Inference of Biological Pathway from Gene Expression Profiles by Time Delay Boolean Networks

Tung-Hung Chueh[1], **Henry Horng-Shing Lu**[2]*

[1] Green Energy and Environment Research Laboratories, Industrial Technology Research Institute, Chutung, Hsinchu, Taiwan, Republic of China, [2] Institute of Statistics, National Chiao Tung University, Hsinchu, Taiwan, Republic of China

Abstract

One great challenge of genomic research is to efficiently and accurately identify complex gene regulatory networks. The development of high-throughput technologies provides numerous experimental data such as DNA sequences, protein sequence, and RNA expression profiles makes it possible to study interactions and regulations among genes or other substance in an organism. However, it is crucial to make inference of genetic regulatory networks from gene expression profiles and protein interaction data for systems biology. This study will develop a new approach to reconstruct time delay Boolean networks as a tool for exploring biological pathways. In the inference strategy, we will compare all pairs of input genes in those basic relationships by their corresponding *p*-scores for every output gene. Then, we will combine those consistent relationships to reveal the most probable relationship and reconstruct the genetic network. Specifically, we will prove that $O(\log n)$ state transition pairs are sufficient and necessary to reconstruct the time delay Boolean network of n nodes with high accuracy if the number of input genes to each gene is bounded. We also have implemented this method on simulated and empirical yeast gene expression data sets. The test results show that this proposed method is extensible for realistic networks.

Citation: Chueh T-H, Lu HH-S (2012) Inference of Biological Pathway from Gene Expression Profiles by Time Delay Boolean Networks. PLoS ONE 7(8): e42095. doi:10.1371/journal.pone.0042095

Editor: Frank Emmert-Streib, Queen's University Belfast, United Kingdom

Received February 21, 2012; **Accepted** July 2, 2012; **Published** August 31, 2012

Funding: The authors acknowledge support from the National Science Council, National Center for Theoretical Sciences, Shing-Tung Yau Center, and Center of Mathematical Modeling and Scientic Computing at the National Chiao Tung University in Taiwan. 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: hslu@stat.nctu.edu.tw

Introduction

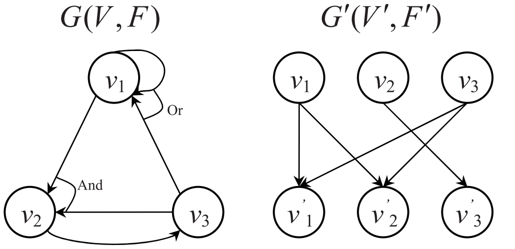
In order to understand complex biological networks and pathways, we need to investigate global structures instead of individual behaviors since there are interactions and associations between genes. Due to the invention of high throughput technology, genome-wide expression profiles can be measured simultaneously [1]. However, it is still a great challenge to identify complex biological networks from genome-wide data because the number of gene interactions is huge [2]. In recent years, there has been a significant progress in research concerning genetic network models and network reconstruction problems.

Clustering and dimension reduction are important methods for grouping genes that have similar expression profiles [3,4]. In the framework of clustering, it is important to define the degree of similarity between genes. By the method of clustering, we can group genes that have similar expressions. However, we still cannot find the causal relationship between genes. Hence, apart from the relationship of similarity, we will also have to consider another causal relationship between genes.

There have been many methods proposed in the literature to tackle the problem of genetic regulatory network reconstruction. For instance, the steady state approach have been used to model gene regulatory networks [5]. In addition, the Bayesian network model is an important technique that has been studied extensively in the past two decades [6–11]. A Bayesian network is a directed acyclic graph (DAG) comprised of two components. The first component is comprised of nodes that correspond to a set of variables and a set of directed edges between variables with Markov properties. The second component describes a conditional distribution for each variable given its parents in the graph. Recently, Bayesian network models have been applied to analyze microarray expression and biological data [12–15]. However, Bayesian network algorithms have limitations when dealing with large-scale gene regulatory networks because of their complex modeling structure [16]. Although algorithms for reconstructing Bayesian networks have already been developed [17,18], the algorithms' computational costs remain a concern for the searching of all potential network structures on the genome-wide expression data.

Therefore, we are considering a simpler model: Boolean networks, which have been studied extensively in a variety of contexts. Boolean networks [19,20] can effectively explain the dynamic behaviors of living systems. Moreover, for large-scale gene regulatory networks, Kim et al. [21] have used Boolean network with chi-square test on the yeast cell cycle microarray gene expression data sets. The chaos and attractors of Boolean network are also discussed widely from the aspect of power spectrum [22–24]. Recently, Boolean network also have been used as a discrete model for the *lac* operon [25].

Boolean networks were originally introduced by Kauffman, and received attention in the studies of gene regulatory networks because of their simple structures [26]. In a Boolean network



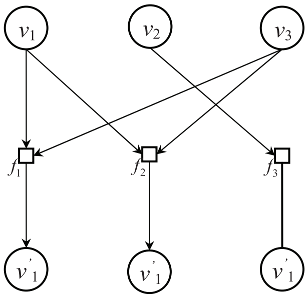
$v_1' = v_1 \text{ or } v_3 \quad v_2' = v_1 \text{ and } v_3 \quad v_3' = v_2$

Input			Output		
v_1	v_2	v_3	v_1'	v_2'	v_3'
0	0	0	0	0	0
0	0	1	1	0	0
0	1	0	0	0	1
0	1	1	1	0	1
1	0	0	1	0	0
1	0	1	1	1	0
1	1	0	1	0	1
1	1	1	1	1	1

Figure 1. Boolean network *G*(*V*,*F*), wiring diagram *G*′(*V*′,*F*′) and its input/output.
doi:10.1371/journal.pone.0042095.g001

model, nodes represent the gene expression states. The status of a gene is quantized to one of the two states: on or off, representing a gene as active or inactive respectively. The wiring of rules between nodes in the graph represents a functional link between genes and determines the expressions of target genes after giving a series of input genes. Under the structure of Boolean networks, the target gene is determined by a set of genes with specific rules. For each gene, if the indegree (i.e., the number of input genes to each gene) is bounded by a constant K, only $O(\log n)$ pairs of state transition are necessary and sufficient to reconstruct the original network with n nodes [27,28]. However, Boolean networks have been criticized for their deterministic nature. The assumption that every affected gene would be expressed immediately at the next time step may be unsound.

