Inference of gene regulatory networks with the strong-inhibition Boolean model

Qinzhi Xia\textsuperscript{1}, Lulu Liu\textsuperscript{1}, Weiming Ye\textsuperscript{1,2} and Gang Hu\textsuperscript{1,3}

\textsuperscript{1} Department of Physics, Beijing Normal University, Beijing 100875, People’s Republic of China
\textsuperscript{2} Center for Complexity Research, Department of Systems Science, School of Management, Beijing Normal University, Beijing 100875, People’s Republic of China
E-mail: ganghu@bnu.edu.cn

\textit{New Journal of Physics 13} (2011) 083002 (15pp)
Received 5 May 2011
Published 3 August 2011
Online at http://www.njp.org/
doi:10.1088/1367-2630/13/8/083002

\textbf{Abstract.} The inference of gene regulatory networks (GRNs) is an important topic in biology. In this paper, a logic-based algorithm that infers the strong-inhibition Boolean genetic regulatory networks (where regulation by any single repressor can definitely suppress the expression of the gene regulated) from time series is discussed. By properly ordering various computation steps, we derive for the first time explicit formulae for the probabilities at which different interactions can be inferred given a certain number of data. With the formulae, we can predict the precision of reconstructions of regulation networks when the data are insufficient. Numerical simulations coincide well with the analytical results. The method and results are expected to be applicable to a wide range of general dynamic networks, where logic algorithms play essential roles in the network dynamics and the probabilities of various logics can be estimated well.
1. Introduction

A gene regulatory network (GRN) [1, 2] consists of a large number of DNA, RNA, proteins and other molecular entities with complicated interactions. The genes influence each other with their expression products in the structure of complex networks [3]. Since a huge number of data is available in GRNs while the detailed interaction connections are still unknown in many cases, the inference of genetic network structure has become a very important topic. A number of studies focused on the reconstruction of general networks [4–7], especially on the inference of GRNs [8–13]. A plethora of methods [14–23] have been proposed for exploring the network structures from different kinds of biological experimental data, such as time series and steady-state data. The development of this field is crucial for understanding the function, stability and evolution of biological systems. However, as pointed in [24], the inference problem is still far from being solved. For instance, one problem of great significance in practice, namely with how large a probability can an interaction be correctly inferred using a given number of measured data, especially when the data are insufficient, has scarcely been touched analytically. This is the central task of the present paper [25].

Different inference methods are based on different network models. Some commonly used methods adopt approximations of quantized variables and time, and infer GRNs with Boolean networks [25–27]. Since the seminal work of Kauffman [28], Boolean networks have been extensively used to depict GRNs [29–31]. In spite of its simplicity, in many cases it can catch the essence of regulations [32]. The results obtained in these simplified models can be used as starting bases to study more general and practical systems with continuous time and variables (usually described by coupled ordinary differential equations). In this paper, we follow this development to propose a logic-based Boolean algorithm in our inference analysis. We consider the problem of how to reconstruct the GRNs from evolution data arbitrarily given with a known regulatory rule. In particular, we focus on a threshold network model widely used in GRN analysis: the strong-inhibition Boolean (SIB) model [30, 31], which in fact is one case of the so-called nested canalyzing function introduced recently in biology [33, 34]. In SIB models, regulation by any single repressor of high density can definitely suppress the expression of the gene regulated. This rule represents some typical biological transcriptional regulatory processes [35–37]. The key result of this paper is that we are able to derive analytic formulae for dealing with the inference problem. We not only determine the inferences by various logics, but also compute analytically the probabilities for different logics to occur. Therefore, we answer a different question that is of great importance when the data are not
enough to determine the whole network: With a given amount of data randomly chosen, with how much probability can one succeed in correctly predicting different types of interactions? Furthermore, we obtain exact forms analytically computing these probabilities by correctly ordering the computations, step by step, for different types of regulations. These results can be used to estimate the necessary number of measured data and experimental costs for the given tasks of GRN inference with required prediction precision. All the analytical predictions are justified perfectly by numerical simulations. Moreover, although we consider only a Boolean model with specific strong inhibition dynamics, the ideas and methods can be extended to all general complex systems in which a logic algorithm (i.e. Boolean maps (BMs)) is important for the system dynamics and the probabilities of various logics are analytically computable. Some systems of practical importance may fall into this class, such as more general genomic regulatory networks, some neural networks and epidemic networks.

2. Model and some concepts

First let us make clear our model and classify different gene sets in inferring the network. In the SIB model, the state of node $i$ at time $t$ is denoted by $S_i(t)$. Each node $i$ has only two possible states, on-state ($S_i = 1$) or off-state ($S_i = 0$), and its state at time $t + 1$ is determined by the state of the network at time $t$ via the SIB rule:

$$S_i(t+1) = \begin{cases} 
1, & \sum_j a_{ij} S_j(t) > 0, \\
0, & \sum_j a_{ij} S_j(t) < 0, \\
S_i(t), & \sum_j a_{ij} S_j(t) = 0,
\end{cases}$$  \hspace{1cm} (1)$$

where $a_{ij} = 1$ if node $j$ activates node $i$, $a_{ij} = -\infty$ if $j$ represses $i$ and $a_{ij} = 0$ if $j$ does not regulate $i$. This rule is inspired by the observation that for a given gene its repressive interactions often dominate the activations in some important genomic regulations [30, 35, 37]. As in [36], we do not consider self-regulation in the networks, i.e. $a_{ii} = 0$ for all nodes in our present discussion as in figure 1.

