RESEARCH ARTICLE

An Algorithm for Finding the Singleton Attractors and Pre-Images in Strong-Inhibition Boolean Networks

Zhiwei He[1], Meng Zhan[2], Shuai Liu[3], Zebo Fang[4], Chenggui Yao[1]*

1 Department of Mathematics, Shaoxing University, Shaoxing, China, **2** State Key Laboratory of Advanced Electromagnetic Engineering and Technology, School of Electrical and Electronic Engineering, Huazhong University of Science and Technology, Wuhan, China, **3** College of Science, Northwest A&F University, Yangling, China, **4** Department of Physics, Shaoxing University, Shaoxing, China

* yaochenggui@usx.edu.cn

Citation: He Z, Zhan M, Liu S, Fang Z, Yao C (2016) An Algorithm for Finding the Singleton Attractors and Pre-Images in Strong-Inhibition Boolean Networks. PLoS ONE 11(11): e0166906. doi:10.1371/journal.pone.0166906

Editor: Peter Csermely, Semmelweis University, HUNGARY

Received: August 18, 2016

Accepted: November 4, 2016

Published: November 18, 2016

Data Availability Statement: All relevant data are within the paper. The code of the program can be seen on GitHub, and the URL is: https://github.com/zhiweihe2016/codAndDatas.

Funding: This work was supported partially by Natural Science Foundation of Zhejiang Province under Grant Nos. LY16A050001 (CY) and LY15A04001 (ZF), the National Natural Science Foundation of China under Grant Nos. 11675112 (CY), 11475253 (MZ), 11605142 (SL) and 51272159 (ZF). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Abstract

The detection of the singleton attractors is of great significance for the systematic study of genetic regulatory network. In this paper, we design an algorithm to compute the singleton attractors and pre-images of the strong-inhibition Boolean networks which is a biophysically plausible gene model. Our algorithm can not only identify accurately the singleton attractors, but also find easily the pre-images of the network. Based on extensive computational experiments, we show that the computational time of the algorithm is proportional to the number of the singleton attractors, which indicates the algorithm has much advantage in finding the singleton attractors for the networks with high average degree and less inhibitory interactions. Our algorithm may shed light on understanding the function and structure of the strong-inhibition Boolean networks.

Introduction

Revealing how a genetic regulatory network is organized for its function is one of the main topics in system biology [1–3]. With the contribution of experimental physiologists, many precise details of gene interactions as well as their functions have been revealed [3]. Based on the experimental data, different kinds of mathematical models have been developed, such as master equations [4], Monte-Carlo method [5, 6], stochastic model [7, 8], ordinary differential equations (ODE) [9–13], Boolean network method [14–16] and other mathematical models [17]. Among all the models, the Boolean network is a simple but powerful mathematical model. It was originally introduced by Kauffman [14] and has been quickly developed into many different types, including the general Boolean network model [15], the AND/OR Boolean network model [16], and the strong-inhibition Boolean network model [18], and so on. In the general Boolean network model, the most details of the network are taken into account. It can perfectly yield insight to global behavior of the network, however, it is difficult to analysis the general Boolean network due to the complexity of large genetic regulatory network. Further, a much simpler AND/OR Boolean network model where many details of the network are

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

neglected is presented. Recently, a strong-inhibition Boolean network model which is a biophysically plausible gene model has been obtained from the transition of the general Boolean network model, as the inhibitory interactions often dominate the activations in some important genomic regulations [18]. The model has been successfully used to reveal the backbone motif structure of the cell-cycle network and reconstruct the network structure from evolution data [18–20], since the model has more details of the network, and a simple rule which is similar to the AND/OR Boolean network model. It is very suitable to be applied to study the genetic regulatory network.

To capture the biological meaning of the genetic regulatory network, it is very necessary to identify the stable states of the dynamic system, such as the cyclic attractor and the singleton attractor. Two different attractors correspond to different functional states of the network. As an example, the cyclic and singleton attractor correspond to cell growth and differentiated states (or apoptosis) in the cell-cycle network, respectively [21]. It is noteworthy that the identification of the attractors is of great importance and is very useful in many applications, such as the treatment of human cancers [22, 23], genetic engineering [24] and validate hypotheses on the transcription processes [25]. In order to identify all attractors of the genetic regulatory network, so far several methods have been proposed, including the methods relying on binary decision diagrams [26, 27], constraint programming [28], feedback vertex sets [29, 30], linear mapping [31]. Moreover, many of these methods have been developed to be more general and effective [32–34].

In particular, the identification of the singleton attractor is especially important for revealing the function of the genetic regulatory network [30]. To find the singleton attractors of the genetic regulatory network in the context of a Boolean network, several algorithms have been proposed. For instance, Leone et al. firstly applied the SAT (the satisfiability problem of Boolean formulas in conjunctive normal form) algorithms to identify the singleton attractor in a Boolean network [35]. Based on this observation, Tamura and Akutsu showed that the detection problem of singleton attractor for a Boolean network with maximum in-degree k can be reduced to $(k + 1)$-SAT, and presented an algorithm for detecting the singleton attractor of an AND/OR Boolean network in $O(1.787^N)$ time [36]. Subsequently, the authors and coworkers succeeded in improving the above algorithm, including proposed an $O(1.587^N)$ time algorithm for determining the singleton attractor of an AND/OR Boolean network [37], an $O(1.757^N)$ time algorithm for determining the singleton attractor of planar and nonplanar Boolean networks [38], the $O(1.799^N)$ and $O(1.619^N)$ time algorithm for determining the singleton attractor of Boolean networks with nested canalyzing functions and chain functions [39], respectively. In the $O(1.799^N)$ time algorithm, it was implicitly assumed that the network does not contain the positive self-loop. While allowing for the presence of positive self-loop, the singleton attractor detection problem was solved in $O(1.871^N)$ time [40]. Besides, Zhang et al. developed a quite fast algorithm using gene ordering [30]. It is shown that the algorithm can identify all singleton attractors for a random Boolean network with maximum indegree two with an average time $O(1.19^N)$. However, the average computational time complexity would increase (approximately $O(1.5^N)$ for maximum indegree ten) with increasing of maximum indegree. Recently, Zou proposed an algorithm by dividing the polynomial equation system into many subsystems [41]. Indeed, the problem of detection for the singleton attractor is still NP-hard [29, 30, 42]. Thus, it is not plausible to solve this problem efficiently in all cases. However, we can develop a new algorithm that is fast and can be applied in different mathematical models.

In this paper, we focus our attention on the strong-inhibition Boolean network model, and propose an algorithm for detecting the singleton attractors and pre-images of genetic regulatory network. Our algorithm is applied to the cell-cycle network of budding yeast, and can accurately identify all the singleton attractors of the network. Furthermore, we show that the average computational time increases exponentially with the growth of the network size N, and the order is approximately $O(1.34^N)$. Moreover, we discover that the computational time is proportional to the number of the singleton attractors of network. Based on this observation, we find that our algorithm has much advantage in finding the singleton attractors for the networks with high average degree and less inhibitory interactions. We further extend the algorithm for studying the pre-image problem. The pre-image problem (or predecessor problem), to find the set of all possible inputs that evolve into the given output, has been addressed by Wuensche in [43]. In recent years, several algorithms have been introduced, such as reverse algorithm [44], probabilistic algorithm [45]. But it has also been shown to be NP-hard in general [46]. In this paper, we extend the previous singleton attractor detection algorithm to find the pre-images of any given state. All pre-images and even the basin of one singleton attractor are successfully and accurately found.

The paper will be arranged as follows: in Sec. II, the state transition model is introduced. In Sec. III, the algorithm for finding the singleton attractor is given. In Sec. IV, we show an example for finding the singleton attractor with this algorithm, and the computational time of the algorithm for finding singleton attractors is analyzed. In Sec. V, we present the algorithm of finding the pre-images of any target network state. Finally, we give a summary in Sec VI.

State transition model

Boolean network model is a discrete dynamical model of genetic regulatory networks. In this model, each node has only two states, 1 or 0, representing "on" (active) or "off" (inactive) state, respectively. For a network system with N components, the network state at time t is denoted by $S(t) = (S_1(t), S_2(t), \ldots, S_N(t))$. The network state in the next step is uniquely determined by the following rule [15]:

$$S_i(t+1) = \begin{cases} 1, & \text{if } \sum_{j=1}^{N} a_{ij}S_j(t) > 0, \\ 0, & \text{if } \sum_{j=1}^{N} a_{ij}S_j(t) < 0, \\ S_i(t), & \text{if } \sum_{j=1}^{N} a_{ij}S_j(t) = 0, \end{cases} \tag{1}$$

where $i, j = 1, 2, \ldots, N$. $A = \{a_{ij}, i, j = 1, 2, \ldots, N\}$ is the adjacency matrix of the network, which denotes the interactions between all the components. a_{ij} can be positive, negative, or zero, standing for an activating interaction, inhibiting interaction or no interaction, respectively. Usually, a_{ii} take −1, 1, or 0, and a_{ij} take $-\gamma$, 1, or 0 for $j \neq i$.

