BIOINFORMATICS

Vol. 16 no. 8 2000
Pages 727–734

Inferring qualitative relations in genetic networks and metabolic pathways

Tatsuya Akutsu [1,*], Satoru Miyano [1] and Satoru Kuhara [2]

[1]Human Genome Center, Institute of Medical Science, University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan and [2]Graduate School of Genetic Resources Technology, Kyushu University, Hakozaki 6-10-1, Higashi-ku, Fukuoka 812-8581, Japan

Received on February 22, 1999; revised and accepted on April 18, 2000

Abstract

Motivation: *Inferring genetic network architecture from time series data of gene expression patterns is an important topic in bioinformatics. Although inference algorithms based on the Boolean network were proposed, the Boolean network was not sufficient as a model of a genetic network.*

Results: *First, a Boolean network model with noise is proposed, together with an inference algorithm for it. Next, a qualitative network model is proposed, in which regulation rules are represented as qualitative rules and embedded in the network structure. Algorithms are also presented for inferring qualitative relations from time series data. Then, an algorithm for inferring S-systems (synergistic and saturable systems) from time series data is presented, where S-systems are based on a particular kind of nonlinear differential equation and have been applied to the analysis of various biological systems. Theoretical results are shown for Boolean networks with noises and simple qualitative networks. Computational results are shown for Boolean networks with noises and S-systems, where real data are not used because the proposed models are still conceptual and the quantity and quality of currently available data are not enough for the application of the proposed methods.*

Contact: *takutsu@ims.u-tokyo.ac.jp*

Introduction

Recently many studies have been performed in order to develop computational methods for reconstructing underlying *genetic networks* from time series data of gene expression patterns, which are obtained by the DNA microarray technology (DeRisi *et al.*, 1997).

Several studies have been carried out using the Boolean network, where a gene takes one of two states (ON or OFF), and a gene regulation rule is given as a Boolean function. Liang *et al.* (1998) developed the algorithm REVEAL (reverse engineering algorithm) for inferring genetic networks from state transition tables, which correspond to time series data of gene expression patterns. We proved that $O(\log n)$ expression patterns are necessary and sufficient to identify the underlying Boolean network of n genes correctly with high probability if the maximum indegree is bounded (Akutsu *et al.*, 1999).

Since the Boolean network is not realistic, other models have been proposed. Thieffry and Thomas (1998) proposed a qualitative model, which was similar to our model. However, they did not give a concrete inference algorithm. Although other hybrid models were proposed (McAdams and Shapiro, 1995; Yuh *et al.*, 1998), inference methods were unclear. Arkin *et al.* (1997) proposed a statistical method to infer chemical networks. Chen *et al.* (1999) and D'haeseleer *et al.* (1999) proposed inference methods based on linear differential equations. However, no method seems to be sufficient.

In this paper, we propose a *qualitative network* model, which is different from the model proposed by Thieffry and Thomas (1998). This new model can be considered as an intermediate model between the Boolean network model and the differential equation model. This model can also be considered as a combination of the Boolean network and qualitative reasoning (de Kleer and Brown, 1984). In this model, regulation rules are represented as qualitative rules and embedded in network structures. We also present inference algorithms for this model. Although the algorithms are based on linear differential equations, they can be applied to nonlinear models to some extent. One of the algorithms can be applied to the inference of S-systems (Irvine and Savageau, 1990; Savageau, 1991; Tominaga and Okamoto, 1998), where S-systems are based on a particular kind of nonlinear differential equation and have been successfully applied to the analysis of various biological networks.

*To whom correspondence should be addressed.

Incidentally, it is also important to develop inference algorithms robust for noises. Thus, we propose a robust algorithm for a Boolean network model with noises, where the technique can also be applied to qualitative networks.

The organization of the paper is as follows. First we present a robust algorithm for Boolean networks with noises. Next we present a qualitative network model and inference algorithms. Then we show computational results using artificial data, and finally we conclude with future work.

Identification of Boolean networks with noises

Boolean network and its identification

Here we briefly review the Boolean network model and our previous result on its identification (Akutsu *et al.*, 1999).

A *Boolean network* $G(V, F)$ consists of a set $V = \{v_1, \ldots, v_n\}$ of nodes representing genes and a list $F = (f_1, \ldots, f_n)$ of *Boolean functions*, where a Boolean function $f_i(v_{i_1}, \ldots, v_{i_k})$ with inputs from specified nodes $v_{i_1}, \ldots, v_{i_k}$ is assigned to each node v_i. An *expression pattern* ψ is a function from V to $\{0, 1\}$. That is, ψ represents the states of nodes (genes), where each node is assumed to take either 0 (not-express) or 1 (express) as its state value. Expression pattern ψ_{t+1} at time $t + 1$ is determined by Boolean functions F from expression pattern ψ_t at time t (i.e. $\psi_{t+1}(v_i) = f_i(\psi_t(v_{i_1}), \ldots, \psi_t(v_{i_k}))$).

In the identification, we are given a set of INPUT/OUTPUT pairs $\{(I_1, O_1), \ldots,(I_m, O_m)\}$, where each I_j corresponds to ψ_t at some time t and each O_j corresponds to ψ_{t+1}. We assume that OUTPUT patterns are generated from corresponding INPUT patterns according to Boolean functions in the underlying Boolean network. The identification problem is, given n and $\{(I_1, O_1), \ldots, (I_m, O_m)\}$, to find the original (underlying) Boolean network.