Another point of view of constructing genetic network is to focus on the indication the pairwise relationships between genes. Most of the works is to find the gene-pairs with similarity relationship [29–33]. The similarity of a gene-pair represents the two genes with the same expression or opposite expression. In 2005, Li and Lu proposed directed acyclic Boolean network and the statistical reconstruction method of SPAN to infer the pair wise relations of every element [34]. The proposed method can reconstruct Boolean networks from noisy array data by assigning an s-p-score for every pair of genes. In the study, they proposed another relationship between two genes: relationship of prerequisite under the Boolean network model. If gene A is a prerequisite for gene B, then the "on" status of gene A is necessary for the "on" status of gene B. Boolean implication network, with the similar aspect, investigated all Boolean implication between pairs of gene for large scale genome microarray datasets [35]. Following the model, Wang et al.[36] proposed a two step counting approach for constructing biological pathways with Boolean network. However, most of these methods only consider pair wise relationship in order to decrease the time complexity. Therefore, if the structure of network is a combination of a set of genes to affect another gene, the algorithms will lose some information and rules in the genetic network reconstruction.



$f_1 = (v_1 \text{ or } v_3) \prec v_1'$

$f_2 = (v_1 \text{ and } v_3) \prec v_2'$

$f_3 = v_2 \ \sim v_3'$

Input			Output		
v_1	v_2	v_3	v_1'	v_2'	v_3'
0	0	0	0	0	0
0	0	1	(0,1)	0	0
0	1	0	0	0	1
0	1	1	(0,1)	0	1
1	0	0	(0,1)	0	0
1	0	1	(0,1)	(0,1)	0
1	1	0	(0,1)	0	1
1	1	1	(0,1)	(0,1)	1

Figure 2. One example of time delay Boolean network and its input/output.
doi:10.1371/journal.pone.0042095.g002

In this study, we will consider a much more generalized model by combining the structure of the above two models. If a Boolean function with one or several genes is a prerequisite for a target gene, then the induction of the Boolean function with input genes is necessary for the expression of the target gene. Hence, the target will be influenced by the Boolean function with several input genes. However, the induction of the Boolean function may not activate the target gene immediately, but at a future time. Therefore, the target gene may not have been influenced right now and we will treat these relationships as time delay affection. In this study, we will infuse these additional relationships for more generalized systems.

Boolean Network

Boolean networks were introduced by Kauffman (1969) forty years ago to represent genetic regulatory networks. First, we will review the definition of a Boolean network. A Boolean network $G(V,F)$ is a directed graph consisting of two components: a set of nodes $V=\{v_1,v_2,\ldots,v_n\}$ that corresponds to genes, and a list of Boolean functions $F=\{f_1,f_2,\ldots,f_n\}$ that corresponds to the rule of interaction and combination of several genes. For every node $v_i \in V$, its expression is simplified to two levels: on and off, representing a gene as active or inactive. For every Boolean function $f_i(v_{i_1},v_{i_2},\ldots,v_{i_k}) \in F$, k specified input nodes $v_{i_1},v_{i_2},\ldots,v_{i_k}$ are assigned to the node v_i in the graph and represent the rules of regulatory mechanisms between genes. The expression of a gene is determined by the expression of the gene directly affecting it with a Boolean function. Therefore, the state of each node $v_i \in V$ is determined by the Boolean function $f_i(v_{i_1},v_{i_2},\ldots,v_{i_k})$.

For each node v_i, the gene expression state at time t is assumed to take either 0 (not-expressed) or 1 (expressed) and is expressed as $\psi_t(v_i)$. In a Boolean network, every gene expression profile at time $t+1$ is completely determined by the expression profile of a set of genes $v_{i_1},v_{i_2},\ldots,v_{i_k}$ at time t and the corresponding Boolean function $f_i \in F$. That is, we can write $\psi_{t+1}(v_i)=f_i(\psi_t(v_{i_1}),\psi_t(v_{i_2}),\ldots,\psi_t(v_{i_k}))$.

Table 1. Count and probabilities table for v_j, v_h and $v_i^{'}$ assuming no misclassification error.

v'_i/v_jv_h	00	01	10	11	v'_i/v_jv_h	00	01	10	11
0	m_{000}	m_{010}	m_{100}	m_{110}	0	q_{000}	q_{010}	q_{100}	q_{110}
1	m_{001}	m_{011}	m_{101}	m_{111}	1	q_{001}	q_{011}	q_{101}	q_{111}

doi:10.1371/journal.pone.0042095.t001

For convenience, we converted the Boolean network $G(V,F)$ to the wiring diagram $G'(V',F')$ (See Figure 1) [37]. For each node $v_i \in V$, suppose $v_{i_1},v_{i_2},\ldots,v_{i_k}$ are the input nodes assigned to v_i. Then we construct an additional node $v_i^{'}$ and connected the edge from v_{i_j} to $v_{i'}$ for each $1 \leq j \leq k$. That is, the set of $\{v_1,\ldots,v_n\}$ represents the gene expression profile at time t and the set of $\{v_1^{'},\ldots,v_n^{'}\}$ corresponds to the gene expression profile at time $t+1$. Hence we can treat the set of $\{v_1,\ldots,v_n\}$ as the input values and the set of $\{v_1^{'},\ldots,v_n^{'}\}$ as the corresponding output values. Therefore, the output values of $\{v_1^{'},\ldots,v_n^{'}\}$ are determined by $v_i^{'}=f_i(v_{i_1},\ldots,v_{i_k})$.

The Structure of Time Delay Boolean Network

In the previous subsection, we found that given the values of the node (V) at time t, the expressions at time $t+1$ will be updated immediately by specific Boolean function (F). That is, for every gene $v_i \in V$, if the expression profile of a set of genes $\{v_{i_1},v_{i_2},\ldots,v_{i_k}\}$ at time t and the corresponding Boolean function f_i is observed, the gene expression of v_i at time $t+1$ is determined by $\psi_{t+1}(v_i)=f_i(\psi_t(v_{i_1}),\psi_t(v_{i_2}),\ldots,\psi_t(v_{i_k}))$. However, in real genetic regulatory situations, the deterministic system has been criticized due to the existence of misclassification error and noise. In addition, some of the gene expression may result in time delay when the gene is influenced by one or several input genes. That is, the induction of Boolean function may not activate the target gene immediately, but in the future. Hence, it would have been much more flexible to use a non-deterministic network system. In this subsection, we will consider two relationships between the Boolean function and the target gene instead of the deterministic relation.