In fact, the inhibiting interactions are dominant for most biomolecular interactions, one prefers $\gamma \geq 1$. Following the "dominant inhibition" assumption $\gamma = \infty$, which represents some typical biological transcriptional regulatory processes [47–49], the Eq (1) can be simplified into a logical equation [18]

$$S_i(t+1) = (\sum_{j\neq i}(S_j(t)g_{ij}) + S_i(t)\overline{r}_{ii} + \overline{S_i(t)}g_{ii})\prod_{j\neq i}(\overline{S_j(t)r_{ij}}), \tag{2}$$

where r_{ij} and g_{ij} represent the putative inhibitory and putative stimulatory edge from node j to i, respectively. The relation between a_{ij} and g_{ij}, r_{ij} is

$$\begin{cases} g_{ij} = 1, \ r_{ij} = 0, & \text{if } a_{ij} = 1, \\ g_{ij} = 0, \ r_{ij} = 1, & \text{if } a_{ij} = -\gamma \ or \ -1, \\ g_{ij} = 0, \ r_{ij} = 0, & \text{if } a_{ij} = 0, \end{cases} \tag{3}$$

where $i, j = 1, 2, \ldots, N$. It should be noted that node j can't have activating and inhibiting effect on node i at the same time, namely g_{ij} and r_{ij} can't both take the value 1 for any fixed i and j. The addition, multiplication and bar in the Eq (2) represent the Boolean operator OR, AND, and NOT, respectively. This model is called strong-inhibition Boolean model due to the "dominant inhibition" assumption.

The algorithm for finding the singleton attractor

To identify the singleton attractor which the network system will eventually evolve into a limited set of stable states, we should check all the states in the network state space. However, a great deal of time will be assumed for the large networks under this enumeration-based algorithm since the state space consists of 2^N different states. So it is very necessary to design more efficient method to identify the singleton attractor.

For the strong-inhibition Boolean model, the singleton attractors are solutions of the following equations

$$S_i(t) = (\sum_{j\neq i}(S_j(t)g_{ij}) + S_i(t)\overline{r}_{ii} + \overline{S_i(t)}g_{ii})\prod_{j\neq i}(\overline{S_j(t)r_{ij}}), \tag{4}$$

where $i, j = 1, 2, \ldots, N$. We can obtain a concise equation by setting $S_i = S_i(t)$,

$$S_i = (\sum_{j\neq i}(S_j g_{ij}) + S_i\overline{r}_{ii} + \overline{S}_i g_{ii})\prod_{j\neq i}(\overline{S_j r_{ij}}), \quad for\ i = 1, 2, \ldots, N. \tag{5}$$

Among all the interactions that regulate node i, we suppose that there are h positive interactions ($g_{ij_1} = g_{ij_2} = \ldots = g_{ij_h} = 1$) and m negative interactions ($r_{ik_1} = r_{ik_2} = \ldots = r_{ik_m} = 1$). Then Eq (5) can be written as:

$$S_i = \overline{S_{k_1}} \wedge \overline{S_{k_2}} \wedge \cdots \wedge \overline{S_{k_m}} \wedge (S_{j_1} \vee S_{j_2} \cdots S_{j_h} \vee X_i), \tag{6}$$

where $\wedge$ and $\vee$ represent AND and OR, respectively. And $X_i = 1$ if $g_{ii} = 1$; $X_i = 0$ if $r_{ii} = 1$; $X_i = S_i$ if $g_{ii} = r_{ii} = 0$. According to the definition in Ref. [40], one can see that the right side of Eq (6) is a chain function, which is a special case of nested canalyzing function (nc-function). It was proved by Akutsu et al. that finding a singleton attractor for an nc-network with chain functions remains NP-hard [39].

Then, some rules are gotten by analysing Eq (5),

rule 1. if $S_i = 1$ and $r_{ji} = 1$, then $S_j = 0$;

rule 2. if $S_i = 1$ and $r_{ij} = 1$, then $S_j = 0$;

rule 3. if $S_i = 1$, $r_{ii} = 1$, $g_{ij_0} = 1$ and $\Sigma_{j\neq j_0}(S_j\, g_{ij}) = 0$, then $S_{j_0} = 1$;

rule 4. if $S_i = 0$ and $\Sigma_{j\neq i}(S_j\, r_{ij}) = 0$, $g_{ij_0} = 1$ and $\Sigma_{j\neq j_0}(S_j\, g_{ij}) = 0$, then $S_{j_0} = 0$;

rule 5. if $r_{ii} = 1$ and $\Sigma_{j\neq i}(S_j\, g_{ij}) = 0$, then $S_i = 0$;

rule 6. if $g_{ii} = 1$ and $\Sigma_{j\neq i}(S_j\, r_{ij}) = 0$, then $S_i = 1$;

rule 7. if $\Sigma_{j\neq i}(S_j\, r_{ij}) = 0$, and there is a node j_0 such that $S_{j_0}\, g_{ij_0} = 1$, then $S_i = 1$.

According to these rules and the given network structure, the states of other nodes can be determined if we already know the states of part of nodes. The states of its neighbors may be determined with the rules 1 – 4 if the state of a node is known; its state may be determined with the rules 5 – 7 if the state of a node is unknown. What's more, according to the rules 1 and 2, we can find that if $S_i = 1$ and the node i has more putative inhibitory connections, the states of its neighbors can be easily determined. Based on these rules, we can design the following algorithm to identify the singleton attractors.

Step 1. Input the network matrix A, and a network state $S = (S_1, S_2, \ldots, S_N)$, where the states of l_0 $(0 < l_0 \leq N)$ nodes are unknown while the states of other $N - l_0$ nodes are known. Those l_0 nodes are variable nodes as their states can be 1 or 0.

Step 2. Find the node i_0 which has the most putative inhibitory connections among the variable nodes of the network.

Step 3. Let $S_{old}^1 = S$ and assign $S_{i_0} = 1$. According to the rules $1 - 7$, we can determine the states of more variable nodes, and obtain new network state S_{new}. Throughout this process, a case may occur: according to one rule, the state of a variable node can be determined as 1, but it is determined as 0 with another rules. This contradiction means that the assignment of $S_{i_0} = 1$ is not appropriate, and the corresponding network state is not a singleton attractor of the network system, so we should remove it. On the other hand, if this case does not occur, we should remember the state S_{new} and count the number of its variable nodes l_{new}. Moreover, if $l_{new} > 0$, we should remember this state and wait the next turn to begin **Step 1**; if $l_{new} = 0$, which means that the states of all nodes are determined and a new singleton attractor is found, return this singleton attractor.

Step 4. Let $S_{old}^0 = S$ and assign $S_{i_0} = 0$. Do the same as those did in **Step 3**.

Next, we will prove that the states found with our algorithm are exactly all the singleton attractors of the network. In our algorithm, we determine the states of variable nodes according to the rules and remove the state if contradiction appears. Actually, the states found with our algorithm are states that do not contradict with the rules, here we use set U to denote them. And we use set V to represent all the singleton attractors of the network. Because a singleton attractor must satisfy Eq (5), so it obeys the rules, thus V is a subset of U, namely $V \subset U$. Afterwards, we will prove that U is also a subset of V.

Suppose that U is not a subset of V, there must be a state S^0 such that $S^0 \in U$ and $S^0 \notin V$. Accordingly, the state S^0 obeys all the rules but does not satisfy Eq (5). Then, there must be a node's state, assuming S_i^0, such that

$$S_i^0 \neq (\sum_{j\neq i}(S_j^0 g_{ij}) + S_i^0 \overline{r}_{ii} + \overline{S_i^0} g_{ii}) \prod_{j\neq i}(\overline{S_j^0 r_{ij}}), \tag{7}$$

where $S_i^0 = 0$ or 1.

Firstly, assuming $S_i^0 = 0$ and inserting it into Eq (7), we obtain

$$(\sum_{j\neq i}(S_j^0 g_{ij}) + g_{ii}) \prod_{j\neq i}(\overline{S_j^0 r_{ij}}) = 1. \tag{8}$$

Therefore, we get

$$\prod_{j\neq i}(\overline{S_j^0 r_{ij}}) = 1, \tag{9}$$

$$(\sum_{j\neq i}(S_j^0 g_{ij}) + g_{ii}) = 1. \tag{10}$$

The Eq (9) indicates

$$\sum_{j\neq i}(S_j r_{ij}) = 0. \tag{11}$$

And from Eq (10), we obtain

$$g_{ii} = 1 \quad or \quad \sum_{j\neq i}(S_j^0 g_{ij}) = 1. \tag{12}$$

Obviously, it contradicts with the rule 6 through Eq (11) and $S_i^0 = 0$ if $g_{ii} = 1$. And if $\sum_{j\neq i}(S_j^0 g_{ij}) = 1$, then there must be a node j_0 such that $S_{j_0}^0 g_{ij_0} = 1$. So we can get

$$S_{j_0}^0 = 1 \quad and \quad g_{ij_0} = 1. \tag{13}$$

Through Eqs (11) and (13) and $S_i^0 = 0$, one can see that it contradicts with the rules 4 and 7. Therefore, it is impossible that S^0 which obeys all the rules does not satisfy the Eq (5), if $S_i^0 = 0$.