We say that a Boolean network is *consistent* with INPUT/OUTPUT patterns if $O_j(v_i) = f_i(I_j(v_{i_1}), \ldots, I_j(v_{i_k}))$ holds for all v_i and for all (I_j, O_j). We say that the Boolean network is *identified* if an identification algorithm finds that there is only one consistent Boolean network.

In most of this paper, we assume that the *indegree* (i.e. the number of input nodes) of each node is bounded by a constant K, because it has been proved that exponentially many patterns are required if K is not bounded (Akutsu *et al.*, 1999). The importance of the constraint on the indegree is also pointed out in several papers (Liang *et al.*, 1998; Chen *et al.*, 1999). Although we assume that the maximum indegree is bounded by K, all algorithms in this paper can be applied to Boolean (or qualitative) networks whose maximum indegree is not bounded: the algorithms correctly identify Boolean (or qualitative) functions assigned to all nodes whose indegrees are at most K.

In our previous work, we developed an identification algorithm (denoted by BOOL-1) for Boolean networks. BOOL-1 is quite simple: it examines for each node independently whether there exists a unique Boolean function consistent with given patterns. Moreover, we proved the following theorem, where $\log(x) \equiv \log_2(x)$ in this paper.

THEOREM 1 (AKUTSU *et al.*, 1999). *If* $O(2^{2K} \cdot (2K + \alpha) \cdot \log n)$ *INPUT patterns are given uniformly randomly, BOOL-1 correctly identifies the underlying Boolean network of maximum indegree* $\leq K$ *with probability at least* $1 - \frac{1}{n^\alpha}$, *where* $\alpha > 1$ *is any fixed constant.*

Noisy Boolean network and its identification

Since real expression patterns may contain noises, we define a *noisy Boolean network*. Let $G(V, F)$ be a Boolean network. Then, a noisy Boolean network consists of $G(V, F)$ and p_{noise}, where p_{noise} is a constant such that $0 \leq p_{\text{noise}} < 1$. There is only one difference between the standard Boolean network and the noisy Boolean network: $O_j(v_i) = f_i(I_j(v_{i_1}), \ldots, I_j(v_{i_k}))$ holds for each node in a standard Boolean network, whereas $O_j(v_i) \neq f_i(I_j(v_{i_1}), \ldots, I_j(v_{i_k}))$ holds with probability $\leq p_{\text{noise}}$ for each node in a noisy Boolean network, where the probability is taken over all possible INPUT patterns I_j.

The identification algorithm (denoted by BOOL-2) for noisy Boolean networks is obtained by slightly modifying BOOL-1. In BOOL-1 each Boolean function inconsistent with at least one INPUT/OUTPUT pattern is discarded, but in BOOL-2 each Boolean function inconsistent with at least $\theta \cdot m$ patterns is discarded. In this paper, we use $\theta = \frac{1}{2^{2K+1}}$ for theoretical analysis, where other appropriate values can be used in practice. The following is a PASCAL-like code of BOOL-2.

```
for i = 1 to n do
  count ← 0;
  for all combinations of K nodes (v_{i_1}, ..., v_{i_K}) do
    for all Boolean function f with K inputs do
      mismatch ← 0;
      for j = 1 to m do
        if O_j(v_i) ≠ f_i(I_j(v_{i_1}), ..., I_j(v_{i_K})) then
          mismatch ← mismatch + 1;
      if mismatch < θ · m then
        output f(v_{i_1}, ..., v_{i_K}) as a function
        assigned to v_i; count ← count + 1;
  if count ≠ 1 then
    output "NOT IDENTIFIED" and halt;
```

It is easy to see that BOOL-2 works in $O(n^{K+1}m)$ time, which is the same order as in BOOL-1. On the number of expression patterns, we can prove the following theorem (see the Appendix for the proof).

THEOREM 2. *Assume that* $p < \frac{1}{e \cdot 2^{2K+2}}$. *If* $O\left(2^{2K} \cdot (\alpha + K + 1) \cdot \left(1 + \frac{1}{\log \frac{1}{p} - \log e - (2K+2)}\right) \cdot \log n\right)$ *INPUT patterns*

are given uniformly randomly, BOOL-2 correctly identifies the underlying Boolean network with maximum indegree K with probability at least $1 - \frac{1}{n^\alpha}$, where $\alpha > 1$ is any fixed constant.

Although the assumption on p is too strong in the above, it seems that a similar property will hold for much larger p (see Computational results).

Inference of qualitative networks

Qualitative network model

Qualitative reasoning has been studied in Artificial Intelligence (de Kleer and Brown, 1984). Theories of qualitative reasoning were developed for predicting and explaining the behavior of physical mechanisms in qualitative terms. In qualitative reasoning, instead of real-valued variables, each variable is described quantitatively: taking on only a small number of values, usually $+$, $-$, or 0. Instead of differential equations, qualitative equations are also used.