First, we will introduce the structure of time delay Boolean networks. Suppose there are n elements, $v_1,v_2,\ldots,v_n$ in a Boolean network. For any elements v_i with specific Boolean function f_i, we have two kinds of pair wise relationship: prerequisite and similarity. We say that a Boolean function f_i with specific k input genes $v_{i_1},v_{i_2},\ldots,v_{i_k}$ at time t is the prerequisite for the target gene v_i at time $t+1$, if the on-status of Boolean function at time t is necessary for the on-status of gene v_i at time $t+1$. This relationship is denoted by $f_i(\psi_t(v_{i_1}),\psi_t(v_{i_2}),\ldots,\psi_t(v_{i_k}))\prec\psi_{t+1}(v_i)$. In other words, if the Boolean function f_i is not active at time t, gene v_i will be inactive at time $t+1$. If it does not cause confusion, we will omit the notation of ψ and input genes as denoted by $f_i \prec v_i$. Moreover, for every gene v_i, we use $\bar{v}_i$ as its dual (from 0 to 1, or from 1 to 0) in this paper. Therefore, for any Boolean function and target gene with a prerequisite relationship there are a total of two possible relationships: $f_i \prec v_i$ and $f_i \prec \bar{v}_i$. In this model, we do not consider the situation of a dual of Boolean function prerequisite to the target gene, that is $\bar{f}_i \prec v_i$ and $\bar{f}_i \prec \bar{v}_i$. Since for any Boolean function whose dual is a prerequisite to the target gene, there must exist another Boolean function that is a prerequisite to the target gene. For instance, if $\bar{f}_i(v_1,v_2)\prec v_3$, where $f_i(v_1,v_2)=(v_1 \; and \; v_2)$, then $f_i^{'}(v_1,v_2)\prec v_3$, where $f_i^{'}(v_1,v_2)=(\bar{v}_1 \; or \; \bar{v}_2)$. Therefore, for the prerequisite relationship, we only consider the Boolean function that is a prerequisite to target gene and the dual of target gene.

The other type of relationship between Boolean function and target gene is similarity. We say that the Boolean function f_i and

Table 2. Count profiles for the basic eight relationships without misclassification error.

$(v_j \textbf{ or } v_h)\prec v'_i$					$(v_j \textbf{ or } v_h)\prec\bar{v}'_i$				
v'_i/v_jv_h	00	01	10	11	v'_i/v_jv_h	00	01	10	11
0	+	+	+	+	0	0	+	+	+
1	0	+	+	+	1	+	+	+	+
$(v_j \textbf{ or } \bar{v}_h)\prec v'_i$					$(v_j \textbf{ or } \bar{v}_h)\prec\bar{v}'_i$				
$v_jv_hv'_i/$	00	01	10	11	$v_jv_hv'_i/$	00	01	10	11
0	+	+	+	+	0	+	0	+	+
1	+	0	+	+	1	+	+	+	+
$(\bar{v}_j \textbf{ or } v_h)\prec v'_i$					$(\bar{v}_j \textbf{ or } v_h)\prec\bar{v}'_i$				
$v_jv_hv'_i/$	00	01	10	11	$v_jv_hmiv'_i/$	00	01	10	11
0	+	+	+	+	0	+	+	0	+
1	+	+	0	+	1	+	+	+	+
$(\bar{v}_j \textbf{ or } \bar{v}_h)\prec v'_i$					$(\bar{v}_j \textbf{ or } \bar{v}_h)\prec\bar{v}'_i$				
$v_jv_hv'_i/$	00	01	10	11	$v_jv_hv'_i/$	00	01	10	11
0	+	+	+	+	0	+	+	+	0
1	+	+	+	0	1	+	+	+	+

doi:10.1371/journal.pone.0042095.t002

Table 3. Count and probabilities table for v_j, v_h and $v_{i'}$ with misclassification error.

v'_i/v_jv_h	00	01	10	11	v'_i/v_jv_h	00	01	10	11
0	n_{000}	n_{010}	n_{100}	n_{110}	0	r_{000}	r_{010}	r_{100}	r_{110}
1	n_{001}	n_{011}	n_{101}	n_{111}	1	r_{001}	r_{011}	r_{101}	r_{111}

doi:10.1371/journal.pone.0042095.t003

target gene v_i are similar if the status of the Boolean function and the status of the target gene are in the same expression, and we denoted this by $f_i \sim v_i$. In the same way, we do not consider the situation of Boolean function similar to the dual of target gene such as $f_i \sim \bar{v}_i$ in this study. Since if there is one Boolean function that is similar to the dual of target gene, there must exist another Boolean function that is similar to the target gene.

In the diagram, if a Boolean function f_i is a prerequisite to v_i, we draw a directed arrow from the vertex f_i to v_i and if f_i is similar to v_i, we use an undirected line to connect f_i and v_i.

In the model of time delay Boolean network we proposed, the output of the gene expression is not completely determined by the input state and Boolean function. The output expression may have more than one possible result in the time delay Boolean network. We illustrate the above construction by an example in Figure 2. It has three elements, one similarity and two prerequisite relationships. The possible outputs for every input state are listed in the right part of the graph. If we knew the network structure, some of the inputs would have more than one possible output expression in the time delay Boolean network.

Methods

Identification Algorithm

First, we only consider Boolean networks in which the maximum number of input genes is bounded by a constant K for every target gene, because it has been proven that the number of profiles required grows exponentially if K is not bounded [38]. For simplicity, we only show algorithms for the case of $K=2$. However, the algorithm can be intuitively generalized to any K in a straightforward way. For the inference of genetic network, we need to clarify the following questions for each target gene.

- Which input genes will affect the target gene?
- What kind of Boolean functions will be used for combining those input genes?
- What kind of relationship exists between the Boolean function and the target gene?

In this subsection, we propose an algorithm to clarify the above questions. The algorithm below is conceptually very simple since it simply uses output Boolean functions with input genes and relationships with target genes that are consistent with the data. First, for each output gene expression at time $t+1$ such as $v_i^{'}$, we consider all the pairs of elements in V at time t, for instance v_j and v_h. Then we count the eight incidents of $(v_j, v_h, v_i^{'})$ being (0,0,0), (0,0,1), …, (1,1,1) from the sample and arrange them in a 2×4 table; see the left part of Table 1. We mark a cell "+" if the count is positive and mark it "0" otherwise.