Then, we assume $S_i^0 = 1$. Inserting it into Eq (7), we obtain

$$(\sum_{j\neq i}(S_j^0 g_{ij}) + \overline{r}_{ii})\prod_{j\neq i}(\overline{S_j^0 r_{ij}}) = 0. \tag{14}$$

Further, we get

$$\sum_{j\neq i}(S_j^0 g_{ij}) + \overline{r}_{ii} = 0 \quad or \quad \prod_{j\neq i}(\overline{S_j^0 r_{ij}}) = 0. \tag{15}$$

If $\sum_{j\neq i}(S_j^0 g_{ij}) + \overline{r}_{ii} = 0$, then $\sum_{j\neq i}(S_j^0 g_{ij}) = 0$ and $r_{ii} = 1$. It contradicts with the rule 5 since $S_i^0 = 1$. If $\prod_{j\neq i}(\overline{S_j^0 r_{ij}}) = 0$, then there must be a node j_0 such that

$$S_{j_0}^0 = 1 \ and \ r_{ij_0} = 1. \tag{16}$$

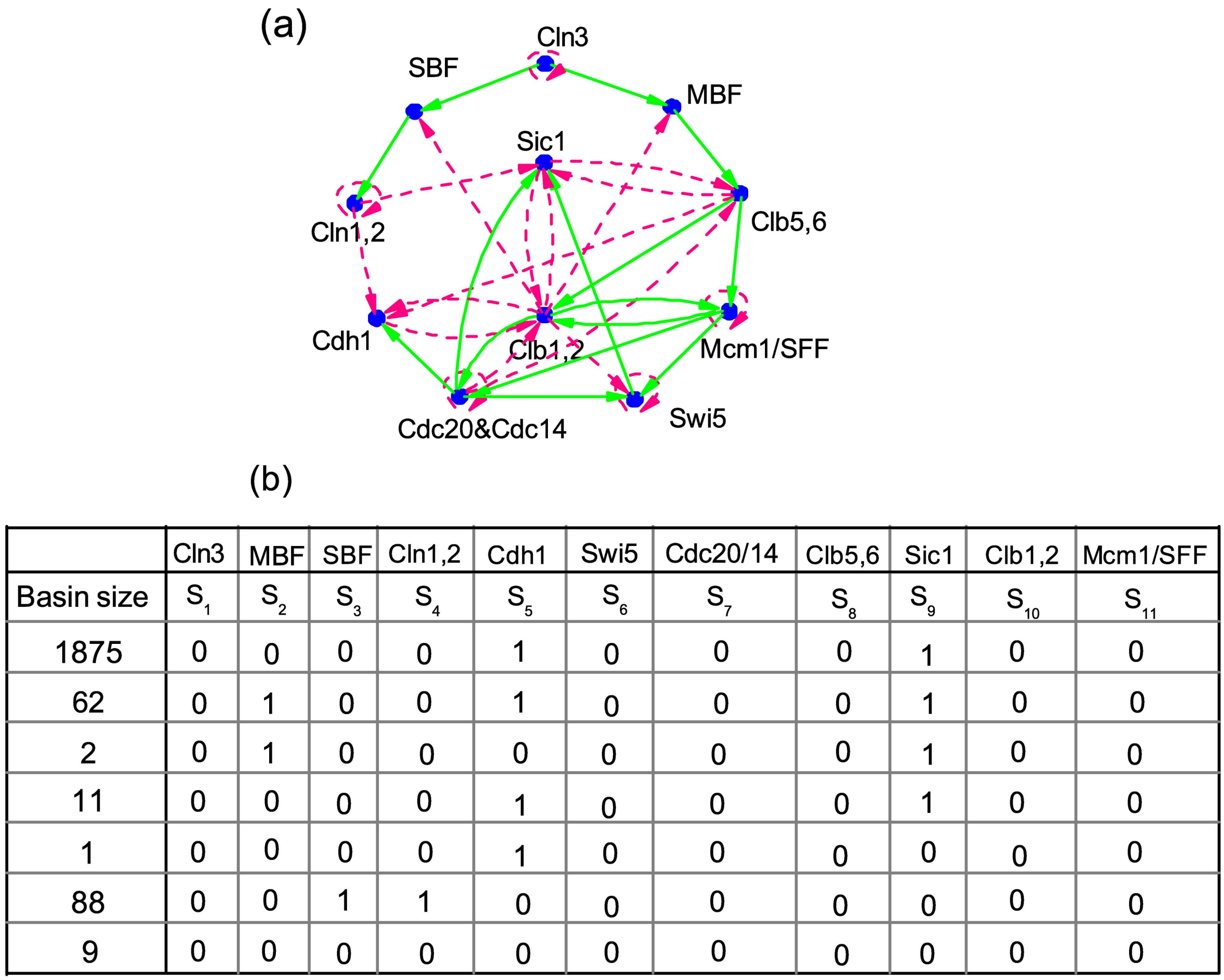
Through Eq (16) and $S_i^0 = 1$, we find that it contradicts with the rules 1 and 2. These results show that if $S_i^0 = 1$, it is also impossible that S^0 which obeys all the rules does not satisfy Eq (5).

Now we can conclude that the state S^0, which obeys all the rules but dissatisfies Eq (5), does not exist. It indicates that U is a subset of V, namely $U \subset V$. Therefore, $U = V$ is proved. And we get the proof that the states found with our algorithm are exactly all the singleton attractors of the network. The code of this algorithm is provided in Supporting Information (S1 File).

An example for finding the singleton attractor

To verify the validity of the above algorithm, we apply it to detect the singleton attractors of the cell-cycle network for budding yeast which is well studied in the research of cell-cycle process [15]. As shown in Fig 1(a), the network consists of 11 nodes and 34 edges. According to prior works, there are 7 singleton attractors among $2^{11} = 2048$ states in the state space of the cell-cycle network, as shown in Fig 1(b). Next, we will show how to find these singleton attractors with our algorithm.

A flow chart for detecting all the singleton attractors of the network is shown in Fig 2. We obtain $S_1 = 0$ according to the known network structure and the rule 5. Next, we begin to detect the states of the variable nodes. We firstly rank all nodes from large to small according to the number of their putative inhibitory connections, the order is node 10, 9, 8, 5, 7, 4, 6, 3, 2, 11, 1.



	Cln3	MBF	SBF	Cln1,2	Cdh1	Swi5	Cdc20/14	Clb5,6	Sic1	Clb1,2	Mcm1/SFF
Basin size	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}	S_{11}
1875	0	0	0	0	1	0	0	0	1	0	0
62	0	1	0	0	1	0	0	0	1	0	0
2	0	1	0	0	0	0	0	0	1	0	0
11	0	0	0	0	1	0	0	0	1	0	0
1	0	0	0	0	1	0	0	0	0	0	0
88	0	0	1	1	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0	0

Fig 1. (a) The cell-cycle network of budding yeast. The green solid and pink dashed arrows represent positive and negative interactions, respectively. (b) Seven singleton attractors are found under the strong-inhibition Boolean model for the network in (a). The basin size for each singleton attractor is also given in the figure.

doi:10.1371/journal.pone.0166906.g001

Firstly, the network state $(S_1, S_2, \ldots, S_N)$ is set to $(0, S_2, \ldots, S_N)$ with $l_0 = N - 1$. Then, we get $S_7 = 1$ according to the rule 7 by setting $S_{10} = 1$. Afterwards, $S_{10} = 0$ is obtained according to the rule 1. Now the contradiction appears, so the state $S = (0, S_2, \ldots, S_9, 1, S_{11})$ should be removed. On the other hand, assign $S_{10} = 0$. It turns out that no more nodes' state can be determined, so we remember the state $S = (0, S_2, \ldots, S_9, 0, S_{11})$ with $l_0 = N - 2$ and begin the next turn.

Secondly, we have network state $S = (0, S_2, \ldots, S_9, 0, S_{11})$ with $l_0 = N - 2$. Then, we get $S_8 = 0$ and $S_4 = 0$ according to the rules 1 and 2 by assigning $S_9 = 1$. We obtain $S_3 = 0$ and $S_7 = S_{11} = 0$ based on these known states and the rules 4, 5. $S_6 = 0$ according to the rule 5. Now, no more nodes' states can be determined, so we remember the state $S = (0, S_2, 0, 0, S_5, 0, 0, 0, 1, 0, 0)$ with $l_0 = 2$ and wait the next turn. Assign $S_9 = 0$, we can not determine the states of any nodes, so we remember the state $S = (0, S_2, \ldots, S_8, 0, 0, S_{11})$ with $l_0 = 8$ and begin the next turn.

Thirdly, there are two network states $S = (0, S_2, 0, 0, S_5, 0, 0, 0, 1, 0, 0)$ with $l_0 = 2$ and $S = (0, S_2, \ldots, S_8, 0, 0, S_{11})$ with $l_0 = 8$. For the first network state, we find that all cases are ok through

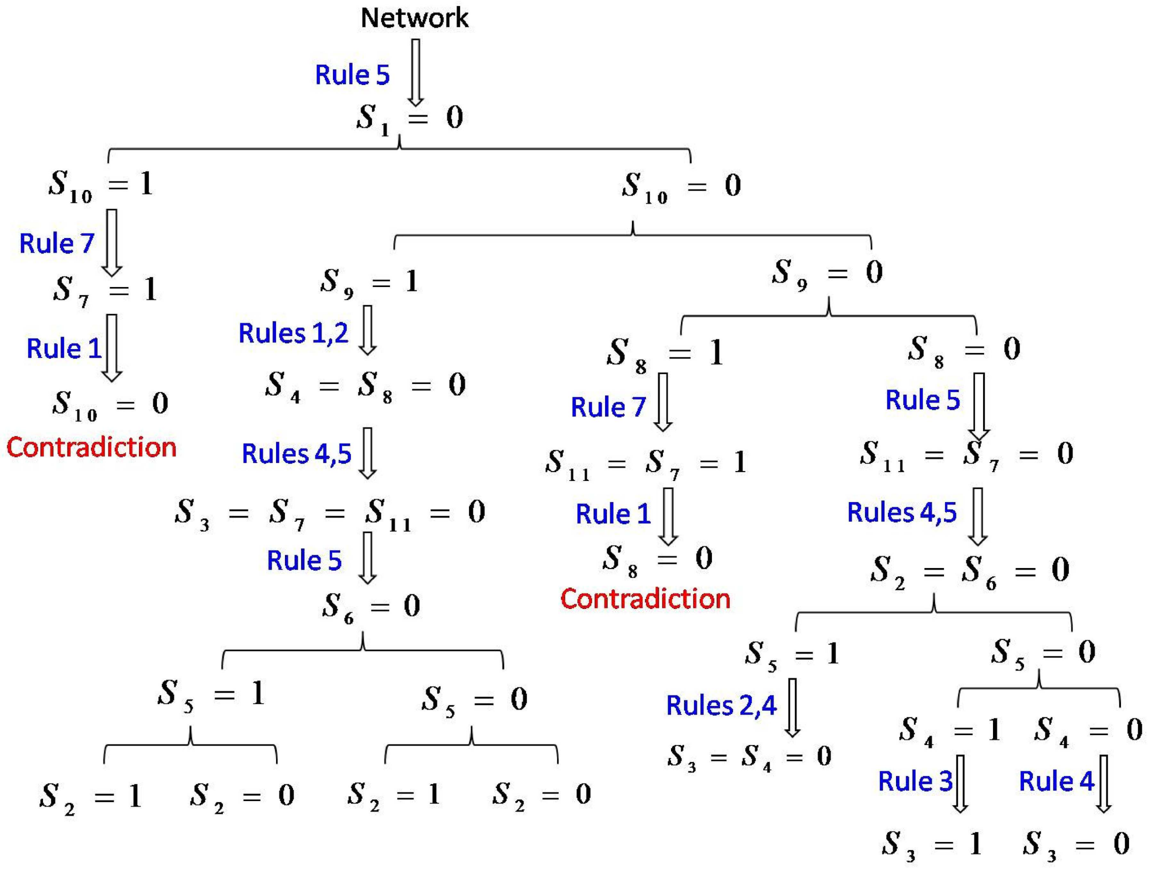


Fig 2. The flow chart for detecting all singleton attractors of the cell-cycle network of budding yeast.