Based on the concept of qualitative reasoning, we define a qualitative network model. A *qualitative network* is a directed graph $G(V, E)$, where each node in $V = \{v_1, \ldots, v_n\}$ corresponds to a gene or a chemical substance, and each directed edge $(v_j, v_i) \in E$ has a label: either *activation* or *inhibition*. In this paper, $v_j \to v_i$ denotes an activation edge (from v_j to v_i) and $v_j \dashv v_i$ denotes an inhibition edge (from v_j to v_i). The meanings of activation and inhibition depend on kinds of differential equation used in kinetic models representing genetic networks and/or metabolic pathways. Qualitative networks based on linear differential equations and inference algorithms for these networks are presented in this section. An inference algorithm for qualitative understanding of S-systems is presented in the next section.

It should be noted that we intend to use qualitative networks not for simulation, but to represent biological knowledge. Thus, we do not need to know precise values of parameters but we need to know topologies of networks. Exact fitting of parameters does not seem to be realistic because it is very difficult to make precise quantitative models of complex biological systems.

Simple qualitative networks

For ease of explanation, we begin with a simplest model, to be extended to more realistic models later. Let $X_i(t)$ be the value (expression level of a gene or concentration of a chemical substance) of v_i at time t, where we sometimes omit '(t)'. We assume that time series data of a biological system are produced according to the following simple system of linear differential equations:

$$\frac{dX_1}{dt} = a_1 X_{j_1}, \frac{dX_2}{dt} = a_2 X_{j_2}, \ldots, \frac{dX_n}{dt} = a_n X_{j_n}.$$

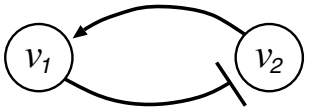

Fig. 1. Qualitative network corresponding to $\left\{\frac{dX_1}{dt} = X_2, \frac{dX_2}{dt} = -X_1\right\}$.

Then, the qualitative network corresponding to this linear system is defined by $V = \{v_1, \ldots, v_n\}$ and $E = \{v_{j_i} \to v_i \mid a_i > 0\} \cup \{v_{j_i} \dashv v_i \mid a_i < 0\}$.

For example, consider a case of $n = 2$, $j_1 = 2$, $j_2 = 1$, $a_1 = 1$ and $a_2 = -1$ (i.e. $X_1(t) = \sin(t + \theta)$ and $X_2(t) = \cos(t + \theta)$ where θ is determined from the initial values). Then, $E = \{v_2 \to v_1, v_1 \dashv v_2\}$ (see Figure 1).

The task of an inference algorithm is, given n and $X_i(t)$, to infer a qualitative network $G(V, E)$ consistent with $X_i(t)$. The inference algorithm (denoted by QNET-1) is given below. QNET-1 is similar to BOOL-1 and BOOL-2. It examines all possible edges and discards edges inconsistent with given data. Note that we assume that values of $X_i(t)$ are given for $t = t_1, t_1 + \Delta, t_1 + 2\Delta, t_1 + 3\Delta, \ldots, t_1 + m\Delta$. Note also that we approximate $\frac{dX_i(t)}{dt}$ by $\frac{\Delta X_i(t)}{\Delta}$, where $\Delta X_i(t)$ denotes $X_i(t+\Delta) - X_i(t)$.

$E \leftarrow \{v_j \to v_i,\ v_j \dashv v_i \mid i = 1 \ldots n,\ j = 1 \ldots n\}$;
for $i = 1$ to n **do**
  for $j = 1$ to n **do**
    for $t = t_1$ to $t_1 + (m-1)\Delta$ **do**
      if $\Delta X_i(t) \cdot X_j(t) < 0$ **then** delete $v_j \to v_i$ from E;
      if $\Delta X_i(t) \cdot X_j(t) > 0$ **then** delete $v_j \dashv v_i$ from E;
  if $indegree(v_i) > 1$ **then**
    output "NOT IDENTIFIED" and **halt**;

In practice, '> 0' and '< 0' in the above should be replaced by '$> \rho$' and '$< -\rho$' using an appropriate threshold value ρ.

It is easy to see that this algorithm works in $O(n^2 m)$ time. Here, we briefly discuss input time series data. It is easy to see that correct edges are not deleted under the assumption that $\mathrm{sign}\left(\frac{\Delta X_i(t)}{\Delta}\right) = \mathrm{sign}\left(\frac{dX_i(t)}{dt}\right)$. However, wrong edges may remain if sufficient data are not given. In most cases, time series data beginning from only one set of initial values (i.e. $f(t_1)$) are not sufficient because time series data may fall into *attractors*. In such a case, time series data beginning from other sets of initial values are required. The importance of using time series data beginning from multiple sets of initial values is discussed by Akutsu *et al.* (1999). The following theorem holds regardless of the existence or sizes of attractors.

THEOREM 3. *Assume that initial values are chosen from* $\{1, -1\}$ *uniformly randomly. Then, QNET-1 identifies*

the correct qualitative network with probability at least $1 - \frac{1}{n^\alpha}$*, if time series data beginning from* $O(\alpha \cdot \log n)$ *sets of initial values are given, where* $\alpha > 1$ *is any fixed constant.*

Note that ± 1 in the above can be replaced by other appropriate values. It seems that similar results still hold if initial values are chosen near uniformly randomly.