For detecting whether there exists a Boolean function which is a prerequisite to the target gene, we will compare the 2×4 output table with the left four basic relationships in Table 2. We consider the basic relationships to be consistent with the output table if the position of 0 cell in the basic relationships is also 0 in the output table. By comparing the output table with the four basic relationships, we can find relationships that are consistent with the output tables. If there is more than one relationship that is consistent with the output tables, we would use the Boolean logic gate "and" to combine the Boolean function and transfer the result to another Boolean function. Hence, the final Boolean function is the prerequisite to the target gene. Similarly, by comparing the 2×4 output table with the right four basic relations in Table 2, we could get another Boolean function which is the prerequisite to the dual of target gene.

Moreover, if only one Boolean function occurred in above relationship, that is, if there is only one Boolean function that is the prerequisite to the target gene or the dual of target gene, we will treat that relationship as our final relationship between the Boolean function and the target gene. However, if both of the two prerequisite relationships happened (i.e. $\exists f_i$ and f'_i $s.t.$ $f_i \prec v_i^{'}$ and $f'_i \prec \bar{v}_i^{'}$), we need to check whether these two relationships are in conflict. If the dual of f_i is equivalent to f'_i, our conclusion for inference will be that f_i is similar to the target gene (that is, $f_i \sim v_i^{'}$);

Table 4. Splitting counts caused by misclassification error.

v'_i/v_jv_h	00		01		10		11	
	$\mathbf{m_{000,000}}$	$\mathbf{m_{000,001}}$	$\mathbf{m_{010,000}}$	$\mathbf{m_{010,001}}$	$\mathbf{m_{100,000}}$	$\mathbf{m_{100,001}}$	$\mathbf{m_{110,000}}$	$\mathbf{m_{110,001}}$
0	$m_{000,010}$	$m_{000,011}$	$m_{010,010}$	$m_{010,011}$	$m_{100,010}$	$m_{100,011}$	$m_{110,010}$	$m_{110,011}$
	$m_{000,100}$	$m_{000,101}$	$m_{010,100}$	$m_{010,101}$	$m_{100,100}$	$m_{100,101}$	$m_{110,100}$	$m_{110,101}$
	$m_{000,110}$	$m_{000,111}$	$m_{010,110}$	$m_{010,111}$	$m_{100,110}$	$m_{100,111}$	$m_{110,110}$	$m_{110,111}$
	$m_{001,000}$	$m_{001,001}$	$m_{011,000}$	$m_{011,001}$	$m_{101,000}$	$m_{101,001}$	$m_{111,000}$	$m_{111,001}$
1	$m_{001,010}$	$m_{001,011}$	$m_{011,010}$	$m_{011,011}$	$m_{101,010}$	$m_{101,011}$	$m_{111,010}$	$m_{111,011}$
	$m_{001,100}$	$m_{001,101}$	$m_{011,100}$	$m_{011,101}$	$m_{101,100}$	$m_{101,101}$	$m_{111,100}$	$m_{111,101}$
	$m_{001,110}$	$m_{001,111}$	$m_{011,110}$	$m_{011,111}$	$m_{101,110}$	$m_{101,111}$	$m_{111,110}$	$m_{111,111}$

doi:10.1371/journal.pone.0042095.t004

Table 5. The eight basic relationships and their probabilistic hypotheses and *p*-scores.

Relation	Hypothesis	Scores
$(v_j\ or\ v_h)\prec\bar{v}'_i$	$q_{000}=0$	$p_{(v_j\ or\ v_h)\prec\bar{v}'_i}$
$(v_j\ or\ \bar{v}_h)\prec\bar{v}'_i$	$q_{010}=0$	$p_{(v_j\ or\ \bar{v}_h)\prec\bar{v}'_i}$
$(\bar{v}_j\ or\ v_h)\prec\bar{v}'_i$	$q_{100}=0$	$p_{(\bar{v}_j\ or\ v_h)\prec\bar{v}'_i}$
$(\bar{v}_j\ or\ \bar{v}_h)\prec\bar{v}'_i$	$q_{110}=0$	$p_{(\bar{v}_j\ or\ \bar{v}_h)\prec\bar{v}'_i}$
$(v_j\ or\ v_h)\prec v'_i$	$q_{001}=0$	$p_{(v_j\ or\ v_h)\prec v'_i}$
$(v_j\ or\ \bar{v}_h)\prec v'_i$	$q_{011}=0$	$p_{(v_j\ or\ \bar{v}_h)\prec v'_i}$
$(\bar{v}_j\ or\ v_h)\prec v'_i$	$q_{101}=0$	$p_{(\bar{v}_j\ or\ v_h)\prec v'_i}$
$(\bar{v}_j\ or\ \bar{v}_h)\prec v'_i$	$q_{111}=0$	$p_{(\bar{v}_j\ or\ \bar{v}_h)\prec v'_i}$

doi:10.1371/journal.pone.0042095.t005

otherwise, we will treat it as if there is no relationship between the input genes and the target gene because we did not gather enough information to judge true relationships between $v_i^{'}$ and (v_j,v_h) at this moment. By the above identification procedure, we can find the corresponding input genes, Boolean function and their relationship for every target gene.

Identification Algorithm with Noisy Array

In previous subsection, we discussed an identification method for data without noise. In this section we will consider the situation of noisy array data. We assume that every element in the entry of $(I_{ij},\ O_{ij})$, $j=1,2,\ldots,m$ switches to its reverse status with a misclassification probability p independently; that is

$$I_{ij}^{*}=\begin{cases} I_{ij} & \text{with probability } 1-p; \\ 1-I_{ij} & \text{with probability } p. \end{cases} \tag{1}$$

$$O_{ij}^{*}=\begin{cases} O_{ij} & \text{with probability } 1-p; \\ 1-O_{ij} & \text{with probability } p. \end{cases} \tag{2}$$

Thus, the observed array $(I_{ij}^{*},\ O_{ij}^{*})$ contains misclassification error. Our goal is to reconstruct time delay Boolean network from noisy array of binary data (I_{ij}^{*},O_{ij}^{*}).

Similar to section 2, we assume that the maximum number of input genes is bounded by 2 for every target gene. We treat the data in the 2×4 table as a multinomial distribution with eight cells whose probabilities are $q_{000},q_{001},\ldots,q_{111}$ as shown in the right part of Table 1, where $q_{000}+q_{001}+\ldots+q_{111}=1$. Similarly, we extract the data with misclassification error for every output gene and each pair of input genes as the 2×4 table. Now the observed data $n_{000},n_{001},\ldots,n_{111}$ are not generated from the multinomial $q_{000},q_{001},\ldots,q_{111}$, but from another multinomial $r_{000},r_{001},\ldots,r_{111}$ as shown in Table 3, where $r_{000}+r_{001}+\ldots+r_{111}=1$.