doi:10.1371/journal.pone.0166906.g002

the assignment of $S_2 = 0$ or 1 and $S_5 = 0$ or 1. Therefore, we can obtain 4 singleton attractors: $S = (0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0)$, $(0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0)$, $(0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0)$ and $(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0)$. For the second network state, assigning $S_8 = 1$, we obtain $S_{11} = 1$, and then we obtain $S_7 = 1$ according to the rule 7. We can get $S_8 = 0$ according to $S_7 = 1$ and the rule 1, when contradiction appears. So this state should be removed. On the other hand, we assign $S_8 = 0$, according to the rule 5, we have $S_{11} = 0$, and further $S_7 = 0$. Then according to the rules 4 and 5, we obtain $S_2 = 0$ and $S_6 = 0$, respectively. Now, we can not determine the states of any nodes, so we remember the state $S = (0, 0, S_3, S_4, S_5, 0, 0, 0, 0, 0, 0)$ with $l_0 = 3$ and begin the next turn.

Fourthly, the network state is $S = (0, 0, S_3, S_4, S_5, 0, 0, 0, 0, 0, 0)$ with $l_0 = 3$. We assign $S_5 = 1$, $S_4 = 0$ is determined according to the rule 2. Further, we have $S_3 = 0$ according to the rule 4. Therefore, we can return the singleton attractor $S = (0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$. On the other hand, we assign $S_5 = 0$, and find that the states of nodes 3 and 4 can not be determined. So we remember the state $S = (0, 0, S_3, S_4, 0, 0, 0, 0, 0, 0, 0)$ with $l_0 = 2$ and begin the next turn.

Fifthly, the network state $S = (0, 0, S_3, S_4, 0, 0, 0, 0, 0, 0, 0)$ with $l_0 = 2$ is gotten. Assigning $S_4 = 1$, we have $S_3 = 1$ according to the rule 3. So we return the singleton attractor $S = (0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0)$. Next, we assign $S_4 = 0$, we obtain $S_3 = 0$ according to the rule 4. So we return the singleton attractor $S = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)$.

Finally, we get all the singleton attractors: $S = (0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0)$, $(0, 1, 0, 0, 1, 0, 0, 0, 1, 0, 0)$, $(0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0)$, $(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0)$, $(0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0)$,

Table 1. Experiment results of five classical networks.

	Network size		Our algorithm		Enum. algorithm	
Network name	**Nodes**	**Edges**	***NS***	**time(*s*)**	***NS***	**time(*s*)**
Cancer cell [50]	8	21	2	$<10^{-8}$	2	$<10^{-8}$
Budding yeast [16]	11	34	7	$<10^{-8}$	34	$<10^{-8}$
Arabidopsis thaliana [51]	15	43	188	$<10^{-8}$	43	0.0156
T-helper cell [52]	23	35	4453	0.1404	4453	2.5896
T-cell receptor [53]	40	596	1364	0.0468	1364	147080.09

doi:10.1371/journal.pone.0166906.t001

(0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0) and (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0). These seven states are exactly all the singleton attractors of the cell-cycle network of budding yeast as shown in Fig 1(b).

Furthermore, the singleton attractors of five classical networks are detected. As shown in Table 1, the number of singleton attractors (*NS*) and running time of our algorithm for each network are given in columns 4 and 5. We also give the results of the enumeration-based algorithm in columns 6 and 7. Comparing our algorithm with the enumeration-based algorithm, our algorithm can not only find the same number of the singleton attractors with the enumeration-based algorithm (columns 4 and 6), but also decrease extremely the running time (columns 5 and 7). Especially for the largest network (T-cell receptor), it took round 0.0468 seconds for our algorithm, whereas the enumeration-based algorithm spent more than 40 hours.

Next, we will show the computational time of our algorithm by numerical simulations on random gene regulatory networks. Similar to the ER random network [54], there exits interaction ($a_{ij} \neq 0$) from node j to i ($i = j$ is allowed) with the probability p. Usually, the parameter p is determined by $p = \langle k \rangle / N$, where $\langle k \rangle$ is the average degree of the network. Furthermore, $a_{ij} \neq 0$ has been assumed to take an independent value (±1) distributed according to a two-point probability distribution function. More specifically, we adopt the following values at random:

$$a_{ij} = \begin{cases} -1, & \text{the inhibiting interaction with probability } r, \\ 1, & \text{the activating interaction with probability } 1 - r. \end{cases} \tag{17}$$

The average degree is about $2 \sim 4$ for many biological networks, and the amount of inhibiting interactions is less than that of activating interactions [15, 55–57], so $\langle k \rangle = 3$ and $r = 0.4$ are fixed. We set $N = 50$ and generate 500 sample networks. The CPU computational time for finding all the singleton attractors of each network with above algorithm is calculated and shown in Fig 3(a). As we can see, the singleton attractors of every network can be detected, and the time is no more than 1 minute. This indicates that the singleton attractors of a network can be found efficiently with our algorithm.

The average CPU computational time *avT* as a function of the network size *N* is also computed. The result is shown in a semi-logarithmic plot in Fig 3(b), with a straight fit (dashed line) to guide an eye. The value of *avT* for each *N* is averaged over 500 samples. And the error bars denote the range of CPU computational time, where the upper and lower ends of bars represent the maximum and minimum values, respectively. In the figure, the average CPU computational time increases as the growth of network size *N*. This behavior is well characterized by the exponential relationship, $avT \propto 1.34^N$. This exponential increase also exists between the maximum CPU computational time and the size of network.

Although the average and maximum CPU computational time increase exponentially with the increase of *N*, we do not know why the minimum CPU computational time is very short for

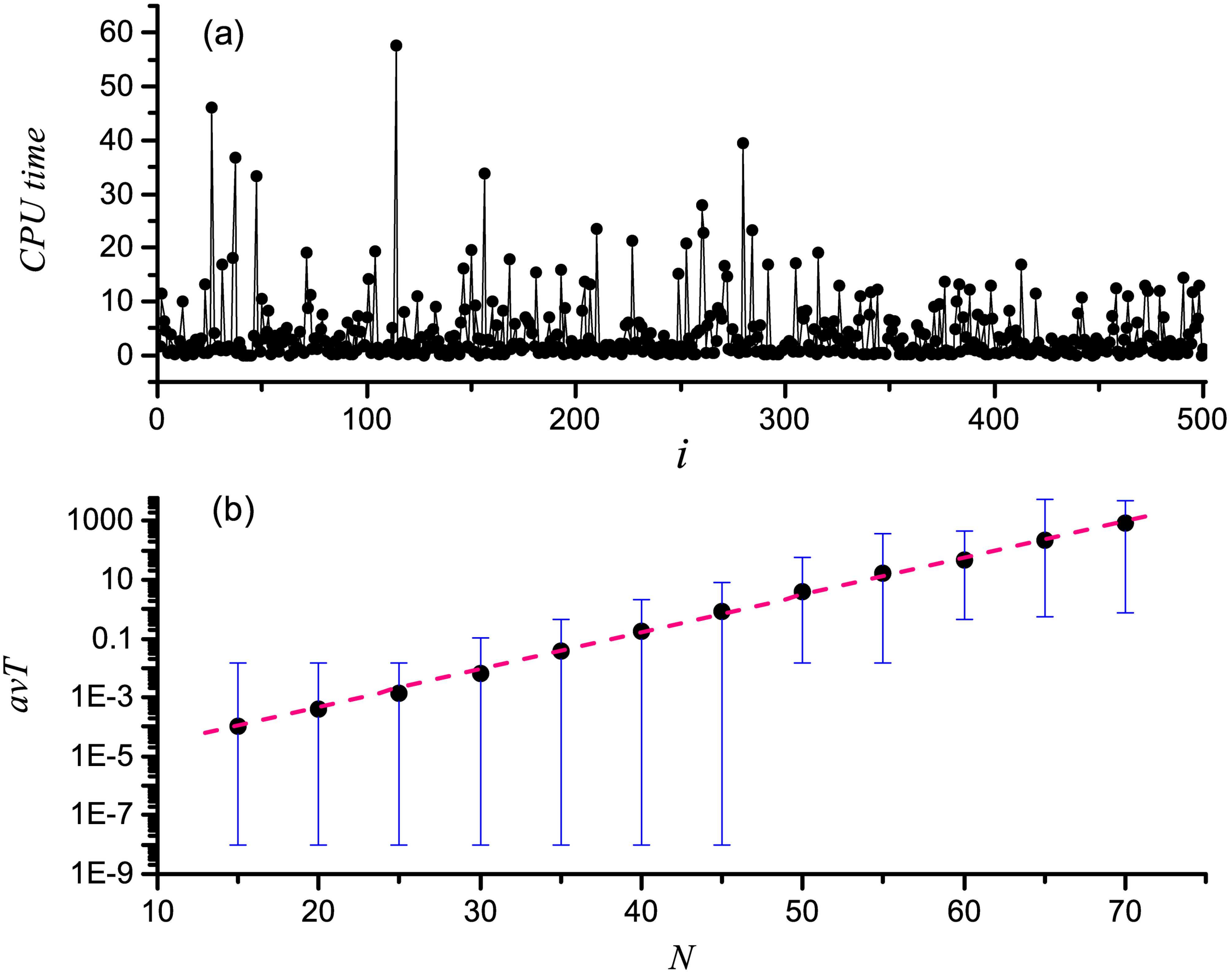


Fig 3. (a) The CPU computational time assumed of the algorithm for finding the singleton attractors of M ($M = 500$) random genetic regulatory network with $N = 50$, $\langle k \rangle = 3$, $r = 0.4$. (b) Semi-logarithmic plot of the averaged CPU computational time (*avT*) as a function of the network size *N*. For each *N*, the value of *avT* is averaged over M ($M \geq 500$) samples. And the error bars denote the range of CPU computational time, while the upper and lower ends of bars represent the maximum and minimum values, respectively. The straight dashed line is linearly fit of the data, indicative of the correlation $avT \propto 1.34^N$.