The structure of GRN can be inferred from the expression of genes. We denote the state of the whole GRN at time $t$, $(S_1(t), S_2(t), \ldots, S_N(t))^T$, by a vector pattern $S(t)$. Under the SIB mechanism, one pattern $S(t)$ of a given GRN will evolve to another pattern $S(t+1)$. $S(t)$ and $S(t+1)$ make up a pair of input/output data which can be used to infer the GRN. For example, if in an input pattern only node $i$ and node $j$ stay at on-state, i.e. $S_i(t) = 1$, $S_j(t) = 1$ and $S_k(t) = 0$ ($k \neq i, j$), whereas in the corresponding output pattern node $i$ switches to off-state, i.e. $S_i(t+1) = 0$, then from this pair of input/output data, we can make a logic derivation that $j$ has repressive regulation on $i$. Given enough pairs of data, it is possible to identify all links existing in GRN. Generally speaking, if we have all the necessary data (maximum $2^N$ different sets of fully chosen data) this inference process can be implemented using Boolean algebra [31]. A practical question is: Given a certain number of data randomly chosen that is much less than the above maximum number, to what extent can we reconstruct the networks? Because the inference from an arbitrary number of data is much more complicated than the example above, we propose an analytic method to simplify this logic-base inference.
Figure 1. Schematic representation of an SIB GRN. Here we prune all the self-regulations of the cell-cycle network of the budding yeast [29] as an example consisting of 11 nodes. The solid links with arrows represent active regulations, while the dotted ones represent repressive regulations.

Our basic idea is the partition of the whole GRN as in figure 2 and this partition will be used to order the computations of different logics in the next section. The symbols used in figure 2 and in this paper are listed below:

\( N \): the number of the nodes in the whole GRN;
\( q \): the number of input/output pairs of patterns used in inference;
\( A \): the aim gene;
\( \tilde{\alpha} \): the set whose elements activate \( A \) actually;
\( \tilde{\beta} \): the set whose elements repress \( A \) actually;
\( \tilde{S}_\alpha \): the set whose elements are suspected to activate \( A \);
\( \tilde{S}_\beta \): the set whose elements are suspected to repress \( A \);
\( \tilde{E}_\alpha \): the set whose elements can be inferred to definitely not activate \( A \);
\( \tilde{E}_\beta \): the set whose elements can be inferred to definitely not repress \( A \);
\( \tilde{I}_\alpha \): the set whose elements can be identified to definitely activate \( A \);
\( \tilde{I}_\beta \): the set whose elements can be identified to definitely repress \( A \);
\( \tilde{D}_\alpha \): the set whose elements in fact do not activate \( A \) but remain in \( \tilde{S}_\alpha \);
\( \tilde{D}_\beta \): the set whose elements in fact do not repress \( A \) but remain in \( \tilde{S}_\beta \);
\( \alpha, \beta, S_\alpha, S_\beta, E_\alpha, E_\beta, I_\alpha, I_\beta, D_\alpha, D_\beta \): the sizes of \( \tilde{\alpha}, \tilde{\beta}, \tilde{S}_\alpha, \tilde{S}_\beta, \tilde{E}_\alpha, \tilde{E}_\beta, \tilde{I}_\alpha, \tilde{I}_\beta, \tilde{D}_\alpha, \tilde{D}_\beta \), respectively, where the symbols with tildes represent the names of sets, while the symbols without tildes represent the sizes of corresponding sets.
Figure 2. Partition of the whole GRN in the algorithm corresponding to the inference of regulations upon a certain gene. $A$ represents the aim gene. $\tilde{S}_\alpha (\tilde{S}_\beta)$: whose elements are suspected to activate (repress) $A$. $\tilde{E}_\alpha (\tilde{E}_\beta)$: whose elements can be inferred definitely not to activate (repress) $A$. $\tilde{I}_\alpha (\tilde{I}_\beta)$: whose elements are correctly inferred as activators (repressors). $\tilde{D}_\alpha (\tilde{D}_\beta)$: whose elements in fact do not activate (repress) $A$ but remain in $\tilde{S}_\alpha (\tilde{S}_\beta)$ because of lack of data. $\tilde{\alpha} (\tilde{\beta})$: the set of activators (repressors). Obviously, $\tilde{I}_\alpha$ is a subset of the true active $\tilde{\alpha}$, and $\tilde{I}_\beta$ is a subset of $\tilde{\beta}$. Note that $\tilde{\alpha} \not\subseteq \tilde{S}_\alpha$, because some nodes in $\tilde{\alpha}$ (the shadow domain) may be suspected to repress $A$ so that they belong to $\tilde{D}_\beta$. $\{i\}_\text{shadow}$ is very small, the meaning of which is given in the text.

In detail, the partition is implemented as follows. For a certain gene $A$, all other genes can be classified into three sets: some genes activate it, some others repress it, and the remaining genes do not regulate it. Assume that set $\tilde{\alpha} (\tilde{\beta})$ represents the set whose elements activate (repress) $A$. Given a certain number of data, we can obtain some information about the links. Some nodes can be inferred to definitely not activate (repress) $A$, which are denoted by $\tilde{E}_\alpha (\tilde{E}_\beta)$. Some nodes are suspected to activate (repress) $A$, which are denoted by $\tilde{S}_\alpha (\tilde{S}_\beta)$. Among the set $\tilde{S}_\alpha (\tilde{S}_\beta)$, we can definitely identify a subset denoted by $\tilde{I}_\alpha (\tilde{I}_\beta)$ whose elements activate (repress) $A$. Obviously we have $\tilde{I}_\alpha \subseteq \tilde{\alpha}$ and $\tilde{I}_\beta \subseteq \tilde{\beta}$. Here the links via which the nodes in $\tilde{I}_\alpha (\tilde{I}_\beta)$ regulate $A$ are necessary in GRN and just called rigid edges in [31]. Moreover, there is a subset $\tilde{D}_\alpha (\tilde{D}_\beta)$ whose elements, in fact, do not activate (repress) $A$ remaining in $\tilde{S}_\alpha (\tilde{S}_\beta)$ because of insufficiency of the data. Strictly speaking, there may still be some intersections between $\tilde{D}_\beta$ and $\tilde{\alpha}$ represented by the shadow in figure 2 (note: no intersection between $\tilde{D}_\alpha$ and $\tilde{\beta}$). Nevertheless, this intersection $\{i\}_\text{shadow} = \tilde{D}_\beta \cap \tilde{\alpha} = \tilde{S}_\beta \cap \tilde{\alpha}$ is small, and it is extremely small for the sparse matrix. With increasing data we can reduce considerably $\tilde{D}_\beta$ as well as $\{i\}_\text{shadow}$, so that we treat $\{i\}_\text{shadow} = \emptyset$ for simplicity in the following analysis. Therefore we have

