Pacific Symposium on Biocomputing 5:290-301 (2000)

ALGORITHMS FOR INFERRING QUALITATIVE MODELS OF BIOLOGICAL NETWORKS

Tatsuya AKUTSU, Satoru MIYANO
Human Genome Center, Institute of Medical Science,
University of Tokyo, 4-6-1 Shirokanedai, Minato-ku, Tokyo 108-8639, Japan
{takutsu,miyano}@ims.u-tokyo.ac.jp

Satoru KUHARA
Graduate School of Genetic Resources Technology,
Kyushu University. Hakozaki 6-10-1, Higashi-ku, Fukuoka 812-8581, Japan
kuhara@grt.kyushu-u.ac.jp

Modeling genetic networks and metabolic networks is an important topic in bioinformatics. We propose a qualitative network model which is a combination of the Boolean network and qualitative reasoning, where qualitative reasoning is a kind of reasoning method well-studied in Artificial Intelligence. We also present algorithms for inferring qualitative networks from time series data and an algorithm for inferring S-systems (synergistic and saturable systems) from time series data, where S-systems are based on a particular kind of nonlinear differential equation and have been applied to the analysis of various biological systems.

1 Introduction

Due to the recent progress of the *DNA microarray* technology [1], it has become possible (to some extent) to measure the gene expression levels of most of the genes of an organism simultaneously. Recently, many studies have been done in order to develop computational methods for reconstructing underlying *genetic networks* from time series data of gene expression patterns.

Several studies have been done using the Boolean network [2], 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.* [2] developed the REVEAL algorithm (reverse engineering algorithm) for inferring genetic networks from state transition tables which correspond to time series data of gene expression patterns. We proved mathematically a small number (precisely, $O(\log n)$) of 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 [3].

Since there are many criticisms on the Boolean network approach, other models are becoming important. Thieffry and Thomas [4] studied a qualitative model, which is similar to our model. However, they did not give a concrete inference algorithm. Although other hybrid models are proposed [5,6], the methods for determine model parameters are unclear. Arkin *et al.* [7] proposed a statistical method to infer chemical networks. Chen *et al.* [8] and D'haeseleer *et al.* [9] proposed methods to infer genetic networks based on linear differential equations. However, no method seems to be sufficient.

Since the Boolean network model is too simple whereas the differential equation model is too specific, we propose a *qualitative network* model (Although it is similar to the model proposed by Thieffry and Thomas [4], there exist several differences). This model can be considered as a medium 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 [10]. In this model, regulation rules are represented as qualitative rules and embedded in network structures. We also present algorithms for inferring qualitative networks from time series data. Although the algorithms are based on linear differential equations, it can be applied to nonlinear models to some extent. Moreover, one of the algorithm can be applied to the inference of S-systems [11,12], 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 [11].

By the way, it is also important to develop inference algorithms robust for noises. Thus, we propose such an 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 the results of computational experiments. Finally, we conclude with future work.

2 Identification of Boolean Networks with Noises

2.1 Boolean Network and Its Identification

In this subsection, we briefly review the Boolean network model [2] and our previous result on its identification [3]. For details, see Ref. (3).

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. In a Boolean network, the expression pattern ψ_{t+1} at time $t+1$ is determined by Boolean functions F from the 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 an expression pattern (ψ_t) at some time t and each O_j corresponds to an expression pattern (ψ_{t+1}) at time $t+1$. 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 part 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 in order to identify input nodes to a high indegree node [3]. The importance of the constraint on the indegree is also pointed out in several papers [2,8].

In our previous work [3], we developed an algorithm (denoted by BOOL-1) for identifying Boolean networks. BOOL-1 is quite simple: it examines each node independently whether there exists a unique Boolean function consistent with given patterns. Moreover, we proved the following theorem [3]. Note that we do not require that Boolean networks are given randomly, but we require that INPUT patterns (I_i's) are given randomly.

Theorem 1. [Ref. (3)]
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.