We can extend QNET-1 to equations of the form $\frac{dX_i}{dt} = a_i X_{j_i} + b_i$. Let

$$X_{i,j}^{(-,\max)} = \max\{X_j(t) | \Delta X_i(t) < 0\},$$
$$X_{i,j}^{(-,\min)} = \min\{X_j(t) | \Delta X_i(t) < 0\},$$
$$X_{i,j}^{(+,\max)} = \max\{X_j(t) | \Delta X_i(t) > 0\},$$
$$X_{i,j}^{(+,\min)} = \min\{X_j(t) | \Delta X_i(t) > 0\}.$$

Then, $X_{i,j}^{(-,\max)} < -\frac{b_i}{a_i} < X_{i,j}^{(+,\min)}$ holds if $a_i > 0$, and $X_{i,j}^{(+,\max)} < -\frac{b_i}{a_i} < X_{i,j}^{(-,\min)}$ holds if $a_i < 0$. Based on this observation, we obtain the following inference algorithm (QNET-2):

$E \leftarrow \{v_j \to v_i,\ v_j \dashv v_i \mid i = 1 \ldots n,\ j = 1 \ldots n\}$;
for $i = 1$ to n **do**
  for $j = 1$ to n **do**
    if $X_{i,j}^{(-,\max)} \geq X_{i,j}^{(+,\min)}$ **then** delete $v_j \to v_i$ from E;
    if $X_{i,j}^{(+,\max)} \geq X_{i,j}^{(-,\min)}$ **then** delete $v_j \dashv v_i$ from E;
  if $indegree(v_i) > 1$ **then**
    output "NOT IDENTIFIED" and **halt**;

Although we assumed linear differential equations, QNET-2 can be applied to differential equations of the form $\frac{dX_i(t)}{dt} = f(X_j(t))$ if $f(x)$ is a monotonically increasing or decreasing function.

For the size of time series data, we have not proved any theoretical result in this case. However, it seems that the correct network will be determined if sufficient data are given. At least, it is guaranteed that correct edges are not deleted.

Qualitative networks based on linear differential equations

Although the maximum indegree is assumed to be 1 (i.e. $K = 1$) in QNET-1 and QNET-2, we can develop an inference algorithm (denoted by QNET-3) for networks with no constraint on indegrees, using LP (linear programming).

In general, a linear differential equation has the following form:

$$\frac{dX_i(t)}{dt} = a_{i,1}X_1(t) + a_{i,2}X_2(t) + \cdots + a_{i,n}X_n(t) + b_i.$$

In this case, the corresponding qualitative network is defined by $V = \{v_1, \ldots, v_n\}$ and $E = \{v_j \to v_i | a_{i,j} > 0\} \cup \{v_j \dashv v_i | a_{i,j} < 0\}$.

D'haeseleer *et al.* (1999) used the linear regression method in order to determine the parameters. However, for that purpose, we should know precise values of $\frac{dX_i(t)}{dt}$. Therefore, instead of linear regression, we use linear programming (LP).

For each X_i, we make a set of linear inequalities as follows. If $\frac{dX_i(t)}{dt} > \rho$ where ρ is some constant, we make the following inequality:

$$a_{i,1}X_1(t) + \cdots + a_{i,n}X_n(t) + b_i > 0.$$

If $\frac{dX_i(t)}{dt} < -\rho$, we make the inequality in which '> 0' is replaced by '< 0'. Next, solving the set of linear inequalities by LP, we determine values of $a_{i,j}$ and b_i. Then, we let $v_j \to v_i$ if $a_{i,j} > 0$ and we let $v_j \dashv v_i$ if $a_{i,j} < 0$.

This LP-based method can also be applied to the case where the maximum indegree is bounded. For example, in the case of $K = 2$, we examine differential equations of the form $\frac{dX_i(t)}{dt} = a_{i,j}X_j(t) + a_{i,k}X_k(t) + b_i$ for all triplets (i, j, k). Although much longer time may be required, the values of $a_{i,j}$ and b_i will be determined more precisely. It should be noted that the time complexity is still $O(n^{K+1}m)$ by using linear time algorithms for LP in fixed dimensions (Motowani and Raghavan, 1994).

In the noisy case, the LP solver may fail to determine the values of $a_{i,j}$ and b_i since there may be no feasible solution. In such a case, robust LP (Bennett and Mangasarian, 1992) might be useful.

Inference of S-systems

The S-system (*synergistic* and *saturable* system) has been developed for modeling various biological systems (Irvine and Savageau, 1990; Savageau, 1991). S-systems have been successfully applied to the analysis of biochemical pathways, genetic networks and immune networks. For example, the well known Michaelis–Menten equation, which expresses enzymatic reaction involving one substrate and one product, is obtained from an S-system using equilibrium state approximation.

An S-system is a set of *nonlinear* differential equations of the form

$$\frac{dX_i(t)}{dt} = \alpha_i \prod_{j=1}^{n} X_j(t)^{g_{i,j}} - \beta_i \prod_{j=1}^{n} X_j(t)^{h_{i,j}}$$

where α_i and β_i are multiplicative parameters called *rate constants* and $g_{i,j}$ and $h_{i,j}$ are exponential parameters called *kinetic orders*.