$$\tilde{S}_\beta = \tilde{\beta} \cup \tilde{D}_\beta, \quad \tilde{S}_\alpha \simeq \tilde{\alpha} \cup \tilde{D}_\alpha$$

(2)

and

$$\{i\}_\text{GRN} = \{A\} \cup \tilde{S}_\beta \cup \tilde{E}_\beta, \quad \tilde{E}_\beta = \tilde{S}_\alpha \cup \tilde{E}_\alpha.$$  

(3)

The last equality in (3) will be better explained in the next section.
Figure 3. Graphs (a)–(d) correspond to the four steps of logic classifications (i)–(iv). (a) ER. The whole rectangle represents all of the nodes except the aim-node A in the network. Before inference all the nodes are suspected to repress A, so all belong to the suspected-repress set \( \tilde{S}_\beta \). Using data of A’s transitions 1 \( \rightarrow \) 1 and 0 \( \rightarrow \) 1 some nodes can be excluded from the set \( \tilde{S}_\beta \) so they belong to the excluded-repress set \( \tilde{E}_\beta \). (b) EA. Using data of A’s transition 0 \( \rightarrow \) 0 some of the nodes in \( \tilde{E}_\beta \) that do not activate A can be excluded. The set \( \tilde{E}_\beta \) is divided into two parts: the suspected-act \( \tilde{S}_\alpha \) and the excluded-act \( \tilde{E}_\alpha \). (c) AR. Using the data of A’s transition 1 \( \rightarrow \) 0 some of the repressive links in the down-triangle can be identified from \( \tilde{S}_\beta \). These nodes are denoted by \( \tilde{I}_\beta \). The square \( \tilde{D}_\beta \) represents these nodes in \( \tilde{S}_\beta \) which in fact do not repress A. They are the nodes that step (i) fails to exclude from \( \tilde{S}_\beta \). (d) AA. Using the data of A’s transition 0 \( \rightarrow \) 1 some nodes in \( \tilde{S}_\alpha \) can be identified to activate A. They are denoted as the up-triangle \( \tilde{I}_\alpha \). The square \( \tilde{D}_\alpha \) represents the nodes that step (ii) fails to exclude from \( \tilde{S}_\alpha \).

3. Algorithm for inference

Based on the partition, an algorithm is proposed for identifying the partitioned sets step by step. Firstly, we can reduce the whole network into a subset of nodes that are suspected to regulate the aim gene, and then identify those links from the reduced suspected set. The first process is named Exclusion and the second one Affirmation. Since there are two kinds of regulations, the entire algorithm consists of four steps as shown in figure 3. Steps (i) and (ii) (figures 3(a) and (b)) are used to reduce the size of the suspected repressive set \( \tilde{S}_\beta \) and active set \( \tilde{S}_\alpha \), respectively, while steps (iii) and (iv) (figures 3(c) and (d)) are used to identify some true repressive \( \tilde{I}_\beta \) and active ones \( \tilde{I}_\alpha \) among these suspected sets. A novel trick in this work is to order these logic derivations in correct sequence so that the derivations of latter steps are based on the results of the previous ones. These steps require different kinds of data, which are summarized in

New Journal of Physics 13 (2011) 083002 (http://www.njp.org/)
Table 1. The data used in the algorithm.

| Step | A’s transition | Conditions for the given step |
|------|----------------|------------------------------|
| ER   | 1 → 1          | \(0 \ldots 0\)                |
|      | 0 → 1          | \(0 \ldots 0\) + \(\ldots 1\ldots\) |
|      |                | \(\tilde{\beta}\)            |
|      |                | \(\tilde{\alpha}\)           |
| EA   | 0 → 0          | \(0 \ldots 0\)                |
|      |                | \(\tilde{S}_\beta\)          |
|      |                | \(\tilde{\alpha}\)           |
| AR   | 1 → 0          | \(0 \ldots 0, 1, 0 \ldots 0\) |
|      |                | \(0 \ldots 0\)                |
|      |                | \(\tilde{\beta}\)            |
|      |                | \(\tilde{D}_\beta\)          |
| AA   | 0 → 1          | \(0 \ldots 0\)                |
|      |                | \(0 \ldots 0, 1, 0 \ldots 0\) |
|      |                | \(0 \ldots 0\)                |
|      |                | \(\tilde{S}_\beta\)          |
|      |                | \(\tilde{\alpha}\)           |
|      |                | \(\tilde{D}_\alpha\)         |

The types of input/output pairs used in the four steps. The first column represents the abbreviations of the steps; the second column denotes the transitions of the aim node A, i.e. \(S_A(t) \rightarrow S_A(t+1)\); the third one specifies the requirements to achieve the input patterns. These requirements include not only the necessary conditions to achieve A’s transition, but also the conditions under which the inference can be implemented in the corresponding step. For example, in step EA, the condition that all the nodes in \(\tilde{\alpha}\) stay in off-state can guarantee A’s transition \(0 \rightarrow 0\). But in order to exclude some nodes from the suspected-act set \(\tilde{S}_\alpha\), a condition that all nodes in \(\tilde{S}_\beta\) stay in off-state in input pattern is additionally required.