doi:10.1371/journal.pone.0166906.g003

each *N*, as shown in Fig 3(b). To understand this question, the maximum, average, and minimum number of the singleton attractors as the function of the size of system *N* are shown in Fig 4(a), respectively. From this figure, we find that the average and the maximum number of the singleton attractors increase exponentially with *N* increasing, and there is not exponential relation between the minimum number of the singleton attractors and the size of system. Furthermore, the sample networks with size of $N = 50$ are generated. We plot the CPU computational time against the number of singleton attractors (*NS*) of the sample networks, and show the result in Fig 4(b). We find that the computational time increases linearly with the number of the singleton attractors increasing. Based on these observations, the minimum CPU computational time nonexponentially increases with the size of system, as the exponential relation between the minimum number of the singleton attractors and the size of system does not exist.

Following the above observation, we may want to know what characteristics of networks could have small number of singleton attractors. So we calculate the average number of

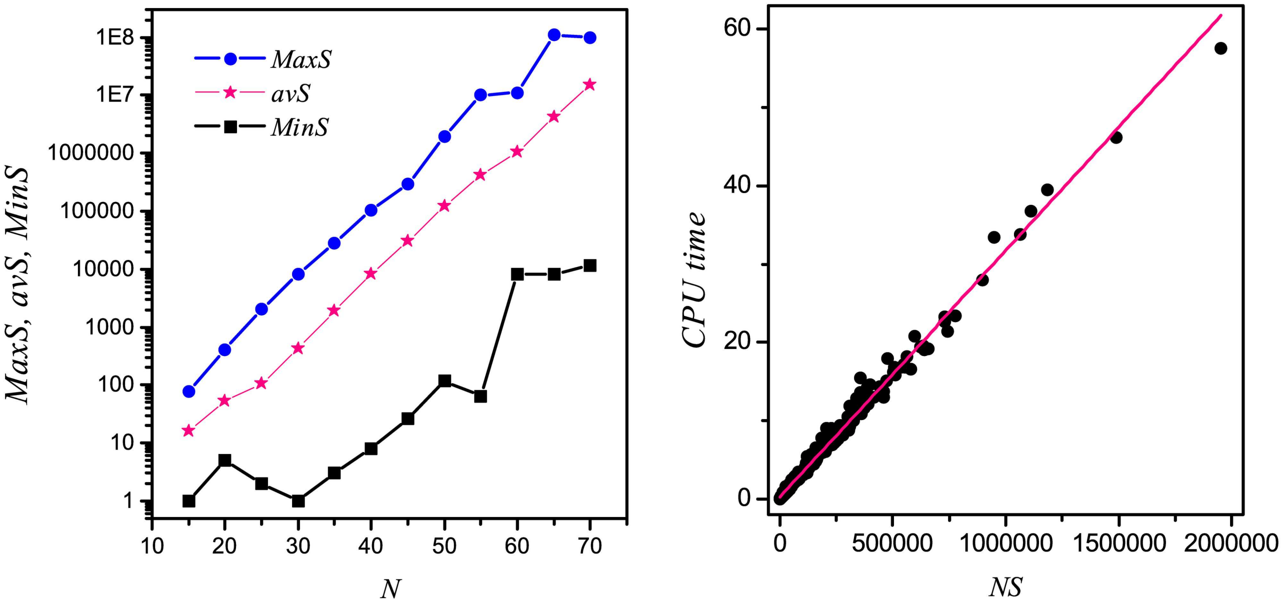


Fig 4. (a) Semi-logarithmic plot of the maximum (*MaxS*), average (*avS*), and minimum (*MinS*) number of the singleton attractors of networks as the function of the size of system *N*. (b) Semi-logarithmic plot of the CPU computational time vs the number of the singleton attractors (*NS*). The straight line is linearly fit of the data to guide an eye.

doi:10.1371/journal.pone.0166906.g004

singleton attractors of networks for different average degree $\langle k \rangle' s$ and inhibiting interactions probability $r' s$. In Fig 5(a) and 5(b), we show the semi-logarithmic plots of *avS* (the average number of singleton attractors) as a function of $\langle k \rangle$ and *r*, respectively. We can find that *avS* decreases extremely fast with the increase of $\langle k \rangle$ and increases exponentially with the increase of *r*. According to the above observation, these results indicate that less computational time is assumed to find the singleton attractors of the networks with high average degree and less inhibitory interactions.

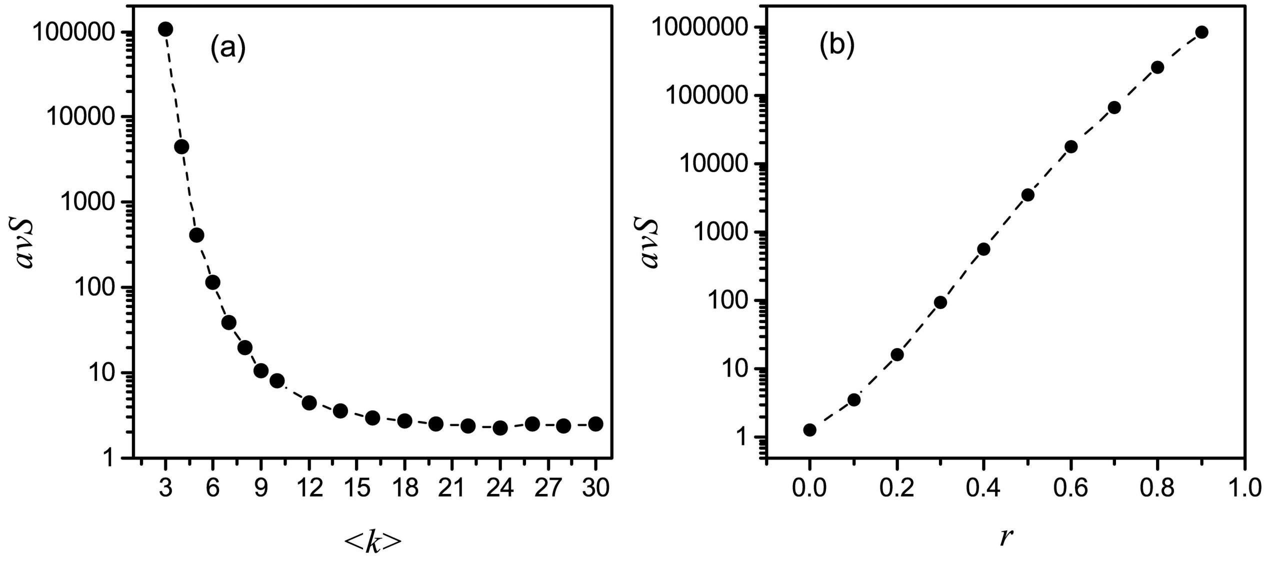


Fig 5. (a) Semi-logarithmic plot of the *avS* as a function of the average degree $\langle k \rangle$ for networks with $N = 50$, $r = 0.4$. (b) Semi-logarithmic plot of the *avS* against the inhibiting interactions probability *r* for networks with $N = 50$, $\langle k \rangle = 5$. Every data in the plots is averaged over M ($M \geq 500$) samples.

doi:10.1371/journal.pone.0166906.g005

Finding the pre-images of any target network state

The pre-images are the set of all possible inputs which evolve into a given output. For the strong-inhibition Boolean model, the pre-images of a given state $S^0(t+1)$ (target state) are the solutions of

$$S_i^0(t+1) = (\sum_{j\neq i}(S_j(t)g_{ij}) + S_i(t)\overline{r}_{ii} + \overline{S_i(t)}g_{ii})\prod_{j\neq i}(\overline{S_j(t)r_{ij}}), \tag{18}$$

For simplity, we will use S_j to represent $S_j(t)$. Through analysing this equation, we can still obtain the following rules if $S_i^0(t+1) = 1$,

rule 1.1. if $r_{ij} = 1$, then $S_j = 0$;

rule 1.2. if $r_{ii} = 1$, $g_{ij_0} = 1$, $\Sigma_{j\neq i}(S_jr_{ij}) = 0$ and $\Sigma_{j\neq j_0}(S_jg_{ij}) = 0$, then $S_{j_0} = 1$;

rule 1.3. if $g_{ii} = 0$, $r_{ii} = 0$, $\Sigma_{j\neq i}(S_jr_{ij}) = 0$ and $\Sigma_{j\neq i}(S_jg_{ij}) = 0$, then $S_i = 1$;

rule 1.4. if $g_{ii} = 0$, $g_{ij_0} = 1$, $\Sigma_{j\neq i}(S_jr_{ij}) = 0$, $\Sigma_{j\neq j_0}(S_jg_{ij}) = 0$ and $S_i = 0$, then $S_{j_0} = 1$;

rule 1.5. if $r_{ii} = 1$ and $\Sigma_{j\neq i}(S_jg_{ij}) = 0$, then contradiction appears.