2.2 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 as defined in Section 2.1. Then, a noisy Boolean network consists of $G(V, F)$ and p_{noise}, where p_{noise} is a constant such that $0 \leq p_{noise} < 1$. There is only one difference between the standard Boolean network and the noisy Boolean network: $O_j(v_i) = f_i(I_j(v_{i_1}), \cdots, I_j(v_{i_k}))$ holds for each node in a standard Boolean network, whereas $O_j(v_i) \neq f_i(I_j(v_{i_1}), \cdots, I_j(v_{i_k}))$ holds with probability $\leq p_{noise}$ for each node in a noisy Boolean network, where the probability is taken over all possible INPUT pattern I_j's.

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
  $count \leftarrow 0$;
  for all combinations of $K$ nodes $(v_{i_1}, \ldots, v_{i_K})$ do
    for all Boolean function $f$ with $K$ inputs do
      $mismatch \leftarrow 0$;
      for $j = 1$ to $m$ do
        if $O_j(v_i) \neq f_i(I_j(v_{i_1}), \ldots, I_j(v_{i_K}))$ then
          $mismatch \leftarrow mismatch + 1$;
      if $mismatch < \theta \cdot m$ then output $f(v_{i_1}, \ldots, v_{i_K})$ as a function
      assigned to $v_i$;  $count \leftarrow count + 1$;
  if $count \neq 1$ then output "NOT IDENTIFIED";  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 using the Chernoff bound[13], where the proof is omitted.

Theorem 2.
Assume that $p < \frac{1}{e\cdot 2^{2K+2}}$. If $O(2^{2K} \cdot (\alpha + K + 1) \cdot (1 + \frac{1}{\log\frac{1}{p} - \log e - (2K+2)}) \cdot \log n)$ 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.

Note that the assumption on p is too strong in Theorem 2. As suggested in Section 6, it seems that a similar property will hold for much larger p.

3 Qualitative Network Model

In Artificial Intelligence, *qualitative reasoning*[10] has been extensively studied. Theories of qualitative reasoning were developed for predicting and explaining the behavior of physical mechanisms in qualitative terms. In qualitative reasoning, instead of continuous real-valued variables, each variable is described quantitatively - taking on only small number of values, usually $+$, $-$, or 0. Instead of differential equations, qualitative equations are also used.

Using the concept of qualitative reasoning, we define a qualitative network model in the following way. 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, $v_j \rightarrow v_i$ 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).

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)". Then, in the *simplest model*, there is the following correspondence:

$$\begin{array}{ccll} v_j \to v_i & \Longleftrightarrow & \frac{dX_i}{dt} > 0 \text{ if } X_j > 0, & \frac{dX_i}{dt} < 0 \text{ if } X_j < 0, \\ v_j \dashv v_i & \Longleftrightarrow & \frac{dX_i}{dt} > 0 \text{ if } X_j < 0, & \frac{dX_i}{dt} < 0 \text{ if } X_j > 0. \end{array}$$

Although we use 0 as a threshold value here, we will use other appropriate values later. It should be noted that we intend to use qualitative networks *not for simulation, but for representing biological knowledge.* Thus, we do not need to know precise values of parameters in differential equations but we only 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.

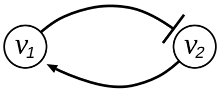
4 Inference of Qualitative Networks

4.1 A Simple Case

For ease of explanation, we begin with a very simple case, where it is to be extended to more realistic cases later. In this case, we assume that time series data of a biological system are produced according to the following simple system of linear differential equations:

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

For example, if $n = 2$, $j_1 = 2$, $j_2 = 1$, $a_1 = 1$ and $a_2 = -1$, then $X_1(t) = \sin(t+\theta)$ and $X_2(t) = \cos(t+\theta)$ where θ is determined from the initial values. The following qualitative network corresponds to this case.



The task of an inference algorithm is, given n and $X_i(t)$'s, to infer a qualitative network $G(V,E)$ consistent with $X_i(t)$'s. Note that, if sufficient time series data are given, the consistent network is uniquely determined (i.e., $E = \{v_{j_i} \to v_i \mid a_i > 0\} \cup \{v_{j_i} \dashv v_i \mid a_i < 0\}$).