Table 1. After these four steps, some nodes regulating the aim gene A can be identified. Then by following the same procedures, these input/output data can be used again to infer the regulations exerted upon other aim nodes. The details of the algorithm can be described as follows:

(i) **Exclusion of repression (ER).** This step is ordered as the first derivation, and all other steps are based on the results of this logic computation. Before inference, all nodes except the aim gene A are suspected to repress A, so they all belong to the suspected-repress set \(\tilde{S}_\beta\). From A’s transitions \(0 \rightarrow 1\) and \(1 \rightarrow 1\) listed in table 1, we can know that the nodes with on-state in the input pattern do not repress A. They are excluded from \(\tilde{S}_\beta\) and classified into a new set, the excluded-repress set \(\tilde{E}_\beta\). The function of this step is to partition all the nodes except A into two parts, \(\tilde{S}_\beta\) and \(\tilde{E}_\beta\), as shown in figure 3(a).

(ii) **Exclusion of activation (EA).** The logic computation of this step depends on the result of (i), and the results of (i) and (ii) serve as the starting bases of steps (iii) and (iv), respectively. Before this step, all nodes in \(\tilde{E}_\beta\) are suspected to activate A. With the help of A’s transition \(0 \rightarrow 0\), if all the nodes in \(\tilde{S}_\beta\) stay in off-state as in table 1, we can exclude the nodes with on-state in the input pattern from \(\tilde{S}_\alpha\) and put them in the excluded-act set \(\tilde{E}_\alpha\). So this step partitions \(\tilde{E}_\beta\) into two sets: suspected-act set \(\tilde{S}_\alpha\) and excluded-act set \(\tilde{E}_\alpha\), as shown in figure 3(b).

(iii) **Affirmation of repression (AR).** This logic computation can be conducted based on the result of step (i). If only one node \(i\) in \(\tilde{S}_\beta\) stays at on-state in the input pattern, and
A switches from $S_A(t) = 1 \rightarrow S_A(t + 1) = 0$ (table 1), it can be inferred that $i$ represses $A$. In this step, we use this kind of data to identify the repressive links. If there are more than one node in $\tilde{S}_\beta$ staying in on-state in the input pattern, we cannot identify exactly which one represses $A$. So the requirements in table 1 include not only these conditions that guarantee that $A$ can switch from 1 to 0, but also the conditions to guarantee that we can identify the precise node repressing $A$. In this logic computation, the states of all nodes in $\tilde{E}_\beta$ set can be arbitrary, and this desirable condition can considerably enhance the efficiency of AR. Note that the information on $\tilde{S}_\beta$ and $\tilde{E}_\beta$ is prepared in step (i). The smaller the size of $\tilde{S}_\beta$, the easier it is to identify the repression. In figure 3(c), the set $I_\beta$ that has been identified to repress $A$ is denoted by the down-triangle. Please note that there are some other situations where some repressive regulations can be affirmed. But these situations are so rare that we ignore them reasonably in this paper.

(iv) Affirmation of activation (AA). This step is similar to AR by considering different sets of data. The logic derivation is based on the results of step (ii). If only one node $i$ in $\tilde{S}_\alpha$ has $S_i(t) = 1$ and all other nodes in $\tilde{S}_\alpha$ have $S(t) = 0$, the $A$’s transition $0 \rightarrow 1$ can help us to affirm that $i$ activates $A$. After this step, the set $I_\alpha$ is determined and is denoted by the up-triangle in figure 3(d).

The time expense of the algorithm can be estimated as follows. Given $N$ nodes and $q$ pairs of input/output data, we are collecting $Nq$ bits of information. For each node we need to examine these data not more than four times since there are four steps in the algorithm. Therefore, to infer the regulations upon $N$ nodes, the computing expense is proportional to $qN^2$.

4. Analytical derivations of inference precision

In the previous section, we showed how to use different logics to infer a given GRN. Here we go further to compute the probabilities of these logics and then derive explicit formulae quantitatively measuring the quality of the inference. In this section, we have two major assumptions. Firstly, we assume that the data about the input expression patterns are generated randomly, i.e. in each input pattern the probability that one node takes on-state or off-state is equal to 50%. Secondly, strong-inhibition dynamics, i.e. we assume the regulations obey the assumptions. Firstly, we assume that the data about the input expression patterns are generated randomly, i.e. in each input pattern the probability that one node takes on-state or off-state is equal to 50%. Secondly, strong-inhibition dynamics, i.e. we assume the regulations obey the simply and wildly observed rule. With these two assumptions we discuss the extent to which we can correctly reconstruct the regulation network given a certain number of data.

This problem can be formulated as follows: for an aim gene $A$ of a GRN that complies with the SIB mechanism of (1), we ask how many rigid edges on average can be identified, given $q$ pairs of input/output patterns with the input randomly chosen. In other words, what are the sizes of sets such as $\tilde{I}_\alpha$ and $\tilde{I}_\beta$?

(i) ER. This step will compute the probability $P(D_\beta)$ as well as $\langle D_\beta \rangle$ and $\langle S_\beta \rangle = \langle D_\beta \rangle + \beta$.