Since S-systems are nonlinear, we can not apply linear regression to inference of S-systems. Tominaga and Okamoto (1998) applied a GA (genetic algorithm) to inference of S-systems with a few parameters. However, it

is unclear whether their method can be extended for large S-systems.

Using the idea of the LP-based method, we developed a simple method (denoted by SSYS-1) for inference of S-systems. Assume that $\frac{dX_i(t)}{dt} > 0$ at time t. By taking 'log' of each side of $\alpha_i \prod X_j(t)^{g_{i,j}} > \beta_i \prod X_j(t)^{h_{i,j}}$, we have

$$\log \alpha_i + \sum_{j=1}^{n} g_{i,j} \log X_j(t) \; > \; \log \beta_i + \sum_{j=1}^{n} h_{i,j} \log X_j(t).$$

Since $X_j(t)$ are known data, this is a linear inequality if we treat $\log \alpha_i$ and $\log \beta_i$ as parameters. In the case of $\frac{dX_i(t)}{dt} < 0$, we can obtain a similar inequality. Therefore, solving these linear inequalities by LP, we can determine parameters.

However, parameters are not determined uniquely even if many data are given, because the inequality can be rewritten as $(\log \alpha_i - \log \beta_i) + \sum (g_{i,j} - h_{i,j}) \log X_j(t) > 0$. Therefore, only relative ratios of $\log \alpha_i - \log \beta_i$ and $g_{i,j} - h_{i,j}$ are determined. However, this information is useful for qualitative understanding of S-systems. Since it seems that $g_{i,j} \neq h_{i,j}$ holds for most (i, j), the fact that $|g_{i,j} - h_{i,j}|$ is not small means that X_i is influenced by X_j (i.e. $v_j \rightarrow v_i$ or $v_j \dashv v_i$).

Computational results

We have implemented BOOL-2, QNET-1 and SSYS-1 using C language. Since we did not have an appropriate data set (see Discussion for the reason), we used artificial time series data. Since QNET-1 is too simple and SSYS-1 is much more complex than QNET-1, we show results on BOOL-2 and SSYS-1.

Results on noisy Boolean networks

We performed computational experiments on BOOL-2, using a Sun Ultra Enterprise 10000 (with 64 CPUs). Since the result of preliminary experiment showed that p_{noise} did not strongly affect the number of INPUT/OUTPUT patterns if $p_{\text{noise}} < \frac{1}{2}\theta$, we examined cases of $n = 10, 20, 40, 80, 160$, $\theta = 0.08, 0.10, 0.12$, where $K = 2$ and $p_{\text{noise}} = 0.04$ were fixed. Note that these values of θ and p_{noise} are larger than those in Theorem 2.

Figure 2 shows the number m of INPUT/OUTPUT patterns required to identify the underlying Boolean network uniquely, where the average number over ten randomly generated Boolean networks is shown for each case. It is seen that the numbers are proportional to $\log n$. Although the numbers are larger than those in the noiseless case (Akutsu *et al.*, 1999), the ratios are not large (less than three).

Results on S-systems

We performed computational experiments on SSYS-1, using a Sun Ultra-2 Workstation (with one CPU). In order to solve LP, we used commercial software SOPT (SAITECH Inc., 1998).

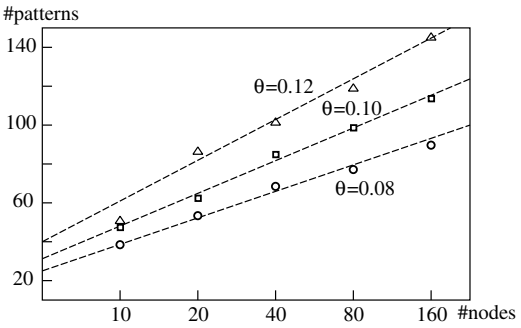

Fig. 2. Result on the number of expression patterns required to identify the noisy Boolean network of $K = 2$ correctly. Note that the X-axis is log scaled.

First we examined the following simple cases of $n = 2$, where case (A) was examined by Tominaga and Okamoto (1998) too.

	i	α_i	$g_{i,1}$	$g_{i,2}$	β_i	$h_{i,1}$	$h_{i,2}$
(A)	1	3.0	0.0	−2.5	3.0	0.125	0.0
	2	3.0	2.5	0.0	3.0	0.0	0.125
(B)	1	3.0	0.0	−2.5	3.0	1.25	0.0
	2	3.0	2.5	0.0	3.0	0.0	1.25

Time series data beginning from randomly generated initial values in $[0.5, 2.0]$ were used as input data. The Euler method was used to generate the time series data, where $\Delta t = 0.02$ was used. Since SSYS-1 could only compute relative values of $g_{i,j} - h_{i,j}$, we compared the ratios $r_1 = \frac{g_{1,1} - h_{1,1}}{g_{1,2} - h_{1,2}}$ and $r_2 = \frac{g_{2,2} - h_{2,2}}{g_{2,1} - h_{2,1}}$. Table 1 shows the result, where average values and standard deviations over 20 trials are shown. m denotes the total number of time points in the data, where 50 point data are generated from each set of initial values.