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 $X_i(t)$'s 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 $X_i(t+\Delta) - X_i(t)$.

```
$E \leftarrow \{v_j \rightarrow v_i,\ v_j \dashv v_i \mid i = 1 \ldots n,\ j = 1 \ldots n\}$;
for $i = 1$ to $n$ do
  for $j = 1$ to $n$ do
    for $t = t_1$ to $t_1 + (m-1)\Delta$ do
      if $\Delta X_i(t) > 0$ and $X_j(t) < 0$ then delete $v_j \rightarrow v_i$ from $E$;
      if $\Delta X_i(t) < 0$ and $X_j(t) > 0$ then delete $v_j \rightarrow v_i$ from $E$;
      if $\Delta X_i(t) > 0$ and $X_j(t) > 0$ then delete $v_j \dashv v_i$ from $E$;
      if $\Delta X_i(t) < 0$ and $X_j(t) < 0$ then delete $v_j \dashv v_i$ from $E$;
  if indegree of $v_i > 2$ then output "NOT IDENTIFIED"; halt;
```

In practice, $\Delta X_i(t) > 0$ (resp. $\Delta X_i(t) < 0$) should be replaced by $\Delta X_i(t) > \rho$ (resp. $\Delta X_i(t) < -\rho$) using some threshold value ρ.

It is easy to see that this algorithm works in $O(n^2 m)$ time. Here, we briefly discuss about input time series data. It is easy to see that correct edges are not deleted (under the assumption that $sign(\frac{\Delta X_i(t)}{\Delta}) = sign(\frac{dX_i(t)}{dt})$, where $sign(x)$ denotes the sign of x). 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., $f(t_1)$'s) 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 and then it is expected that time series data from other attractors are newly given. The importance of using time series data beginning from multiple sets of initial values is discussed in Ref. (3). The following theorem holds regardless of 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 Theorem 3 can be replaced by other appropriate values. Although we do not examine details, it seems that similar results hold if initial values that are chosen near-uniformly randomly are used.

We can extend QNET-1 to equations of the form $\frac{dX_i}{dt} = a_i X_{j_i} + b_i$. Let $X_{i,j}^{(-,\max)} = \max\{X_j(t) | \Delta X_i(t) < 0\}$, $X_{i,j}^{(-,\min)} = \min\{X_j(t) | \Delta X_i(t) < 0\}$, $X_{i,j}^{(+,\max)} = \max\{X_j(t) | \Delta X_i(t) > 0\}$, and $X_{i,j}^{(+,\min)} = \min\{X_j(t) | \Delta X_i(t) > 0\}$. If $\frac{dX_i}{dt} = a_i X_{j_i} + b_i$ and $a_i > 0$, then $X_{i,j}^{(-,\max)} < X_{i,j}^{(+,\min)}$ holds. Moreover, $X_{i,j}^{(-,\max)} < -\frac{b_i}{a_i} < X_{i,j}^{(+,\min)}$ holds (see Fig. 1). Similarly, $X_{i,j}^{(+,\max)} < -\frac{b_i}{a_i} <$

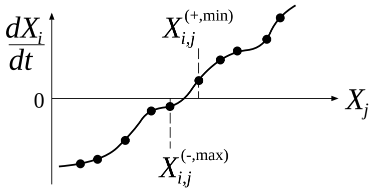


Figure 1: For any monotonically increasing function $f(x)$, the inference algorithm QNET-2 can be applied.

$X_{i,j}^{(-,\min)}$ holds in the case of $a_i < 0$. Based on this observation, we obtain the following algorithm (QNET-2).

```
$E \leftarrow \{v_j \rightarrow v_i,\ v_j \dashv v_i \mid i = 1 \ldots n,\ j = 1 \ldots n\}$;
**for** $i = 1$ to $n$ **do**
  **for** $j = 1$ to $n$ **do**
    **if** $X_{i,j}^{(-,\max)} \geq X_{i,j}^{(+,\min)}$ **then** delete delete $v_j \rightarrow v_i$ from $E$;
    **if** $X_{i,j}^{(+,\max)} \geq X_{i,j}^{(-,\min)}$ **then** delete delete $v_j \dashv v_i$ from $E$;
  **if** indegree of $v_i > 2$ **then** output "NOT IDENTIFIED"; **halt**;
```

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

4.2 An LP-based Method

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

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

$$\frac{dX_i(t)}{dt} = \alpha_{i,1}X_1(t) + \alpha_{i,2}X_2(t) + \ldots + \alpha_{i,n}X_n(t) + \beta_i\,,$$

where $\alpha_{i,1}$'s and β_i's are parameters to be inferred. D'haeseleer *et al.* [9] used the linear regression method in order to determine the parameters. However, for that purpose, we should know precise values of $\frac{dX_i(t)}{dt}$'s. Therefore, instead of linear regression, we use linear programming.