Because of the misclassification error, a portion of the samples of m_{000} may change to the other seven cells. We use the notations of $m_{000,000},m_{000,001},\ \ldots,m_{000,111}$ to represent the counts of eight cells changed from m_{000}. Analogous notations are defined for $m_{001},m_{010},\ldots,m_{111}$. The splitting is shown in Table 4. Consequently, the generated probabilities $(q_{000},q_{001},\ldots,q_{111})$ are calculated as follows: $q_{i_1i_2i_3,j_1j_2j_3}=p^{I(i,j)}(1-p)^{3-I(i,j)}q_{i_1i_2i_3}$, where $I(i,j)=\sum_{k=1}^{3}|i_k-j_k|$. Here, we adopt the notation $q_{i_1i_2i_3,j_1j_2j_3}$ analogous to $m_{i_1i_2i_3,j_1j_2j_3}$. The above parameters and splits are shown in Table 4. In the table, it is easy to find that the correspondence between two sets of counts and probabilities is the following:

$$\begin{cases} n_{j_1j_2j_3}=\sum\limits_{i_1,i_2,i_3=0,1} m_{i_1i_2i_3,j_1j_2j_3} \\ r_{j_1j_2j_3}=\sum\limits_{i_1,i_2,i_3=0,1} q_{i_1i_2i_3,j_1j_2j_3} \end{cases} \tag{3}$$

and

$$\begin{cases} m_{i_1i_2i_3}=\sum\limits_{j_1,j_2,j_3=0,1} m_{i_1i_2i_3,j_1j_2j_3} \\ q_{i_1i_2i_3}=\sum\limits_{j_1,j_2,j_3=0,1} q_{i_1i_2i_3,j_1j_2j_3} \end{cases}$$

For the complete data $\{m_{i_1i_2i_3,j_1j_2j_3}\}$, the log-likelihood is given by

$$L=\sum_{i_1,i_2,i_3,j_1,j_2,j_3=0,1} m_{i_1i_2i_3,j_1j_2j_3}\log q_{i_1i_2i_3,j_1j_2j_3} \tag{4}$$

where $q_{i_1i_2i_3,j_1j_2j_3}$ are those splitting probabilities. Since the complete data $\{m_{i_1i_2i_3,j_1j_2j_3}\}$ are not observable, we use the EM algorithm to maximize the log-likelihood. In the E-step, the splitting counts of complete data $\{m_{i_1i_2i_3,j_1j_2j_3}\}$ are evaluated by the conditional expectations using the current values of the parameters by the following formula

$$E_{p,q_{000},q_{001},\ldots,q_{111}}(m_{i_1i_2i_3,j_1j_2j_3}|n_{j_1j_2j_3})=\frac{n_{j_1j_2j_3}q_{i_1i_2i_3,j_1j_2j_3}}{\sum\limits_{i'_1i'_2i'_3=0,1} q_{i'_1i'_2i'_3,j_1j_2j_3}} \tag{5}$$

where $i_1,i_2,i_3,j_1,j_2,j_3=0,1$. One probabilities of $q_{000},q_{001},\ldots,q_{111}$ are zero in those different hypotheses specified in Table 5. In the M-step, we maximize the conditional expectation of the log-likelihood for the complete data to obtain the maximum likelihood estimates (MLEs) of the parameters. According to the MLEs, we can compute the *p*-score for every pair of input genes and target gene, which are obtained by estimating for the misclassification probability under every prerequisite relationship.

For the first step, we would like to determine the most probable relationships between every pair of input genes and one output gene. Next, we find the most probable Boolean function with pair input genes for every output gene and select candidate pairs of input genes and output gene for the watch list. Finally, we reconstruct a time delay Boolean network by integrating the relationship of those genes selected.

For one output gene $v_i^{'}$ and a pair of input genes v_j and v_k, we define the *p*-scores $p_{(v_j\ or\ v_k)\prec\bar{v}'_i}$, $p_{(v_j\ or\ \bar{v}_k)\prec\bar{v}'_i}$, $p_{(\bar{v}_j\ or\ \bar{v}_k)\prec\bar{v}'_i}$, $p_{(v_j\ or\ v_k)\prec v'_i}$, $p_{(\bar{v}_j\ or\ v_k)\prec v'_i}$, $p_{(v_j\ or\ \bar{v}_k)\prec v'_i}$, $p_{(\bar{v}_j\ or\ \bar{v}_k)\prec v'_i}$ are, respectively, the maximum likelihood estimates of p under the triangular model: $q_{000}=0$, $q_{010}=0$, $q_{100}=0$, $q_{110}=0$, $q_{001}=0$, $q_{011}=0$, $q_{101}=0$, $q_{111}=0$.

According to the EM algorithm described above, we can evaluate the *p*-score for every output gene. We use the MLE $\hat{p}$ to measure how well each hypothesis fits: the smaller the score is, the more likely that the corresponding hypothesis could be true.

If the samples are generated from a time delay Boolean network, *p*-score are quite useful for the discovery of true

Table 6. By the time delay Boolean network in Figure 1, we generate 100 samples with p = 0.05.

Samples		**Hypotheses**								**Relation**
Input	**Output**	$q_{000}=0$	$q_{010}=0$	$q_{100}=0$	$q_{110}=0$	$q_{001}=0$	$q_{011}=0$	$q_{101}=0$	$q_{111}=0$	
v_1,v_2	v_1'	0.493	0.418	0.273	0.379	0.148	0.178	0.372	0.343	
v_1,v_3	v_1'	0.438	0.147	0.248	0.222	0.016	0.245	0.182	0.241	$(v_1 \text{ or } v_3)\prec v'_1$
v_2,v_3	v_1'	0.318	0.260	0.571	0.214	0.189	0.293	0.138	0.374	
v_1,v_2	v_2'	0.326	0.300	0.304	0.297	0.091	0.092	0.232	0.209	
v_1,v_3	v_2'	0.338	0.216	0.349	0.197	0.039	0.069	0.038	0.243	$(v_1 \text{ and } v_3)\prec v'_2$
v_2,v_3	v_2'	0.326	0.253	0.390	0.174	0.052	0.141	0.017	0.169	
v_1,v_2	v_3'	0.211	0.011	0.355	0.029	0.040	0.228	0.011	0.294	
v_1,v_3	v_3'	0.338	0.290	0.402	0.734	0.669	0.291	0.379	0.360	$v_2\sim v'_3$
v_2,v_3	v_3'	0.247	0.312	0.030	0.011	0.039	0.011	0.283	0.241	

doi:10.1371/journal.pone.0042095.t006

relationships. Here we can consider the *maximum compatibility criterion*: to choose the maximum threshold value so that the selected relationships contain no conflicts [34]. We collect those relationships whose p-scores are smaller than a threshold. Known biological results are helpful for the determination of a threshold. For example, if we know the relationship $(v_1 \text{ or } v_2)\prec v_3$ is true, then the p-scores smaller than $p_{(v_1 \text{ or } v_2)\prec v_3}$ should be in our watch list. As more relationships are included in the watch list, the more likely we are to observe incompatible ones. In general, we can choose the threshold that allows the maximum number of relationships with no conflicting relationships. Next we will demonstrate the method by illustration examples.