In order to explore the set $\tilde{E}_\beta$, we need $A$’s transition $1 \rightarrow 1$ or $0 \rightarrow 1$. The first kind of transition requires that $S_A(t) = 1$ and $S_i(t) = 0$, $\forall i \in \tilde{\beta}$, and the second kind requires that $S_A(t) = 0$, $S_i(t) = 0$, $\forall i \in \beta$ and $\exists j \in \tilde{\alpha}$, $S_j(t) = 1$. The probability that one pair of input/output data belongs to these two kinds can be exactly computed by $p_1$.

$$p_1 = 2^{-(\beta+1)} + 2^{-(\beta+1)}(1 - 2^{-\alpha}),$$

where the first term corresponds to the first kind of maps $S_A(t) = 1 \rightarrow S_A(t + 1) = 1$, while the second term considers maps with $S_A(t) = 0 \rightarrow S_A(t + 1) = 1$. With $q$ pairs of random data,
the probability \( p(k_\beta) \) that \( k_\beta \) pairs satisfy condition (4) can be computed analytically by the binomial distribution \( B(q, k_\beta, p_1) \) as

\[
p(k_\beta) = B(q, k_\beta, p_1) = \binom{q}{k_\beta} p_1^{k_\beta} (1 - p_1)^{q-k_\beta}.
\]  

(5)

After this step, node \( i \not\in \tilde{\beta} \) can remain in \( \tilde{D}_\beta \subseteq \tilde{S}_\beta \) if its value is 0 in all of these \( k_\beta \) input patterns. Here, the probability \( p(D_\beta|k_\beta) \) that in \( \tilde{S}_\beta \) there are \( D_\beta \) nodes not repressing \( A \) can be calculated as

\[
p(D_\beta|k_\beta) = B(N - \beta - 1, D_\beta, 2^{-k_\beta}).
\]  

(6)

Hence, the total probability of \( D_\beta \) is

\[
p(D_\beta) = \sum_{k_\beta=0}^{q} p(D_\beta|k_\beta) p(k_\beta),
\]  

(7)

which yields the expectation of \( D_\beta \)

\[
\langle D_\beta \rangle = \sum_{D_\beta=0}^{N-\beta-1} D_\beta p(D_\beta) = (N - \beta - 1) \left(1 - \frac{p_1}{2}\right)^q \approx (N - \beta - 1) e^{-(1/2)q p_1},
\]

(8)

and

\[
\langle S_\beta \rangle = \langle D_\beta \rangle + \beta.
\]  

(9)

(ii) EA. This step will derive \( P(D_a) \) and \( \langle D_a \rangle \) based on the results of \( D_\beta \).

Node \( A \)’s transition \( 0 \rightarrow 0 \) is required in this step. If \( S_A(t) = 0 \) and \( S_i(t) = 0 \), \( \forall i \in \tilde{\alpha} \), we can have the transition of \( S_A(t) = 0 \rightarrow 0 \). But in order to exclude the influences of nodes from \( \tilde{S}_\beta \), we need additionally \( S_j(t) = 0 \), \( \forall j \in \tilde{S}_\beta \), so that we can infer node \( l \in \tilde{E}_a \) if \( S_i(t) = 1 \). As a result, the requirement of step (ii) is that \( S_A(t) = 0 \) and \( S_i(t) = 0 \), \( \forall i \in \tilde{S}_\beta \cup \tilde{\alpha} \) (table 1). We call the input/output data that satisfy the requirement utilizable pairs of data for the EA step. Based on the two assumptions at the beginning of this section, the probability for a pair of data belonging to this utilizable set is \( p_2 \):

\[
p_2 = 2^{-(\alpha+\beta+D_\beta+1)}.
\]  

(10)

According to the binomial distribution, given the size of \( \tilde{D}_\beta \), the probability that there are \( k_\alpha \) pairs of utilizable data is denoted by

\[
p(k_\alpha|D_\beta) = B(q, k_\alpha, p_2) = \binom{q}{k_\alpha} p_2^{k_\alpha} (1 - p_2)^{q-k_\alpha}.
\]  

(11)

Similar to (6), the probability of the size of \( \tilde{D}_a \) can be computed as

\[
p(D_a|k_\alpha, D_\beta) = B(N - \alpha - \beta - D_\beta - 1, D_a, 2^{-k_\alpha}).
\]  

(12)

Then the total probability \( p(D_a|D_\beta) \) that, in the set \( \tilde{S}_a \), there are \( D_a \) nodes that do not activate node \( A \) can be summed by

\[
p(D_a|D_\beta) = \sum_{k_\alpha=0}^{q} p(D_a|k_\alpha, D_\beta) p(k_\alpha|D_\beta),
\]  

(13)
leading to the expectation of $D_\alpha$

$$\langle D_\alpha \rangle_{D_\beta} = (N - \alpha - \beta - D_\beta - 1) \left(1 - \frac{p_2}{2}\right)^q \approx (N - \alpha - \beta - D_\beta - 1)e^{-(1/2)qp_2}. \quad (14)$$

so

$$\langle S_\alpha \rangle_{D_\beta} = \alpha + \langle D_\alpha \rangle_{D_\beta}. \quad (15)$$

(iii) AR. Based on $D_\alpha$ and $D_\beta$, the probability $P(I_\beta)$ and the expectation $\langle I_\beta \rangle$ can be computed.