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

$$\alpha_{i,1}X_1(t) + \ldots + \alpha_{i,n}X_n(t) + \beta_i \ > 0.$$

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 parameters. Then, we let $v_j \rightarrow v_i$ if $\alpha_{i,j} > 0$ and we let $v_j \dashv v_i$ if $\alpha_{i,j} < 0$.

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 $\frac{dX_i(t)}{dt} = \alpha_{i,j}X_j(t) + \alpha_{i,k}X_k(t) + \beta_i$ for all triplets (i, j, k). Although much longer time may be required, parameters will be determined more precisely. It should be noted that the time complexity is still $O(n^{K+1}m)$ by using *theoretically efficient* algorithms for LP in fixed dimensions[13].

In a noisy case, LP solver may fail to determine the values of parameters. In such a case, *robust linear programming*[14] might be useful.

5 Inference of S-systems

In order to analyze biological systems, the S-system (*synergistic* and *saturable* system) has been developed[11]. S-systems have been successfully applied to the analysis of biochemical pathways, genetic networks and immune networks [11]. 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[9] to inference of S-systems. Tominaga and Okamoto[12] applied GA (Genetic Algorithm) to inference of S-systems with a few parameters. However, it is unclear whether their method can be extended for inference of large S-systems.

Using the idea of the LP-based method described in Section 4.2, we developed a method (denoted by SSYS-1) for inference of S-systems. The method is quite simple. 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 $X_j(t)$'s are known data, this inequality is linear if we treat $\log \alpha_i$'s and $\log \beta_i$'s 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 a lot of data are given, because the inequality can be re-written 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 $g_{i,j} - h_{i,j}$'s are determined (for each i). But, this information is useful for qualitative understanding of S-systems. Since $\prod X_j(t)^{g_{i,j}}$ contributes to the net production of X_i, $\prod X_j(t)^{h_{i,j}}$ contributes to the net degradation of X_i and it is not usual that X_j contributes to both the net production and the net degradation, either $g_{i,j} = 0$ or $h_{i,j} = 0$ holds for each (i,j) in most cases. Thus, the fact that $|g_{i,j} - h_{i,j}|$ is large means that X_i is influenced by X_j.

6 Computational Experiments

We have implemented BOOL-2, QNET-1 and SSYS-1 using C-language. Since we do not have appropriate data set, we use artificial time series data. Because of the space limit, we show results on BOOL-2 and SSYS-1.

6.1 Noisy Boolean Networks

We made computational experiments on BOOL-2, using SUN ULTRA ENTERPRISE 10000 (with 64CPU). The result of preliminary experiment showed that p_{noise} does not strongly affect the sample complexity (m) if $p_{noise} < \frac{1}{2}\theta$. Therefore, we examined cases of $n = 10, 20, 40, 80, 160$, $\theta = 0.08, 0.10, 0.12$, where $K = 2$ and $p_{noise} = 0.04$ are fixed. Note that these values of θ and p_{noise} are larger than those in Theorem 2.

Fig. 2 shows the number m of INPUT/OUTPUT patterns required to identify the underlying Boolean network uniquely, where the average number over randomly generated 10 Boolean networks is shown for each case. It is

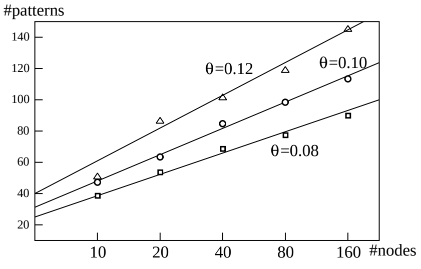


Figure 2: Result on the number of expression patterns required to identify the noisy Boolean network of $K = 2$ correctly. Note that X-axis is log-scaled.

seen that the numbers are proportional to $\log n$. Although the numbers are larger than in the noiseless case[3], the ratios are not large (< 3).