Results and Discussion

Theoretical Results

First, we will analyze the number of input/output pairs required for the network reconstruction of time delay Boolean network to be unique. The theoretical results of classical Boolean networks only consider the similar relationship [27,38,39]. The following results prove the theoretical results time delay Boolean networks that has a more flexible structure and consider both similar and prerequisite relationship.

Proposition 1. *For all subsets of V with $2K$ genes, if all assignments (i.e., 2^{2K} assignments) of Boolean values appear in input expression patterns and all of its possible output expression patterns of the target gene are present, the identification of genetic network is determined to be unique, if it exists.*

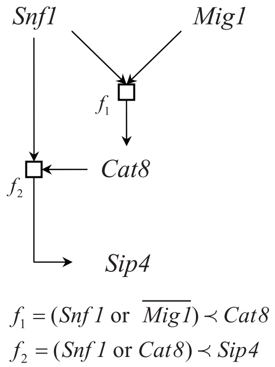

Snf1
Mig1
f_1
Cat8
f_2
Sip4
$f_1 = (Snf1 \text{ or } \overline{Mig1}) \prec Cat8$
$f_2 = (Snf1 \text{ or } Cat8) \prec Sip4$

Figure 3. Network reconstruct from the expression data of yeast Saccharomyces cerevisiae.
doi:10.1371/journal.pone.0042095.g003

(Proof) Let z be any gene in V and suppose z is controlled by a Boolean function $g(x_{i_1},\ldots,x_{i_k})$ with similarity or prerequisite relationship (i.e., $g\sim z$ or $g\prec z$). If the Boolean function g is similar to z, the case is proved for the classical Boolean networks in Akutsu et al. (1998). Next, we consider the case of Boolean function g as a prerequisite to z. In this case, there must exist a specific input value $\{a_1,\ldots,a_k\}$ for $\{x_{i_1},\ldots,x_{i_k}\}$ such that z have two possible values 0 and 1. Hence, any other genes would not control z because all assignments of Boolean values are appearance. Let us illustrate the above statement by the example for the case of $K=1$ and $K=2$. If $K=1$ and $x\prec z$, when the input of x is 1, the outcome of z being both 0 and 1 will appearance. Therefore, given the input of $x=1$, the outcome of z is not deterministic no matter the value of any other gene y is 1 or 0. Hence, any other gene y would not affect gene z. If $K=2$ and $g(x_1,x_2)\prec z$ for some Boolean function g, there must exist an input (a_1,a_2) such that $g(a_1,a_2)=1$. Then, for any other pair of gene $\{y_1,y_2\}$ where $\{y_1,y_2\}\cap\{x_1,x_2\}=\phi$, the outcome of z is not deterministic for any input of $\{y_1,y_2\}$, if the input of $\{x_1,x_2\}$ is $\{a_1,a_2\}$. In a case of $\{y_1,y_2\}\cap\{x_1,x_2\}\neq\phi$, we can prove that gene y_i which does not belong to $\{x_1,x_2\}$ would not affect the gene z in a similar way.

Proposition 2. *The probability that one sub-assignment with all of its possible results in the target gene does not appear among m random input expression pattern is at most $2(1-\frac{1}{2^{2K+1}})^m$.*

(Proof) For any fixed set of nodes $\{v_{i_1},v_{i_2},\ldots,v_{i_{2K}}\}$, the probability that a sub-assignment $v_{i_1}=v_{i_2}=\ldots=v_{i_{2K}}=1$ does not appear in one random input expression pattern is $1-\frac{1}{2^{2K}}$. Thus, among the m random input expressions, the probability that $v_{i_1}=v_{i_2}=\ldots=v_{i_{2K}}=1$ appears is t times is equal to $\frac{m!}{t!(m-t)!}(\frac{1}{2^{2K}})^t(1-\frac{1}{2^{2K}})^{m-t}$, where $t\leq m$. In addition, the probability that all of the possible results in the target gene does not appear among t times input is smaller than $(\frac{1}{2})^{t-1}$ for $1\leq t\leq m$ and equal to 1 for $t=0$. Hence the probability that one sub-assignment and all of its possible results does not appear among m random input expression is smaller than $(1-\frac{1}{2^{2K}})^m+\sum_{t=1}^{m}\frac{m!}{t!(m-t)!}(\frac{1}{2^{2K}})^t(1-\frac{1}{2^{2K}})^{m-t}(\frac{1}{2})^{t-1}$, and this can be bounded by $2(1-\frac{1}{2^{2K+1}})^m$ by an algebra calculation.

Next we prove the main theorem.

Theorem 1. *For the identification of one time delay Boolean network of n nodes with maximum indegree* $\leq K$, $O(2^{2K+1}\cdot(2K+\alpha)\cdot\log n)$ *uniformly and randomly sampled input patterns are sufficient for exact inference with probability at least* $1-\frac{1}{n^{\alpha}}$.

(Proof) We consider the probability that the condition of Proposition 1 is not satisfied under m random input expression patterns.

By Proposition 2, the probability that $v_{i_1}=v_{i_2}=\ldots=v_{i_{2K}}=1$ with all of its possible results in the target gene does not appear among the m random input expression patterns is bounded by $2(1-\frac{1}{2^{2K+1}})^m$ for any fixed set of nodes $\{v_{i_1},v_{i_2},\ldots,v_{i_{2K}}\}$. Since the number of combinations of $2K$ nodes from a set of n possibilities is bounded by $2^{2K}\cdot n^{2K}$, the probability that the condition of Proposition 1 is not satisfied is at most $2^{2K}\cdot n^{2K}\cdot 2(1-\frac{1}{2^{2K+1}})^m$. It is not difficult to see that $2^{2K}\cdot n^{2K}\cdot 2(1-\frac{1}{2^{2K+1}})^m<p$ holds for $m>\ln 2\cdot 2^{2K+1}\cdot(2K+1+2K\log n+\log\frac{1}{p})$. Hence, we obtain the theorem by letting the non-identification probability $p=\frac{1}{n^{\alpha}}$.

