Sun J (2011) Controllability of probabilistic Boolean control networks. *Automatica* 47:2765–2771.
- Li F, Sun J, Wu Q (2011) Observability of Boolean control networks with state time delays. *IEEE Trans. Neural Networks* 22: 948–954.
- Cheng D, Li ZQ, Qi H (2010) Realization of Boolean control networks. *Automatica* 46:62–69.
- Gonze D, Goldbeter A (2000) A model for a network of phosphorylation–dephosphorylation cycles displaying the dynamics of dominoes and clocks. *J Theor Biol* 210:167–186.
- Heidel J, Maloney J, Farrow C, Rogers JA (2003) Finding cycles in synchronous Boolean networks with applications to biochemical systems. *International Journal of Bifurcation and Chaos* 13:535–552.
- Goldbeter A (1996) *Biochemical Oscillations and Cellular Rhythms*. Cambridge Univ. Press, Cambridge.
- Aubry L, Firtel R (1999) Integration of signaling networks that regulate Dictyostelium differentiation. *Ann Rev Cell Dev Biol* 15:469–517.

## Author Contributions

Conceived and designed the experiments: CL XYW. Performed the experiments: CL. Analyzed the data: CL XYW. Contributed reagents/materials/analysis tools: CL XYW. Wrote the paper: CL XYW.