Some rules are given if $S_i^0(t+1) = 0$,

rule 2.1. if $\Sigma_{j\neq i}(S_jr_{ij}) = 0$ and there is a node j_0 such that $g_{ij_0} = 1$, then $S_{j_0} = 0$;

rule 2.2. if $g_{ii} = 0$, $r_{ii} = 0$ and $\Sigma_{j\neq i}(S_jr_{ij}) = 0$, then $S_i = 0$;

rule 2.3. if $g_{ii} = 1$, $r_{ij_0} = 1$, and $\Sigma_{j\neq j_0}(S_jr_{ij}) = 0$, then $S_{j_0} = 1$;

rule 2.4. if $r_{ij_0} = 1$, $\Sigma_{j\neq i,\, j\neq j_0}(S_jr_{ij}) = 0$ and $g_{ii} = 1$ or there is a node j_0' such that $S_{j_0'}g_{ij_0'} = 1$, then $S_{j_0} = 1$;

rule 2.5. if $g_{ii} = 1$ and $\Sigma_{j\neq i}(S_jr_{ij}) = 0$, then contradiction appears.

If one observes carefully those rules, we find that the previous algorithm for detecting the singleton attractors can be extended to find the pre-images for any target network state. Here, we just need to input the target state in **Step 1** and replace the rules 1 − 7 with rules 1.1 − 1.5 and 2.1 − 2.5 in **Steps 3** and 4.

Next, we will prove that the states found with this algorithm are precisely all the pre-images of the target network state. Here, we still use U and V to stand for the set of the states found with this algorithm and the set of all the pre-images of the target network state, respectively. Obviously, any pre-image of $S^0(t+1)$ must satisfy Eq (18), and it follows certainly all the rules. So V is a subset of U, namely $V \subset U$. Therefore, to prove $U = V$, we just need to prove $U \subset V$.

Suppose that U is not a subset of V, then there must be a state $S^0(t)$ such that $S^0(t) \in U$ and $S^0(t) \notin V$. The state $S^0(t)$ obeys all the rules but does not satisfy Eq (18), namely, there is a node's state which dissatisfies the equation. We assume that this state is $S_i^0(t+1)$, and we have

$$S_i^0(t+1) \neq (\sum_{j\neq i}(S_j^0(t)g_{ij}) + S_i^0(t)\overline{r}_{ii} + \overline{S_i^0(t)}g_{ii})\prod_{j\neq i}(\overline{S_j^0(t)r_{ij}}), \tag{19}$$

where $S_i^0(t+1) = 0$ or 1.

Firstly, assuming $S_i^0(t+1) = 1$ and inserting it into Eq (19), we obtain

$$(\sum_{j\neq i}(S_j^0(t)g_{ij}) + S_i^0(t)\overline{r}_{ii} + \overline{S_i^0(t)}g_{ii})\prod_{j\neq i}(\overline{S_j^0(t)r_{ij}}) = 0. \tag{20}$$

This equation indicates

$$\prod_{j\neq i}(\overline{S_j^0(t)r_{ij}}) = 0 \tag{21}$$

or

$$\sum_{j\neq i}(S_j^0(t)g_{ij}) + S_i^0(t)\overline{r}_{ii} + \overline{S_i^0(t)}g_{ii} = 0. \tag{22}$$

Through Eq (21), there must be a node j_0 such that $S_{j_0}^0(t) = 1$ and $r_{ij_0} = 1$, we can find that it contradicts with the rules 1.1. So it is necessary that $\prod_{j\neq i}(\overline{S_j^0(t)r_{ij}}) = 1$, hence we have $\Sigma_{j\neq i}(S_j r_{ij}) = 0$. If Eq (22) holds, one case is $\sum_{j\neq i}(S_j^0(t)g_{ij}) = 0$, $g_{ii} = r_{ii} = 0$ and $S_i^0(t) = 0$, it contradicts with the rule 1.3 through $\Sigma_{j\neq i}(S_j r_{ij}) = 0$; the other case is $\sum_{j\neq i}(S_j^0(t)g_{ij}) = 0$ and $r_{ii} = 1$, we can observe that the contradiction appears according to the rule 1.5. Therefore, it is impossible that $S^0(t)$ which obeys all the rules does not satisfy Eq (18), if $S_i^0(t+1) = 1$.

Afterwards, we assume $S_i^0(t+1) = 0$. Inserting it into Eq (19), we obtain

$$(\sum_{j\neq i}(S_j^0(t)g_{ij}) + S_i^0(t)\overline{r}_{ii} + \overline{S_i^0(t)}g_{ii})\prod_{j\neq i}(\overline{S_j^0(t)r_{ij}}) = 1. \tag{23}$$

From this equation, we can get

$$\prod_{j\neq i}(\overline{S_j^0(t)r_{ij}}) = 1, \tag{24}$$

$$(\sum_{j\neq i}(S_j^0(t)g_{ij}) + S_i^0(t)\overline{r}_{ii} + \overline{S_i^0(t)}g_{ii}) = 1. \tag{25}$$

The Eq (24) indicates

$$\sum_{j\neq i}(S_j^0(t)r_{ij}) = 0. \tag{26}$$

And from Eq (25), we obtain

$$\sum_{j\neq i}(S_j^0(t)g_{ij}) = 1 \tag{27}$$

or

$$S_i^0(t)\overline{r}_{ii} + \overline{S_i^0(t)}g_{ii} = 1. \tag{28}$$

If Eq (27) works, then there must be a node j_0 such that

$$S_{j_0}^0(t) = 1 \quad and \quad g_{ij_0}(t) = 1. \tag{29}$$

Through Eqs (26) and (29), we can find that it contradicts with the rule 2.1. On the other hand, if Eq (28) is valid, two cases should be discussed: one case is $g_{ii} = r_{ii} = 0$ and $S_i^0(t) = 1$, it contradicts with the rule 2.2 since Eq (26) holds; the other case is $g_{ii} = 1$ ($r_{ii} = 0$ at this time), combining with Eq (26), one can see that the contradiction appears according to the rule 1.5. In summary, if $S_i^0(t+1) = 0$, it is also impossible that $S^0(t)$ obeys all the rules and dissatisfies Eq (18) at the same time.

These results demonstrate that the state $S^0(t)$, which obeys all the rules but dissatisfies the Eq (18), does not exist. Therefore, U is a subset of V ($U \subset V$), and $U = V$ is proved. Now, we

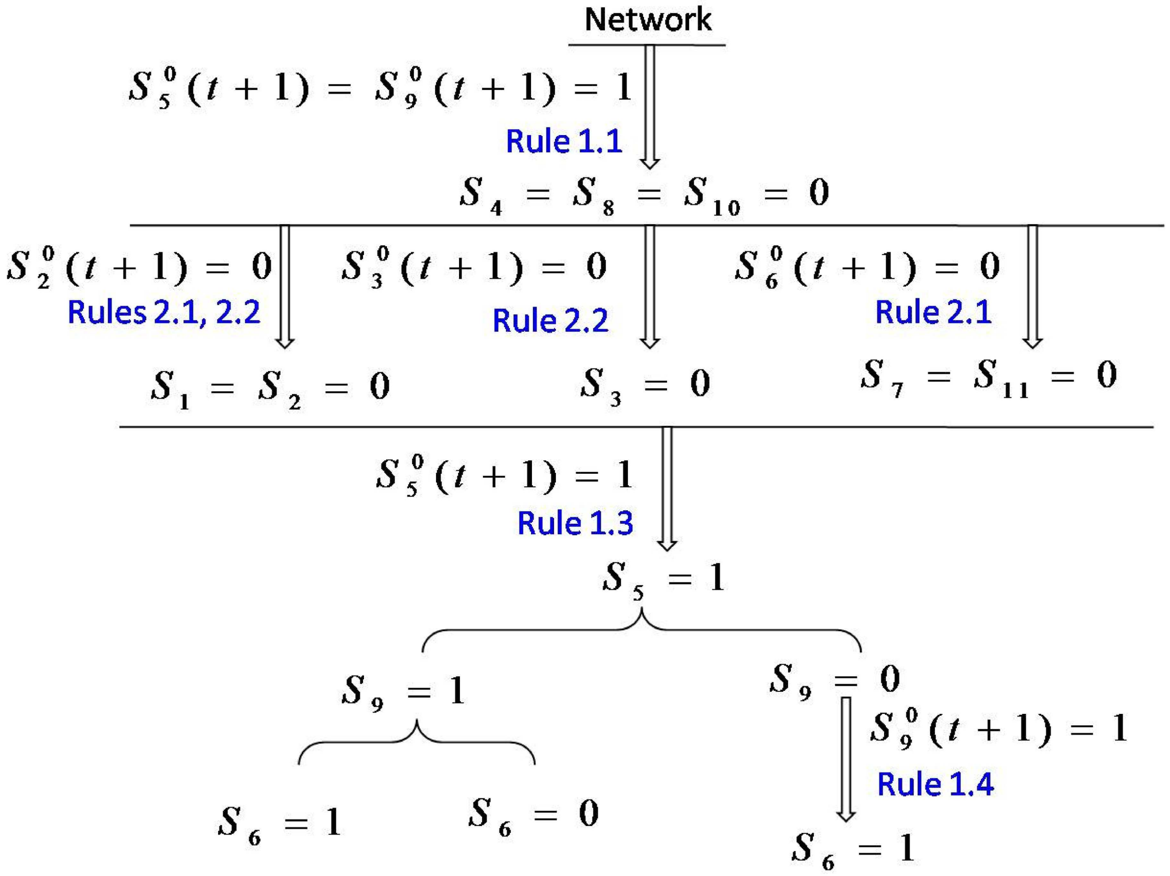


Fig 6. The flow chart for determing the pre-images of target state *S* = (0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0).

doi:10.1371/journal.pone.0166906.g006

can conclude that the states found with our algorithm are precisely all the pre-images of the target network state.