If in an input pattern, $S_A(t) = 1, \exists j \in \tilde{\beta}$, $S_j(t) = 1$ and $S_i(t) = 0, \forall i \in (\tilde{\beta} \cup \tilde{D}_\beta - \{j\})$, then node $A$ will transit from $1 \rightarrow 0$ and node $j$ can be identified to repress $A$. In other words, in total $(\beta + D_\beta + 1)$ nodes in the pattern should stay at required states, so the probability for an input pattern to satisfy the requirement is $2^{-(\beta + D_\beta + 1)}$. Then, given $q$ pairs of data, the probability that node $j \in \tilde{\beta}$ can be affirmed is

$$p_3 = 1 - (1 - 2^{-(\beta + D_\beta + 1)})^q. \quad (16)$$

Treating the affirmation of each node in $\tilde{\beta}$ as independent, the probability of the size of the total affirmed repressive set $\tilde{I}_\beta$ reads

$$p(I_\beta|D_\beta) = B(\beta, I_\beta, p_3). \quad (17)$$

Therefore, given $D_\beta$, the expectation of $I_\beta$ reads

$$\langle I_\beta \rangle_{D_\beta} = \sum_{I_\beta=0}^{\beta} I_\beta p(I_\beta|D_\beta) = \sum_{I_\beta=0}^{\beta} I_\beta \left(\frac{\beta}{I_\beta}\right) p_3 I_\beta (1 - p_3)^{\beta - I_\beta} = \beta p_3. \quad (18)$$

(iv) AA. Finally, the formulae of $P(I_\alpha)$ and $\langle I_\alpha \rangle$ can be derived after those of $D_\alpha$ and $D_\beta$.

If $S_A(t) = 0, \exists j \in \tilde{\alpha}$, $S_j(t) = 1$ and $S_i(t) = 0, \forall i \in (\tilde{\beta} \cup \tilde{D}_\beta \cup \tilde{\alpha} \cup \tilde{D}_\alpha - \{j\})$, and if further we observe $S_A(t + 1) = 1$, the link that $j$ activates $A (j \in \tilde{\alpha})$ can be affirmed. The probability that an input pattern can be used to identify $j \in \tilde{\alpha}$ is $2^{-(\alpha + \beta + D_\alpha + D_\beta + 1)}$. And given $q$ pairs of data, the probability that $\forall j \in \tilde{\alpha}$ can be affirmed reads

$$p_4 = 1 - (1 - 2^{-(\alpha + \beta + D_\alpha + D_\beta + 1)})^q. \quad (19)$$

Given the sizes of $\tilde{D}_\alpha$ and $\tilde{D}_\beta$ from steps (i) and (ii), the probability that $I_\alpha$ active links are affirmed in this step can be computed as

$$p(I_\alpha|D_\alpha, D_\beta) = B(\alpha, I_\alpha, p_4) \quad (20)$$

and

$$\langle I_\alpha \rangle_{D_\alpha, D_\beta} = \sum_{I_\alpha=0}^{\alpha} I_\alpha p(I_\alpha|D_\alpha, D_\beta) = \alpha p_4. \quad (21)$$

Finally, given $q$ pairs of data, the expectation of the rigid edges we can identify can be calculated as

$$\langle I_\alpha \rangle = \sum_{D_\beta=0}^{N-\alpha-\beta-1} \sum_{D_\alpha=0}^{N-\alpha-\beta-D_\beta-1} \langle I_\alpha \rangle_{D_\alpha, D_\beta} p(D_\alpha|D_\beta)p(D_\beta). \quad (22)$$
\[ (I_\beta) = \sum_{D_\beta=0}^{N-\beta-1} \langle I_\beta \rangle |_{D_\beta} p(D_\beta) \]  

(23)

and

\[ (I_\alpha + I_\beta) = (I_\alpha) + (I_\beta), \]  

(24)

where \( p(D_\beta), p(D_\alpha|D_\beta), \langle I_\beta \rangle |_{D_\beta} \) and \( \langle I_\alpha \rangle |_{D_\alpha, D_\beta} \) are given in equations (7), (13), (18) and (21), respectively. The probability \( p_\alpha (p_\beta) \) that an active link (repressive link) regulating gene \( A \) can be identified is

\[ p_\alpha = \frac{\langle I_\alpha \rangle}{\alpha}, \quad p_\beta = \frac{\langle I_\beta \rangle}{\beta}. \]  

(25)

Note that the quantities \( \langle I_\alpha \rangle, \langle I_\beta \rangle \) and \( \langle I_\alpha + I_\beta \rangle \) as well as \( p_i, i = 1, 2, 3, 4, \) all depend on \( \alpha \) and \( \beta \) of node \( A \); therefore, we should denote them as \( \langle I_\alpha \rangle |_{\alpha, \beta}, \langle I_\beta \rangle |_{\alpha, \beta} \) and \( \langle I_\alpha + I_\beta \rangle |_{\alpha, \beta} \) and so on.

In the next section, we will compare the expectation calculated by (22), (23) and (24) with the average inference results of direct numerical simulations. We use the method proposed in [38] to calculate all binomial distributions.

5. Numerical results

In this section, we compare our analytic results with numerical simulations of the algorithm in section 3 to check the validity of (24). For this purpose, we consider an example of a GRN with \( N = 128 \) and random links between nodes. Random input patterns are assigned to the network, based on which output patterns are generated according to the SIB mechanism (1). Given a group of data that consists of \( q \) pairs of input/output patterns, the logic algorithm can numerically identify a part of the links. For a node that is actually activated by \( \alpha \) nodes and repressed by \( \beta \) nodes, the number of identified active links and repressive links can be denoted by \( I_\alpha |_{\alpha, \beta} \) and \( I_\beta |_{\alpha, \beta} \), respectively. For different sets of data, different inference results \( I_\alpha |_{\alpha, \beta} \) and \( I_\beta |_{\alpha, \beta} \) can be obtained.

In figure 4, circles represent numerical results \( \langle I_\alpha + I_\beta \rangle |_{\alpha, \beta} \) averaged from 100 groups of different random data. The value of \( q \) ranges from 64 to 2048. We compare in figure 4 the numerical results of \( (\alpha, \beta) = (5, 0), (2, 1), (4, 2), (5, 3) \) and \( (2, 7) \) with those of the analytic results of (24), and observe that the theoretical predictions coincide with the numerical data very well.