In each case, parameters were inferred within 1 s, which was much faster than the GA-based methods (Tominaga and Okamoto, 1998). On the other hand, the errors (in case (A)) were larger than those inferred by the GA-based method, but this is not a serious problem because we do not aim at determining precise values. We only want to know whether each $|g_{i,j} - h_{i,j}|$ is relatively large or small. Note that the errors are small for $m = 50$ in case (B), whereas the errors are not small even for $m = 500$ in case (A). This observation suggests that good values are not inferred if parameters in the different levels are included (note that $g_{2,1} = 2.5$ whereas $h_{1,1} = 0.125$ in case (A)).

Table 1. Ratios of parameters inferred by the LP-based method for S-systems with two variables.

		Correct	$m = 1 \times 50$	$m = 5 \times 50$	$m = 10 \times 50$
(A)	(r_1, σ)	(0.05, -)	(0.129, 0.032)	(0.081, 0.009)	(0.077, 0.011)
	(r_2, σ)	(-0.05,-)	(-0.261,0.232)	(-0.086,0.023)	(-0.085,0.011)
(B)	(r_1, σ)	(0.5, -)	(0.653, 0.099)	(0.598, 0.054)	(0.574, 0.040)
	(r_2, σ)	(-0.5,-)	(-0.648,0.108)	(-0.568,0.032)	(-0.538,0.029)

Table 2. Average ratios of correctly identified nodes for randomly generated S-systems with 10 variables

m	25×20	50×20	100×20
$K = 2$	30%	86%	100%
$K = 4$	26%	69%	87%

Next we examined whether or not qualitative relations are correctly inferred, by applying SSYS-1 to the case of $n = 10$ and $K = 2$ and the case of $n = 10$ and $K = 4$. Note that only the case of $n = 2$ was examined by Tominaga and Okamoto (1998). In these cases, we did not try to infer precise values of parameters, but we tried to infer whether or not X_i is influenced by X_j. We say that the set of input nodes $\{v_{i_1}, \ldots, v_{i_K}\}$ to v_i is correctly inferred if SSYS-1 outputs the same set for v_i, where we say that v_j is an input node to v_i if $h_{i,j} \neq 0$ and $g_{i,j} \neq 0$ hold in the original S-system. We inferred the input nodes by the following rule: v_j is an input node to v_i if $r_{i,j} > 0.1$, where $r_{i,j}$ is defined by$r_{i,j} = |\hat{g}_{i,j} - \hat{h}_{i,j}|/\max\{|\hat{g}_{i,j'} - \hat{h}_{i,j'}| \mid j' = 1, \ldots, n\}$, and $\hat{g}_{i,j}$ and $\hat{h}_{i,j}$ denote the values of parameters inferred by SSYS-1. We counted the number of nodes for which the sets of input nodes were correctly inferred. The result is shown in Table 2. In the table, the average ratios (%) of correctly inferred nodes over ten randomly generated S-systems are shown, where the following values were used: $\Delta t = 0.01$, $\alpha_i = \beta_i = 3.0$, $0.5 < |g_{i,j}| < 3.0$, $0.5 < |h_{i,j}| < 3.0$. Even in the case of $m = 100 \times 20$, each inference was made within 30 s (CPU time). From Table 2, it is seen that the sets of input nodes are correctly inferred for most nodes if m is large enough.

Finally, we examined the case of $n = 100$, $K = 4$, and $m = 1000 \times 20$. In this case, SSYS-1 inferred the sets of input nodes correctly for 96 nodes using less than 5 h (with one CPU), where $\Delta t = 0.005$. This result demonstrates the power of SSYS-1 because we are tackling a very hard problem, inference of nonlinear systems with more than $100 \times 100 \times 2$ parameters.

Discussion

In this paper, we proposed novel methods that might be useful for inferring qualitative relations in genetic networks and metabolic pathways from time series data. The most important feature of the methods is that they can be applied to nonlinear systems to some extent.

However, as shown in computational results, the proposed methods require many time series data beginning from different sets of initial values, where different sets correspond to different environments or different conditions. For example, it is estimated from Figure 2 that the number of INPUT/OUTPUT patterns (i.e. the number of time points) required to identify a noisy Boolean network of 6000 nodes is approximately 245 $\left(\approx 60 + (140 - 60) \cdot \frac{\log(6000) - \log(10)}{\log(160) - \log(10)}\right)$ in the case of $\theta = 0.12$. More data must be required in a practical case because time series data are not randomly generated. For inference of S-systems, many more data are required although we do not have a concrete method to estimate the size of data. Recall that 1000×20 point time series data (20 point data beginning from 1000 different sets of initial values) were required even for qualitative inference of an S-system with 100 nodes. Since time series data of 7 or 17 points beginning from a few different sets of initial values were only available (DeRisi *et al.*, 1997; Cho *et al.*, 1998), we could not apply the proposed methods to real data. However, many biological experiments are currently being performed using gene disruptions and gene overexpressions, and it is expected that a large number of more precise data will be available in the near future. For example, several hundreds of disruptants of *Saccharomyces cerevisiae* are being made by the group to which the third author of this paper belongs. If we could collect all time series data in the world, many more data would be available. Therefore, the assumption of existence of times series data beginning from many initial value sets will become realistic in the future. Of course, if we focused on a part of the network, the number of time series data required for the inference could be reduced.