Next we develop an information theoretic lower bound on the number of input/output pairs needed for the identification of a time delay Boolean network.

Theorem 2. *If the maximum indegree* $\leq K$, *at least* $\Omega(2^K+K\log n)$ *input/output pairs are required for the identification of a time delay Boolean network in the worst case.*

(Proof) The number of time delay Boolean networks is given by all the possible combination of Boolean function with k nodes from a set of n possibilities with all possible relations between Boolean functions with target node. Since there are $\Omega(n^K)$ possible combinations of input nodes, 2^{2^K} possible Boolean functions and 3 possible relations between Boolean function with each node, there are $\Omega((2^{2^K}\cdot n^K\cdot 3)^n)$ Boolean networks whose maximum indegree is at most K. On the other hand, there are at most 2^n possible output patterns with one input expression pattern. Therefore, $\Omega(\log_{2^n}((2^{2^K}\cdot n^K\cdot 3)^n))$ which is the same as $\Omega(2^K+K\log n)$ input/output pairs are required in the worst case.

Example with Simulation and Real Data

We will illustrate our method by the example described in Figure 2. For the pair of samples consist of three elements list in the right part of Figure 2, we uniformly generated 100 input samples and their corresponding possible output samples with misclassification probability $p=0.05$. For the prerequisite relationship, if the status of Boolean function with input genes is on, then we allow the output value to have equal probability of on or off. The data can be arranged as input/output sample similar to that obtained from the microarray data with time. Namely, the input of each sample can represent the gene expression at time t and the output can represent the gene expression at time $t+1$. For each pair of input and output genes, we compute the 8 basic p-scores that represent the 8 basic hypotheses in Table 5 for all of pair input genes and output genes. After the calculation, the simulation results of every p-score are listed in Table 6.

Beside the example with 3 elements, in order to shows the superiority of the proposed method can be applied to a larger network, a more comprehensive example with a larger network is given in Figure S1.

Next, we have to decide the threshold for choosing the relations. When we increase the threshold of the p-score, the relations whose p-score are smaller than the threshold will be chosen. Moreover, when the number is 0.138, the conflict occurs, since we have $(v_1\ or\ v_3)\prec v'_1$ and $(\bar{v}_2\ or\ v_3)\prec v'_1$. However, in our model, there are at most two genes that would affect an output gene. Therefore, we can choose 0.138 as our threshold and include relations whose p-score is smaller than the threshold. By these procedures, we can reconstruct the time delay Boolean network identical to Figure 2.

In the area of gene regulatory network study, Schuller has summarized regulatory cis-acting elements of structural genes of the nonfermentative metabolism and described the molecular interactions among general regulators and pathway-specific factors [40]. In the gene regulation of gluconeogenesis by Sip4 and Cat8 pathway, the carbon source control could be identified for the regulator Cat8; see (Figure 6) in Schuller [40]. In this study, we apply our proposed approach to explore the expression profiles and show some exploratory result on the Cat8 pathway.

In order to demonstrate the effectiveness of reconstruction, we use the microarray expression dataset of yeast *Saccharomyces cerevisiae* produced by DeRisi et al. [1] and Spellman et al. [41]. In total, the data is comprised of 41 experiments after filtering out experiments with missing values. By these experimental microarray data sets, we can use our proposed method to reconstruct the biological pathway and the genetic regulation network result is shown in Figure 3. The result is consistent with the genetic network in the literature. That is, the restraint of Mig1 or activation of Snf1 is a prerequisite for the decreasing of Cat8. Moreover, the restraint of Snf1 or Cat8 is a prerequisite for the decreasing of Mls1. However, the negative similarity between Snf1 and Mig1 is undetectable in our current model.

Conclusions

In this paper, we have introduced the model of time delay Boolean network that generalizes the Boolean network model in order to cope with dependencies that have two kinds of relationships: similarity and prerequisite. The approach for reconstruction of genetic network inference from gene expression data relies on the assumption that the expression of a gene is likely to be controlled by a relatively small number (say k) of genes. Also, some bounds on the size of data required for the identification of the time delay Boolean networks under constant of indegree are stated and discussed. Moreover, the algorithm of the network reconstruction from noisy array data is developed.

One characteristic of a Boolean network is that all the variables in the graph are binary. If the data we observed is continuous or quantized to have more than two levels, we need to discretize them. For microarray data, the ratios of expression level would be one possible approach of discretization. That is, we can treat the gene as on (active) if the log-ratio of its expression is larger than zero. We treat it as off (inactive) otherwise. In general, biological background knowledge will be helpful for setting thresholds for discretizaion. On the other hand, if the samples are obtained from a time course, then we can consider the gene as on or off by detecting whether the gene is either increasing or decreasing with time.

The work in progress is aimed at evaluating the effectiveness of the described approach for inferring genetic networks from biological gene expression time series data. Besides that, implementation on some other real biological data is also an important task.

For the implement of the network reconstruction algorithm, the greatest complexity is the computation of p-score for each of the

$\frac{n!}{k!(n-k)!}$ input elements and n output elements, where n is the number of elements and k is the number of indegree. It is an iterative algorithm to compute the MLE for the p-scores by EM procedure while the common practice is to set an upper bound for iterations in numerical implementation. Consequently, this keeps the $O(n^{k+1})$ complexity for the computation of MLE. In addition, the sorting algorithm for the $\frac{n!}{k!(n-k)!}n$ data cost $O(n^{k+1}\log n)$ in terms of time. Hence, the overall time complexity for the network reconstruction is $O(n^{k+1}\log n)$ for this algorithm.

Supporting Information

Figure S1 An example of genetic network with 8 nodes. (PDF)

Acknowledgments

We thank the editor and reviewers for their constructive comments.