The above comparison demonstrates that with analytical results \( \langle I_\alpha + I_\beta \rangle_\text{tot} = \sum_{i=1}^{N} (I_\alpha_i + I_\beta_i) \), we are able to answer a question of practical importance: how many data are required when we want to identify a required fraction of links of a GRN. The above calculations are based on the information on detailed \( \alpha_i \) and \( \beta_i \) values. Unfortunately, these pieces of information are often not available in practical cases. Instead, some statistical features of GRNs, e.g. the distribution of \( \alpha \) and \( \beta \), \( P(\alpha, \beta) \), can possibly be estimated; we can then derive the formula from this statistics

\[ \langle I_\alpha + I_\beta \rangle_\text{tot} = N \sum_{\alpha, \beta} P(\alpha, \beta) \langle I_\alpha + I_\beta \rangle |_{\alpha, \beta}. \]  

(26)
Figure 4. Comparison of the simulation results (circles) with the analytic results (solid curves). Circles represent the simulation results obtained by inferring randomly generated networks from randomly generated data, while the lines represent the prediction from (24). The numerical data are averaged from 100 random GRN samples.

For instance, let us consider the inference of a realistic biological network, where we use the cell-cycle network of the budding yeast [29] pruning all self-regulations as in figure 1 as an example. Assume that the distribution of links exerted on each node is quasi-Poisson as

\[ P(i) = e^{-(E-1)}(E-1)^{i-1}/(i-1)! \quad i = 1, 2, \ldots, \]  

(27)

which guarantees that each node is regulated by at least one link. Assume further that the numbers of active and repressive links exerted on each node are almost the same, i.e. \( \beta = \alpha \) (the degree is even) or \( \beta = \alpha + 1 \) (the degree is odd). Then the expectation of inference of the whole GRN can be calculated as

\[ \langle I_\alpha + I_\beta \rangle_{\text{tot}} = N \sum_i \left( P(2i) \langle I_\alpha + I_\beta \rangle_{|\alpha=i, \beta=i} + P(2i+1) \langle I_\alpha + I_\beta \rangle_{|\alpha=i, \beta=i+1} \right). \]  

(28)

Given the actual node number \( N = 11 \) and the average regulation \( E = 2.9 \), we calculate the expectation with (28) shown by the solid line in figure 5. The numerically averaged results of inference with \( q \) pairs of data are recorded as circles, and the standard deviations are shown by error bars. The figure demonstrates that the formulae we derived can predict well the inference process even when the interaction degrees of each node are not fully accessible.

Regarding the temporal data, each initial input pattern generates a sequence of data that consists of a number of input/output patterns until the system evolves to a stationary state. As a result, given \( q \) initial input patterns, more than \( q \) pairs of input/output data can be acquired, so that as shown by the dotted line in the inset of figure 5, many more regulations can be inferred with these data. However, in order to predict the inference with temporal data, the equations in this paper should be considerably modified. It may be related to the properties of attractors and trajectories, and further study is required.
Figure 5. Comparison of simulation results (circles) with analytic results (red line) of the inference of the cell-cycle network of the budding yeast without self-regulation. Error-bars indicate the fluctuations of numerical experiments. The simulation results are averaged over 100 groups of data consisting of $q$ pairs of random input patterns. In contrast, the dotted line demonstrates the numerical result of inference with temporal data. In the case of temporal data, each initial input pattern generates a sequence of data which consists of multiple pairs of input/output data. So given $q$ initial input patterns, the algorithm can identify more links in the GRN, especially at the early stage of inference. The curve is also averaged over 100 groups of data. The details when $q \leqslant 40$ are enlarged in the inset.

6. Conclusion

In this paper, we discussed the inference of GRNs with SIB logic dynamics, and obtained analytical formulae to calculate the probability that one link can be inferred from a certain number of data. The theoretical predictions are consistent with the numerical results.

The novelty of our work lies in two aspects. Firstly, in contrast with previous works on the inference of Boolean networks [25–27] in which the GRN is a ‘black box’ where we do not have any information about the regulation rules, we propose here an algorithm to infer GRNs treating them as ‘gray boxes’ as in [7], where we know a priori the rules of regulation. This algorithm can be used to efficiently infer GRNs in which the genes regulate each other via the SIB mechanism. Furthermore, its idea, i.e. to partition the network according to different kinds of regulations and to order properly the computation steps, is expected to apply to inference based on general regulation rules.
Secondly, we present analytical formulae for the number of links that can be inferred when the data are insufficient. In many cases, the data that can be acquired are not enough to realize the completely reconstruction of networks. Since the inference of GRNs is always underdetermined [8], researchers in practice are usually concerned with the problem of how many links can be inferred with insufficient information. However, to date most of the works on this topic are not analytical as in this paper, but just numerical. With these formulae presented above, we can estimate, for some classes of gene transcriptional regulatory networks, the experimental costs when we require a certain precision of inference to reveal the topology of GRNs [39]. This capability for estimation is important in many practical inference tasks.

Finally, the approach proposed in the paper needs to be extended in several aspects. Interesting directions for further research could include, for instance, treating the temporal data or uncertainties in the data. These topics will be explored in future work.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant no. 10975015 and the National Basic Research Program of China (973 Program; 2007CB814800).