As an example, we will show that how we find the pre-images of one singleton attractor (0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0) of the cell-cycle network of budding yeast. For this case, $S^0(t + 1) = (0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0)$ in Eq (18). As shown in Fig 6, according to the known network structure and the rules 1.1 – 1.5 and 2.1 – 2.5, the states of many nodes are determined: we obtain $S_4 = S_8 = S_{10} = 0$ based on $S_5^0(t+1) = S_9^0(t+1) = 1$ and the rule 1.1; then, $S_1 = 0$ and $S_2 = 0$ are gotten according to $S_2^0(t+1) = 0$ and the rules 2.1 and 2.2, respectively; according to $S_3^0(t+1) = 0$ and the observation 2.2, we obtain $S_3 = 0$; according to $S_6^0(t+1) = 0$ and the observation 2.1, we obtain $S_7 = S_{11} = 0$; then again according to $S_5^0(t+1) = 1$ and the rule 1.3, we obtain $S_5 = 1$. Now, the remain nodes whose states can not be determined are nodes 6 and 9, so we remember the state $S = (0, 0, 0, 0, 1, S_6, 0, 0, S_9, 0, 0)$ with $l_0 = 2$.

For the network state $S = (0, 0, 0, 0, 1, S_6, 0, 0, S_9, 0, 0)$ with $l_0 = 2$. We assign $S_9 = 1$ firstly. However, the state of node 6 can not be determined, and we find that $S_6 = 0$ and $S_6 = 1$ both are the solutions of the equations. So we return the pre-images $S = (0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0)$ and (0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0). Then, we assign $S_9 = 0$. According to the rule 1.4, we obtain $S_6 = 1$ and the pre-image is $S = (0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0)$. Finally, we successfully find three pre-images of the target network state $S^0(t + 1)$: $S = (0, 0, 0, 0, 1, 1, 0, 0, 1, 0, 0)$, (0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0) and (0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0).

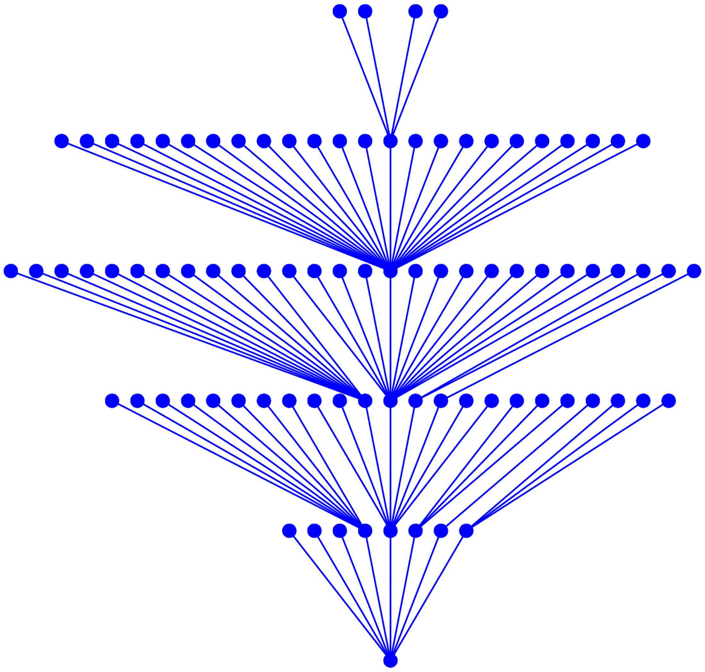


Fig 7. The basin of the singleton attractor (0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0) is displayed hierarchically.

doi:10.1371/journal.pone.0166906.g007

We can also find the basin of any singleton attractor with this algorithm by reversely inferring the pre-images step by step: we can obtain the pre-images of the singleton attractor first, then find the pre-images of all these known pre-images, go on until all states of the basin are found. The basin of the singleton attractor (0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0) is displayed hierarchically in Fig 7, where we can clearly see how far a state is from the singleton attractor.

Conclusion

In summary, we have presented a novel algorithm for finding the singleton attractors in strong-inhibition Boolean networks. The average case time complexity of the proposed algorithm is approximately $O(1.34^N)$ which is much faster than the enumeration-based algorithm. It may not be faster than the out-degree based ordering algorithm in [30] for networks with very low maximum indegree. However, we find that the computational time assumed of the algorithm is proportional to the number of the singleton attractors, it shows that our algorithm will work much better for networks with less singleton attractors, especially for the networks with high average degree and less inhibitory interactions. What's more, the algorithm can be extended to identify the pre-images of any network state and the basin of any singleton attractor. Therefore, the proposed algorithm could be effective and practical. We hope it has good applications in the study of biological networks. On the other hand, we can also know that the algorithm has its own limitations. For example, our algorithm completely relies on the strong-inhibition Boolean model, it may not be workable for other kinds of model, such as the general Boolean model. The computational time of the algorithm increases exponentially with the size of network, so it may not be applied to networks with several hundreds or more nodes. Moreover, we didn't get the theoretical results of the time complexity of the algorithm though we have made great efforts. We hope some of these limitations can be overcome and this work can be further improved in the future.

Supporting Information

S1 File. Data and code of the algorithm. This data set contains the code of the algorithm for finding the singleton attractors and pre-images in strong-inhibition Boolean networks. The adjacency matrices of five classical networks are also given.
(ZIP)

Acknowledgments

We thank members in our laboratory for inspiring discussions and helpful comments on earlier versions of the manuscript.

Author Contributions

Conceptualization: MZ ZH.

Methodology: MZ ZH.

Software: ZH MZ SL CY.

Writing – original draft: ZH CY ZF.

Writing – review & editing: ZH CY ZF.

References

1. Kauffman SA. The origins of order: self-organization and selection in ecolution. Oxford: Oxford University Press; 1993. doi: 10.1007/978-94-015-8054-0_8
2. Rigoutsos I, Stephanopoulos G. Systems biology: Volume II: networks, models, and applications. Oxford: Oxford University Press; 2007.
3. Alon U. An introduction to system biology: design principles of biological circuits. Chapman & Hall, Boca Raton, FL; 2006.
4. Nicolis G, Prigogine I. Self-organization in nonequilibrium systems, from dissipative structures to order through fluctuations. Wiley, New York; 1977.
5. Gillespie DT. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. Journal of Computational Physics. 1976; 22(4): 403–434. doi: 10.1016/0021-9991(76)90041-3
6. Gillespie DT. Exact stochastic simulation of coupled chemical reactions. The Journal of Physical Chemistry. 1977; 81(25): 2340–2361. doi: 10.1021/j100540a008
7. Cai XD and Yuan ZM. Stochastic modeling and simulation of the p53-MDM2/MDMX Loop. Journal of Computational Biology. 2009; 16(7): 917–933. doi: 10.1089/cmb.2008.0231 PMID: 19580521
8. Zhang YP, Qian M, Ouyang Q, Deng M, Li F, Tang C. Stochastic model of yeast cell-cycle network. Physica D. 2006; 219(1): 35–39. doi: 10.1016/j.physd.2006.05.009
9. Goodwin BC. Temporal organization in cells: a dynamic theory of cellular control. Academic Press, London; 1963. doi: 10.5962/bhl.title.6268
10. Chen KC, Csikász-Nagy A, Gyorffy B, Val J, Novak B, Tyson JJ. Kinetic analysis of a molecular model of the budding yeast cell cycle. Mol. Biol. Cell. 2000; 11(1): 369–391. doi: 10.1091/mbc.11.1.369 PMID: 10637314
11. Novak B, Pataki Z, Ciliberto A, Tyson JJ. Mathematical model of the cell division cycle of fission yeast. Chaos. 2001; 11(1): 277–286. doi: 10.1063/1.1345725 PMID: 12779461
12. Tyson JJ, Chen KC, Novak B. Network dynamics and cell physiology. Nat. Rev. Mol. Cell Biol. 2001; 2(12): 908–916. doi: 10.1038/35103078 PMID: 11733770
13. Csikász-Nagy A. Computational systems biology of the cell cycle. Brief Bioinform. 2009; 10(4): 424–434. doi: 10.1093/bib/bbp005 PMID: 19270018
14. Kauffman SA. Metabolic stability and epigensis in randomly constructed genetic nets. Journal of Theoretical Biology. 1969; 22(3): 437–467. PMID: 5803332

15. Li FT, Long T, Lu Y, Ouyang Q, Tang C. The yeast cell-cycle network is robustly designed. PNAS. 2004; 101(14): 4781–4786. doi: 10.1073/pnas.0305937101 PMID: 15037758

16. Bornholdt S. Systems biology: less is more in model large genetic networks. Science. 2005; 310(5747): 449–451. doi: 10.1126/science.1119959 PMID: 16239464

17. Jong d.H. Modeling and simulation of genetic regulatory systems: a literature review. Journal of Computational Biology. 2002; 9(1): 67–103. doi: 10.1089/10665270252833208 PMID: 11911796

18. Wang GY, Du C, Chen H, Simha R, Rong Y, Xiao Y, Zeng C. Process-based network decomposition reveals backbone motif structure. PNAS. 2010; 107(23): 10478–10483. doi: 10.1073/pnas.0914180107 PMID: 20498084