Even if we have enough data, it seems difficult to apply the proposed methods to real data for the following reasons.

1. Regulation rules of genes may be much more complex than linear differential equations and S-systems although S-systems are based on the power-law formalism describing mass action in chemistry (Savageau, 1991).
2. The formalism using differential equations implicitly assumes that concentration of transcription factors (i.e. proteins made from mRNA) can be observed, but it is much more difficult to monitor the concentration of proteins, and there may be a long delay of hours (for eukaryotes) between mRNA and protein.
3. The proposed methods for differential equations are not robust for noises.
4. The proposed methods, especially the method for S-systems, are not fast enough for handling many genes (e.g. $n > 1000$).

Although these are serious problems, there is much room for improvements in the proposed methods. For example, the differential equations might be modified so that the effect of time delay is taken into account. This modification might be useful against item 2 of the above list and should be studied. As previously mentioned, robust LP (Bennett and Mangasarian, 1992) might be useful for handling noisy data.

It seems that the computation time is not a serious problem (at least for low indegree nodes) since the average computation time will be reduced significantly by using various heuristics and massively parallel computers. Although it may be still impossible to handle all genes simultaneously, it will be possible to handle several hundreds of genes. Handling of several hundreds of genes could be useful if we focused on a part of the network in which we were interested, or we grouped the genes into operons, and considered bacteria only (on average an operon contains three genes in bacteria, and these organisms have a few thousand genes).

Another drawback of the proposed methods is that complex enzymatic reactions (for example, three-stage enzymatic reactions) can not be handled directly: these reactions can not be represented exactly in the form of the S-system (Savageau, 1991). Therefore, development of the methods to infer complex enzymatic reactions is important future work.

Acknowledgements

The authors would like to thank anonymous referees for helpful comments. This work was partially supported by the Grant-in-Aid for Scientific Research on Priority Areas 'Genome Science' of the Ministry of Education, Science, Sports and Culture in Japan.

Appendix

Proof of Theorem 2

Since the algorithm tries to identify a Boolean function assigned to each node independently, we consider each node independently. We consider the probability that the algorithm does not output the correct Boolean function for a fixed node v_i. There are two cases where the algorithm makes an error: (A) the correct Boolean function is discarded, (B) an incorrect Boolean function is not discarded.

First we consider Case (A). Let $f_i(v_{i_1}, \ldots, v_{i_K})$ be the correct Boolean function assigned to v_i. Let M be the value of variable $mismatch$ for this function. Since incorrect OUTPUT values are generated with probability p, the expectation of M is pm. From the Chernoff bound $\mathrm{Prob}(M > (1+\delta)\mu) < \left(\frac{e}{1+\delta}\right)^{(1+\delta)\mu}$ (see Motowani and Raghavan, 1994) and $\mu = pm$ where μ denotes the expectation of M and $0 < \delta \leq 1$, the probability of (A) is

$$\mathrm{Prob}\left(M > \frac{m}{2^{2K+1}}\right) < (e \cdot p \cdot 2^{2K+1})^{\frac{m}{2^{2K+1}}},$$

by letting $\delta = 1 - \frac{1}{p \cdot 2^{2K+1}}$.

Next we consider Case (B). For any Boolean function $g_i(v'_{i_1}, \ldots, v'_{i_K})$ different from $f_i(v_{i_1}, \ldots, v_{i_K})$, we consider the probability that this function is not discarded. Since the expected value of $mismatch$ for g_i is at least $\frac{m}{2^{2K}}$ in the noiseless model (Akutsu *et al.*, 1999), the probability is at most the sum of the probabilities of the following two cases: (B1) the number of INPUT patterns where the value of f_i does not coincide with the value of g_i is less than $\frac{3}{4} \cdot \frac{m}{2^{2K}}$, (B2) the number of OUTPUT patterns such that $O(v_i) \neq f_i(I_j(v_{i_1}), \ldots, I_j(v_{i_k}))$ is greater than $\frac{1}{4} \cdot \frac{m}{2^{2K}} = \frac{m}{2^{2K+2}}$.

As in Case (A), the probability of (B2) is bounded above by $(e \cdot p \cdot 2^{2K+2})^{\frac{m}{2^{2K+2}}}$ for $p < \frac{1}{e \cdot 2^{2K+2}}$. Let X be the number of INPUT patterns considered in (B1). From the Chernoff bound $\mathrm{Prob}(X < (1-\delta)\mu) < e^{-\frac{\mu\delta^2}{2}}$ and the fact that $\mu \geq \frac{m}{2^{2K}}$, the probability of (B1) is bounded above by

$$\mathrm{Prob}\left(X < \frac{3}{4} \cdot \frac{m}{2^{2K}}\right) < e^{-\frac{1}{32} \cdot \frac{m}{2^{2K}}},$$

where we let $\delta = \frac{1}{4}$.