6.2 Inference of S-systems

We made computational experiments on SSYS-1, using a SUN ULTRA-2 Workstation (with 1 CPU). In order to solve LP, we used SOPT[15].

First we examined the following simple cases of $n = 2$, where case (A) was examined in Ref. (12) 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

As input data, time series data beginning from randomly generated initial values in $[0.5, 2.0]$ were used. The Euler method was used to generate the time series data, where $\Delta t = 0.02$ was used. Since SSYS-1 can only compute relative values of $g_{i,j} - h_{i,j}$'s, we compare the ratios $r_1 = \frac{g_{1,1}-h_{1,1}}{g_{1,2}-h_{1,2}}$ and $r_2 = \frac{g_{2,2}-h_{2,2}}{g_{2,1}-h_{2,1}}$. The following table 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.

		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)

In each case, parameters were inferred within 1 second, which is much faster than the GA-based algorithm[12]. On the other hand, the errors (in case (A)) are larger. But, it 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.

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 in Ref. (12). 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, using the method described in Section 5. We say that the set of input nodes $\{X_{i_1}, \cdots, X_{i_K}\}$ to X_i is *correctly inferred* if SSYS-1 outputs the same set for X_i, where we say that X_j is an input node to X_i if $h_{i,j} \neq 0$ and $g_{i,j} \neq 0$ hold in the original S-system. We count the number of nodes for which the sets of input nodes are correctly inferred. The result is shown in the table below. In the table, the average ratios (%) of correctly inferred nodes over 10 randomly generated S-systems are shown, where the following values are 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 can be done within 30 sec. (CPU time).

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

From this table, 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 hours (with 1 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.

7 Concluding Remarks

In this paper, we proposed novel methods which might be useful for inferring biological networks 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 experiments, 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. Since time series data of 7 or 17 points beginning from a few different sets of initial values were only available [1,16], we could not apply the proposed methods to real data. *It seems almost impossible to get more information than those obtained by clustering, if only a small number of time series data can be used.* However, many biological experiments are currently being done 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. Therefore, the assumption of existence of times series data beginning from many initial value sets will become realistic in the near future. Of course, much faster algorithms should be developed for handling a large amount of data.

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

Acknowledgments

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.

References

1. J.L. DeRisi, V.R. Lyer and P.O. Brown, *Science* **278**, 680 (1997).
2. S. Liang, S. Fuhrman and R. Somogyi, *Pacific Symp. on Biocomputing* **3**, 18 (1998).
3. T. Akutsu, S. Miyano and S. Kuhara, *Proc. Pacific Symp. Biocomputing* **4**, 17 (1999).
4. D. Thieffry and R. Thomas, *Pacific Symp. on Biocomputing* **3**, 77 (1998).
5. H.H. McAdams and L. Shapiro, *Science* **269**, 650 (1995).
6. C-H. Yuh, H. Bolouri and E.H. Davidson, *Science* **279**, 1896 (1998).
7. A. Arkin, P. Shen and J. Ross, *Science* **277**, 1275 (1997).
8. T. Chen, H.L. He and G.M. Church, *Proc. Pacific Symp. Biocomputing* **4**, 29 (1999).
9. P. D'haeseleer, X. Wen, S. Fuhrman and R. Somogyi, *Proc. Pacific Symp. Biocomputing* **4**, 41 (1999).
10. J. de Kleer and J.S. Brown, *Artificial Intelligence* **24**, 7 (1984).
11. D.H. Irvine and M.A. Savageau, *SIAM J. Numer. Anal.* **27**, 704 (1990).
12. D. Tominaga and M. Okamoto, *Proc. IFAC Int. Conf.*, CAB7, 85 (1998).
13. R. Motowani and P. Raghavan, *Randomized Algorithms*, Cambridge Univ. Press (1994).
14. K.P. Bennett and O.L. Mangasarian, *Optimization Method and Software* **1**, 23 (1992).
15. *Smart Optimizer User's Guide*, SAITECH Inc. (http://www.saitech-inc.com/math.htm) (1998).
16. R.J. Cho *et al.*, *Molecular Cell* **2**, 65 (1998).