References

[1] de Jong H 2002 Modeling and simulation of genetic regulatory systems: a literature review J. Comput. Biol. 9 67–103
[2] Karlebach G and Shamir R 2008 Modelling and analysis of gene regulatory networks Nat. Rev. Mol. Cell Biol. 9 770
[3] Costanzo M et al 2010 The genetic landscape of a cell Science 327 425–31
[4] Vejmelka M and Paluš M 2008 Inferring the directionality of coupling with conditional mutual information Phys. Rev. E 77 026214
[5] Kramer M A, Eden U T, Cash S S and Kolaczyk E D 2009 Network inference with confidence from multivariate time series Phys. Rev. E 79 061916
[6] Nawrath J, Carmen Romano M, Thiel M, Kiss I Z, Wickramasinghe M, Timmer J, Kurths J and Schelter B 2010 Distinguishing direct from indirect interactions in oscillatory networks with multiple time scales Phys. Rev. Lett. 104 038701
[7] Shandilya S G and Timme M 2011 Inferring network topology from complex dynamics New J. Phys. 13 013004
[8] Yeung M K, Tegnér J and Collins J J 2002 Reverse engineering gene networks using singular value decomposition and robust regression Proc. Natl Acad. Sci. USA 99 6163–8
[9] Kashtan N and Alon U 2005 Spontaneous evolution of modularity and network motifs Proc. Natl Acad. Sci. USA 102 13773–8
[10] Lau K, Ganguli S and Tang C 2007 Function constrains network architecture and dynamics: a case study on the yeast cell cycle Boolean network Phys. Rev. E 75 051907
[11] Faith J J et al 2007 Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles PLoS Biol. 5 e8
[12] Amit I et al 2009 Unbiased reconstruction of a mammalian transcriptional network mediating pathogen responses Science 326 257–63
[13] Carro M S et al 2010 The transcriptional network for mesenchymal transformation of brain tumours Nature 463 318–25
[14] De la Fuente A, Brazhnik P and Mendes P 2002 Linking the genes: inferring quantitative gene networks from microarray data Trends Genet. 18 395
Gardner T S, di Bernardo D, Lorenz D and Collins J J 2003 Inferring genetic networks and identifying compound mode of action via expression profiling Science 301 102

Tegnér J, Yeung M K, Hasty J and Collins J J 2003 Reverse engineering gene networks: integrating genetic perturbations with dynamical modeling Proc. Natl Acad. Sci. USA 100 5944–9

Friedman N 2004 Inferring cellular networks using probabilistic graphical models Science 303 799

di Bernardo D et al 2005 Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks Nat. Biotechnol. 23 377

Rice J J, Tu Y and Stolovitzky G 2005 Reconstructing biological networks using conditional correlation analysis Bioinformatics 21 765

Bonneau R et al 2006 The inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology datasets de novo Genome Biol. 7 R36

de la Fuente A and Makhecha D P 2006 Unravelling gene networks from noisy underdetermined experimental perturbation data IEE Proc. Syst. Biol. 153 257

Bansal M, Belcastro V, Ambesi-Impiombato A and di Bernardo D 2007 How to infer gene networks from expression profiles Mol. Syst. Biol. 3 78

Hecker M, Lambeck S, Toepfer S, van Someren E and Guthke R 2009 Gene regulatory network inference: data integration in dynamic models—a review Biosystems 96 86–103

Marbach D, Prill R J, Schaffter T, Mattiussi C, Floreano D and Stolovitzky G 2010 Revealing strengths and weaknesses of methods for gene network inference Proc. Natl Acad. Sci. USA 107 6286–91

Akutsu T, Miyano S and Kuhara S 1999 Identification of genetic networks from a small number of gene expression patterns under the boolean network model Pac. Symp. Biocomput. 4 17–28

Liang S, Fuhrman S and Somogyi R 1998 Reveal, a general reverse engineering algorithm for inference of genetic network architectures Pac. Symp. Biocomput. 3 18–29

Lähdesmäki H, Shmulevich I and Yli-Harja O 2003 On learning gene regulatory networks under the Boolean network model Mach. Learn. 52 147–67

Kauffman S A 1969 Metabolic stability and epigenesis in randomly constructed genetic nets J. Theor. Biol. 22 437–67

Li F, Long T, Lu Y, Ouyang Q and Tang C 2004 The yeast cell-cycle network is robustly designed Proc. Natl Acad. Sci. USA 101 4781–6

Wu Y, Zhang X, Yu J and Ouyang Q 2009 Identification of a topological characteristic responsible for the biological robustness of regulatory networks PLoS Comput. Biol. 5 e1000442

Wang G, Du C, Chen H, Simha R, Rong Y, Xiao Y and Zeng C 2010 Process-based network decomposition reveals backbone motif structure Proc. Natl Acad. Sci. USA 107 10478

Bornholdt S 2008 Boolean network models of cellular regulation: prospects and limitations J. R. Soc. Interface 5 S85–S94

Kauffman S, Peterson C, Samuelsson B J and Troein C 2003 Random Boolean network models and the yeast transcriptional network Proc. Natl Acad. Sci. USA 100 14796–9

Jarrah A S, Raposa B and Laubenbacher R 2007 Nested canalizing, unate cascade, and polynomial functions Physica D 233 167–74

Albert R and Othmer H G 2003 The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in drosophila melanogaster J. Theor. Biol. 223 1–18

Tan N and Ouyang Q 2006 Design of a network with state stability J. Theor. Biol. 240 592

Fortuna M A and Meli C J 2007 Do scale-free regulatory networks allow more expression than random ones? J. Theor. Biol. 247 331–6

Newman M E J and Ziff R M 2001 Fast Monte Carlo algorithm for site or bond percolation Phys. Rev. E 64 016706

Guimerà R and Sales-Pardo M 2009 Missing and spurious interactions and the reconstruction of complex networks Proc. Natl Acad. Sci. USA 106 22073–8