Since Case (A) is included in Case (B1) and there are at most $2^{2^K} \cdot n^K$ Boolean functions $g_i(v'_{i_1}, \ldots, v'_{i_K})$, the probability that the correct Boolean function is not output for node v_i is bounded above by

$$(e \cdot p \cdot 2^{2K+2})^{\frac{m}{2^{2K+2}}} + 2^{2^K} \cdot n^K \cdot e^{-\frac{1}{32} \cdot \frac{m}{2^{2K}}}.$$

Therefore, the total probability that correct Boolean functions are not identified for at least one node is bounded above by n times this value. Solving $n(e \cdot p \cdot 2^{2K+2})^{\frac{m}{2^{2K+2}}} < \frac{1}{2n^{\alpha}}$, we have

$$m > \frac{2^{2K+2}(1 + (\alpha + 1)\log n)}{\log \frac{1}{p} - \log e - (2K + 2)}.$$

Solving $2^{2^K} \cdot n^{K+1} \cdot e^{-\frac{1}{32} \cdot \frac{m}{2^{2K}}} < \frac{1}{2n^{\alpha}}$, we have

$$m > \frac{32 \cdot 2^{2K}}{\log e} \left((K + \alpha + 1)\log n + 2^K + 1 \right).$$

Combining these conditions, we have the bound in the theorem.

Proof of Theorem 3

First note that $\mathrm{sign}\left(\frac{\Delta X_i(0)}{\Delta}\right) = \mathrm{sign}(X_{j_i}(0))$ if $a_i > 0$, $\mathrm{sign}\left(\frac{\Delta X_i(0)}{\Delta}\right) = -\mathrm{sign}(X_{j_i}(0))$ if $a_i < 0$, otherwise $\frac{\Delta X_i(0)}{\Delta} = 0$. Note also that the value of $\frac{\Delta X_i(0)}{\Delta}$ depends only on the value of $X_{j_i}(0)$. Then, by regarding $(X_1(0), \ldots, X_n(0))$ as an INPUT pattern and $\left(\frac{\Delta X_1(0)}{\Delta}, \ldots, \frac{\Delta X_n(0)}{\Delta}\right)$ as an OUTPUT pattern, the identification of a simple qualitative network is almost the same as the identification of a Boolean network of $K = 1$. Since initial values are chosen from $\{1, -1\}$ uniformly randomly, the assumption in Theorem 1 is still valid in this case. Therefore, by letting $K = 1$ in Theorem 1, we obtain an $O(\alpha \cdot \log n)$ bound, where a constant factor is hidden by the 'O' notation.

References

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. *Proc. Pacific Symp. on Biocomputing*, **4**, 17–28.

Arkin,A., Shen,P. and Ross,J. (1997) A test case of correlation metric construction of a reaction pathway from measurements. *Science*, **277**, 1275–1279.

Bennett,K.P. and Mangasarian,O.L. (1992) Robust linear programming discrimination of two linear separable sets. *Optimization Method and Software*, **1**, 23–34.

Chen,T., He,H.L. and Church,G.M. (1999) Modeling gene expression with differential equations. *Proc. Pacific Symp. on Biocomputing*, **4**, 29–40.

Cho,R.J., Campbell,M.J., Winzeler,E.A., Steinmetz,L., Conway,A., Wodicka,L., Wolfsberg,T.G., Gabrielian,A.E., Landsman,D., Lockhart,D.J. and Davis,R.W. (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, **2**, 65–73.

de Kleer,J. and Brown,J.S. (1984) Qualitative physics based on confluences. *Artificial Intelligence*, **24**, 7–83.

DeRisi,J.L., Lyer,V.R. and Brown,P.O. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, **278**, 680–686.

D'haeseleer,P., Wen,X., Fuhrman,S. and Somogyi,R. (1999) Linear modeling of mRNA expression levels during CNS development and injury. *Proc. Pacific Symp. on Biocomputing*, **4**, 41–52.

Irvine,D.H. and Savageau,M.A. (1990) Efficient solution of nonlinear ordinary differential equations expressed in S-system canonical form. *SIAM J. Numer. Anal.*, **27**, 704–735.

Liang,S., Fuhrman,S. and Somogyi,R. (1998) REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. *Proc. Pacific Symp. on Biocomputing*, **3**, 18–29.

McAdams,H.H. and Shapiro,L. (1995) Circuit simulation of genetic networks. *Science*, **269**, 650–656.

Motowani,R. and Raghavan,P. (1994) *Randomized Algorithms*. Cambridge University Press, Cambridge.

Saitech Inc. (1998) *Smart Optimizer User's Guide*, Saitech (http://www.saitech-inc.com/math.htm).

Savageau,M.A. (1991) 20 years of S-systems. In Voit,E.O. (ed.), *Cononical Nonlinear Modeling. S-system Approach to Understanding Complexity* Van Nostrand Reinhold, New York, pp. 1–44.

Thieffry,D. and Thomas,R. (1998) Qualitative analysis of gene networks. *Proc. Pacific Symp. on Biocomputing*, **3**, 77–88.

Tominaga,D. and Okamoto,M. (1998) Design of canonical model describing complex nonlinear dynamics. *Proc. IFAC Int. Conf.*, **CAB7**, 85–90.

Yuh,C.-H., Bolouri,H. and Davidson,E.H. (1998) Genomic Cis-regulatory logic: experimental and computational analysis of a sea urchin gene. *Science*, **279**, 1896–1902.