Author Contributions

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

References

1. DeRisi JL, Iyer VR, Brown PO (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. Science 278: 680–686.
2. Bornholdt S (2005) Less is more in modeling large genetic networks. Science 310: 449–451.
3. Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. Proc Natl Acad Sci U S A 95: 14863–14868.
4. Tzeng J, Lu HH-S, Li WH (2008) Multidimensional scaling for large genomic data sets. BMC Bioinformatics 9: 179.
5. Rawool S, Venkatesh K (2007) Steady state approach to model gene regulatory networks–simulation of microarray experiments. Biosystems 90: 636–655.
6. Jensen FV (1996) An introduction to Bayesian networks. London: University College London Press.
7. Jensen FV (2001) Bayesian networks and decision graphs. New York: Springer.
8. Moler EJ, Radisky DC, Mian IS (2000) Integrating naive bayes models and external knowledge to examine copper and iron homeostasis in s. cerevisiae. Physiol Genomics 4: 127–135.
9. Pearl J (1988) Probabilistic reasoning in intelligent systems: networks of plausible inference. San Mateo: Morgan Kaufmann.
10. Needham C, Bradford J, Bulpitt A, Westhead D (2007) A primer on learning in Bayesian networks for computational biology. PLoS Comput Biol 3: e129.
11. Reynolds S, Käll L, Riffle M, Bilmes J, Noble W (2008) Transmembrane topology and signal peptide prediction using dynamic Bayesian networks. PLoS Computational Biology 4: e1000213.
12. Friedman N, Linial M, Nachman I, Pe'er D (2000) Using bayesian networks to analyze expression data. Journal of Computational Biology 7: 601–620.
13. Imoto S, Higuchi T, Goto T, Tashiro K, Kuhara S, et al. (2004) Combining microarrays and biological knowledge for estimating gene networks via bayesian networks. Journal of Bioinformatics and Computational Biology 2: 77–98.
14. Scharfe C, Lu HH-S, Neuenburg JK, Allen EA, Li GC, Klopstock T, Cowan TM, Enns GM, Davis RW (2009) Mapping gene associations in human mitochondria using clinical disease phenotypes. PLoS Computational Biology 5: e1000374.
15. Liu T, Sung W, Mittal A (2006) Learning gene network using time-delayed bayesian network. International Journal of Artificial Intelligence Tools 15: 353–370.
16. Friedman N, Nachman I, Pe'er D (1999) Learning bayesian network structure from massive datasets: the sparse candidate algorithm. In: Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence. 206–215.
17. Heckerman D, Geiger D, Chickering DM (1995) Learning bayesian networks: the combination of knowledge and statistical data. Machine Learning 20: 197–243.
18. Balakrishnan S, Madigan D (2006) A one-pass sequential Monte Carlo method for Bayesian analysis of massive datasets. Bayesian Analysis 1: 345–362.
19. Huang S (1999) Gene expression profiling, genetic networks and cellular states: An integrating concept for tumorigenesis and drug discovery. Journal of Molecular Medicine 77: 469–480.
20. Shmulevich I, Gluhovsky I, Hashimoto RF, Dougherty ER, Zhang W (2003) Steady-state analysis of genetic regulatory networks modelled by probabilistic Boolean networks. Comparative and Functional Genomics 4: 601–608.
21. Kim H, Lee JK, Park T (2007) Boolean networks using the chi-square test for inferring large-scale gene regulatory networks. BMC Bioinformatics 8: 37.
22. Zhang R, de SCavalcante HLD, Gao Z, Gauthier DJ, Socolar JES, et al. (2009) Boolean chaos. Phys Rev E 80: 045202.
23. Socolar JES, Kauffman SA (2003) Scaling in ordered and critical random Boolean networks. Physical Review Letters 90: 68702.
24. Dealy S, Kauffman SA, Socolar JES (2005) Modeling pathways of differentiation in genetic regulatory networks with boolean networks: Research articles. Complex 11: 52–60.
25. Veliz-Cuba A, Stigler B (2011) Boolean models can explain bistability in the lac operon. Journal of Computational Biology 18: 783–794.
26. Kauffman SA (1969) Metabolic stability and epigenesis in randomly constructed genetic nets. Journal of Theoretical Biology 22: 437–467.
27. Akutsu T, Miyano S (1999) Identification of genetic networks from a small number of gene expression patterns under the boolean network model. Proc Pacific Symposium on Biocomputing : 17–28.
28. Ideker T, Thorsson V, Karp R (2000) Discovery of regulatory interactions through perturbation: inference and experimental design. Proc Pacific Symposium on Biocomputing : 302–313.
29. Allocco DJ, Kohane IS, Butte AJ (2004) Quantifying the relationship between co-expression, coregulation and gene function. BMC Bioinformatics 5: 18.
30. Bosl WJ (2007) Systems biology by the rules: hybrid intelligent systems for pathway modeling and discovery. BMC Systems Biology 1: 13.
31. Jordan IK, Marino-Ramirez L, Wolf YI, Koonin EV (2004) Conservation and coevolution in the scale-free human gene coexpression network. Molecular Biology and Evolution 21: 2058–2070.
32. Lee HK, Hsu AK, Sajdak J, Qin J, Pavlidis P (2004) Coexpression analysis of human genes across many microarray data sets. Genome Research 14: 1085–1094.
33. Opgen-Rhein R, Strimmer K (2007) From correlation to causation networks: a simple approximate learning algorithm and its application to high-dimensional plant gene expression data. BMC Systems Biology 1: 37.
34. Li LM, Lu HH-S (2005) Explore biological pathways from noisy array data by directed acyclic boolean networks. Journal of Computational Biology 12: 170–185.
35. Sahoo D, Dill D, Gentles A, Tibshirani R, Plevritis S (2008) Boolean implication networks derived from large scale, whole genome microarray datasets. Genome Biology 9: R157.
36. Wang H, Lu HH-S, Chueh TH (2011) Constructing biological pathway by a two-step counting approach. Plos one 6: e20074.
37. Somogyi R, Sniegoski CA (1996) Modeling the complexity of genetic networks: Understanding multigene and pleiotropic regulation. Complexity 1: 45–63.
38. Akutsu T, Kuhara S, Maruyama O, Miyano S (1998) Identification of gene regulatory networks by strategic gene disruptions and gene overexpression. Proc 9th ACM-SIAM Symp Discrete Algorithms: 695–702.
39. Akutsu T, Kuhara S, Maruyama O, Miyano S (2003) Identification of genetic networks by strategic gene disruptions and gene overexpressions under a boolean model. Theoretical Computer Science 298: 235–251.
40. Schuller HJ (2003) Transcriptional control of nonfermentative metabolism in the yeast saccharomyces cerevisiae. Current Genetics 43: 139–160.
41. Spellman PT, Sherlock G, Zhang MQ, Iyer VR, Anders K, et al. (1998) Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization. Molecular Biology of Cell 9: 3273–3297.