19. Xia QZ, Liu LL, Ye WM, Hu G. Inference of gene regulatory networks with the strong-inhibition Boolean model. New Journal of Physics. 2011; 13(8): 083002. doi: 10.1088/1367-2630/13/8/083002

20. Wang GY, Rong YW, Chen H, Pearson C, Du CH, Simha R, Zeng C. Process-driven inference of biological network structure: feasibility, minimality, and multiplicity. PLoS ONE. 2012; 7(7): e40330. doi: 10.1371/journal.pone.0040330 PMID: 22815739

21. Huang S. Gene expression profiling, genetic networks and cellular states: an integrating concept for tumorigenesis and drug discovery. Journal of Molecular Medicine. 1999; 77(6): 469–480. doi: 10.1007/s001099900023 PMID: 10475062

22. Hupp T, Lane D, Ball K. Strategies for manipulating the p53 pathway in the treatment of human cancer. Biochemical Journal. 2000; 352: 1–17. doi: 10.1042/0264-6021:3520001 PMID: 11062053

23. Lane D. Exploiting the p53 pathway for cancer diagnosis and therapy. British Journal of Cancer. 1999; 80: 1–5. PMID: 10466753

24. Ostergaard S, Olsson L, Johnston M, Nielsen J. Increasing galactose consumption by Saccharomyces cerevisiae through metabolic engineering of the GAL gene regulatory network. Nature biotechnology. 2000; 18(12): 1283–1286. doi: 10.1038/82400 PMID: 11101808

25. Orlando DA, Lin CY, Bernard A, Wang JY, Socolar JES, Iversen ES, et al. Global control of cell-cycle transcription by coupled CDK and network oscillators. Nature. 2008; 453(7197): 944–947. doi: 10.1038/nature06955 PMID: 18463633

26. Garg A, Xenarios I, Mendoza L, De Micheli G. An efficient method for dynamic analysis of gene regulatory networks and in silico gene perturbation experiments. In Annual International Conference on Research in Computational Molecular Biology. 2007; pp. 62–76.

27. Garg A, Di Cara A, Xenarios I, Mendoza L, De Micheli G. Synchronous versus asynchronous modeling of gene regulatory networks. Bioinformatics. 2008; 24(17): 1917–1925. doi: 10.1093/bioinformatics/btn336 PMID: 18614585

28. Devloo V, Hansen P, Labbé M. Identification of all steady states in large networks by logical analysis. Bulletin of mathematical biology. 2003; 65(6): 1025–1051. doi: 10.1016/S0092-8240(03)00061-2 PMID: 14607287

29. Akutsu T, Kuhara S, Maruyama O, Miyano S. A system for identifying genetic networks from gene expression patterns produced by gene disruptions and overexpressions. Genome Informatics. 1998; 9: 151–160. PMID: 11072331

30. Zhang SQ, Hayashida M, Akutsu T, Ching WK, Ng MK. Algorithms for finding small attractors in Boolean networks. Eurasip J. Bioinf. Syst. Biol. 2007; 2007(1): 20180. doi: 10.1155/2007/20180 PMID: 18253467

31. Shlomi T, Berkman O, Ruppin E. Regulatory on/off minimization of metabolic flux changes after genetic perturbations. PNAS. 2005; 102(21): 7695–7700. doi: 10.1073/pnas.0406346102 PMID: 15897462

32. Ay F, Xu F, Kahveci T. Scalable steady state analysis of Boolean biological regulatory networks. PloS ONE. 2009; 4(12): e7992. doi: 10.1371/journal.pone.0007992 PMID: 19956604

33. Zheng D, Yang G, Li X, Wang Z, Liu F, He L. An efficient algorithm for computing attractors of synchronous and asynchronous Boolean networks. PloS ONE. 2013; 8(4): e60593. doi: 10.1371/journal.pone.0060593 PMID: 23585840

34. Zheng D, Yang G, Li X, Wang Z, Hung WNN. An efficient algorithm for finding attractors in synchronous boolean networks with biochemical applications. Genet. Mol. Res. GMR. 2013; 12(4): 4656–4666. doi: 10.4238/2013.October.18.4 PMID: 24222242

35. Michele L, Andrea P, Giorgio P, Osvaldo Z. Finite size corrections to random Boolean networks. Journal of Statistical Mechanics: Theory and Experiment. 2006; 2006(12): P10212.

36. Tamyura T, Akutsu T. An $O(1.787^n)$-time algorithm for detecting a singleton attractor in a Boolean network consisting of AND/OR nodes. International Symposium on Fundamentals of Computation Theory. 2007; pp. 494–505.

37. Melkman AA, Tamura T, Akutsu T. Determining a singleton attractor of an AND/OR Boolean network in *O*(1.587^{n}) time. Information Processing Letters. 2010; 110(14): 565–569.

38. Tamyura T, Akutsu T. Algorithms for singleton attractor detection in planar and nonplanar AND/OR Boolean networks. Mathematics in Computer Science. 2009; 2(3): 401–420. doi: 10.1007/s11786-008-0063-5

39. Akutsu T, Melkman AA, Tamyura T, Yamamoto M. Determining a singleton attractor of a Boolean network with nested canalyzing functions. Journal of Computational Biology. 2011; 18(10): 1275–1290. doi: 10.1089/cmb.2010.0281 PMID: 21554129

40. Melkman AA, Akutsu T. An improved satisfiability algorithm for nested canalyzing functions and its application to determining a singleton attractor of a Boolean network. Journal of Computational Biology. 2013; 20(12): 958–969. doi: 10.1089/cmb.2013.0060 PMID: 24073923

41. Zou Y. An algorithm for detecting fixed points of Boolean network. ICME International Conference On Complex Medical Engineering, IEEE. 2013; pp. 670–673.

42. Milano M, Roli A. Solving the satisfiability problem through Boolean networks. In Proceedings of the 6th Congress of the Italian Association for Artificial Intelligence on Advances in Artificial Intelligence, vol. 1792 of Lecture Notes in Artificial Intelligence. 1999; pp. 72–83. doi: 10.1007/3-540-46238-4_7

43. Wuensche A. The ghost in machine: Basins of attraction of random Boolean networks. Aritifical life III Proceeding, Santa Fe institude Studies in the Science of Complex Addison-Wesley; 1994.

44. Wuensche A, Lesser MJ. The global dynamics of cellular automata: An atlas of basin of attraction fields of one-dimensional cellular automata. Addison-Wesley, Reading, MA; 1992.

45. Klotz JG, Bossert M, Schober S. Computing preimages of Boolean networks. BMC Bioinformatics. 2013; 14 (Suppl. 10): S4. doi: 10.1186/1471-2105-14-S10-S4 PMID: 24267277

46. Akutsu T, Hayashina M, Zhang SQ, Ching WK, Ng MK. Analyses and algorithms for predecessor and control problems for Boolean networks of bounded indegree. Information and Media Technologies. 2009; 4(2): 338–349. doi: 10.2197/ipsjtbio.1.23

47. Albert R, Othmer HG. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster. Journal of Theoretical Biology. 2003; 223(1): 1–18. doi: 10.1016/S0022-5193(03)00035-3 PMID: 12782112

48. Tan N, Ouyang Q. Design of a network with state stability. Journal of Theoretical Biology. 2006; 240(4): 592. doi: 10.1016/j.jtbi.2005.10.019 PMID: 16343546

49. Fortuna MA, Meli CJ. Do scale-free regulatory networks allow more expression than random ones? Journal of Theoretical Biology. 2007; 247(2): 331–336. doi: 10.1016/j.jtbi.2007.03.017 PMID: 17452043

50. Yang LJ, Meng Y, Bao C, Liu W, Ma C, Li A, et al. Robustness and backbone motif of a cancer network regulated by miR-17-92 cluster during the G1/S transition. PloS ONE. 2013; 8(3): e57009. doi: 10.1371/journal.pone.0057009 PMID: 23469179

51. Chaos A, Aldana M, Espinosa-Soto C, de Leon BGP, Arroyo AG, Alvarez-Buylla ER. From genes to flower patterns and evolution: dynamic models of gene regulatory networks. J. Plant Growth Regul. 2006; 25(4): 278–289. doi: 10.1007/s00344-006-0068-8

52. Luis M, Ioannis X. A method for the generation of standardized qualitative dynamical systems of regulatory networks. Theor. Biol. Med. Model. 2006; 3(1): 13. doi: 10.1186/1742-4682-3-13 PMID: 16542429

53. Klamt S, Saez-Rodriguez J, Lindquist JA, Simeoni L, Gilles ED. A methodology for the structural and functional analysis of signaling and regulatory networks. BMC Bioinformatics. 2006; 7(1): 56. doi: 10.1186/1471-2105-7-56 PMID: 16464248

54. Erdös P, Rényi A. On the evolution of random graphs. Publ. Math. Inst. Hung. Acad. Sci. 1960; 5: 17–61.

55. Bornholdt S, Rohlf T. Topological evolution of dynamical networks: global criticality from local dynamics. Phys. Rev. L. 2000; 84(26): 6114–6117. doi: 10.1103/PhysRevLett.84.6114 PMID: 10991137

56. Costanzo MC, Crawford ME, Hirschman JE, Kranz JE, Olsen P, Robertson LS, et al. YPDTM, PombePDTM and WormPDTM: model organism volumes of the BioKnowledge? Library, an integrated resource for protein information. Nucleic. Acida. Research. 2001; 29(1): 75–79. doi: 10.1093/nar/29.1.75 PMID: 11125054

57. Lee TI, Rinaldi NJ, Robert F, Odom DT, Bar-Joseph Z, Gerber GK, et al. Transcriptional regulatory networks in Saccharomyces cerevisiae. Science. 2002; 298(5594): 799–804. doi: 10.1126/science.1075090 PMID: 12399584