Dynamic modeling of genetic networks using genetic algorithm and S-system

Shinichi Kikuchi[1,2,*], Daisuke Tominaga[1], Masanori Arita[1,2], Katsutoshi Takahashi[1] and Masaru Tomita[2]

[1]*Computational Biology Research Center (CBRC), National Institute of Advanced Industrial Science and Technology (AIST), 2-41-6 Aomi, Koto-ku, Tokyo, 135-0064, Japan and* [2]*Institute for Advanced Biosciences, Keio University, 14-1, Baba, Tsuruoka, Yamagata, 997-0035, Japan*

ABSTRACT

Motivation: The modeling of system dynamics of genetic networks, metabolic networks or signal transduction cascades from time-course data is formulated as a reverse-problem. Previous studies focused on the estimation of only network structures, and they were ineffective in inferring a network structure with feedback loops. We previously proposed a method to predict not only the network structure but also its dynamics using a Genetic Algorithm (GA) and an S-system formalism. However, it could predict only a small number of parameters and could rarely obtain essential structures. In this work, we propose a unified extension of the basic method. Notable improvements are as follows: (1) an additional term in its evaluation function that aims at eliminating futile parameters; (2) a crossover method called Simplex Crossover (SPX) to improve its optimization ability; and (3) a gradual optimization strategy to increase the number of predictable parameters.
Results: The proposed method is implemented as a C program called PEACE1 (Predictor by Evolutionary Algorithms and Canonical Equations 1). Its performance was compared with the basic method. The comparison showed that: (1) the convergence rate increased about 5-fold; (2) the optimization speed was raised about 1.5-fold; and (3) the number of predictable parameters was increased about 5-fold. Moreover, we successfully inferred the dynamics of a small genetic network constructed with 60 parameters for 5 network variables and feedback loops using only time-course data of gene expression.
Contact: kikuchi@sfc.keio.ac.jp

INTRODUCTION

Complex biological systems such as gene regulation networks, metabolic pathways, and signal transduction

---

*To whom correspondence should be addressed.

cascades are comprised of many interacting components. In many cases, the detailed molecular mechanisms that govern the interactions among system components are not well understood. Generally, it is difficult to model these complex processes mathematically. Since most of them are nonlinear, their description requires a representation that is general enough to capture the essence of experimentally observed responses.

Many gene regulation models have been proposed; they are listed in the recent reviews (D'haeseleer *et al.*, 2000; Jong, 2002). The models can vary from the very abstract, e.g. Boolean networks (Akutsu *et al.*, 2000), to the very concrete, such as fully biochemical interaction models with stochastic kinetics (Arkin *et al.*, 1998). The former approach is mathematically tractable, and its simplicity allows examination of large systems. However, it cannot infer networks with feedback loops. The latter approach is better suited to the biochemical reality and may look realistic enough for the experimental biologists. However, due to its computational complexity, analysis is necessarily restricted to very small systems.

The Boolean approximation assumes highly cooperative binding and/or positive feedback loops to make the variables saturate in the ON or OFF position. However, examination of real gene expression data shows that gene expression levels tend to be continuous rather than binary. Furthermore, important concepts in control theory that seem indispensable for gene regulation systems either cannot be implemented with Boolean variables, or result in radically different dynamic behavior.

Others have reported dynamic modeling methods of gene expression data. Reinitz and Sharp (1995) employed their dynamic equation (connectionist model) and simulated annealing. Holter *et al.* (2001) used a time translational matrix for modeling these data within a linear framework. However, the character of such equations is the topic of much current research.

One of the well-studied dynamic and continuous approaches is the 'S-system' (Savageau, 1976; Voit, 2000). It is a type of power-law formalism and is based on a particular type of ordinary differential equation in which the component processes are characterized by power-law functions. The structure of the S-system is rich enough to capture many relevant biological dynamics. For example, Shiraishi and Savageau (1976) have developed a TCA cycle model using the S-system, and others have also addressed the S-system (Alves and Savageau, 2000; Voit and Radivoyevitch, 2000). It provides a great advantage in terms of system analysis and control design because it allows the customizing of analytical and computational methods. Using S-system parameters, steady-state evaluation, control analysis, and sensitivity analysis of a given system are established mathematically (Voit, 2000). On the other hand, the S-system has a major disadvantage; all of its large number of parameters must be estimated. This number is $2n(n + 1)$, where n is the number of state variables (ex. concentration). The estimation of these parameters often causes bottlenecks, and fitting the model to experimentally observed data is not simple.

We have proposed a technique for the dynamic modeling of complex biosystems by combining a Genetic Algorithm (GA) and the S-system (Tominaga and Okamoto, 1998; Tominaga *et al.*, 2000). GAs perform a global search that avoids local minima, and they can estimate many parameters simultaneously (Goldberg, 1989). In fact, the GA has been applied to parameter estimations of biosystems, and its usefulness and practicality have been already documented (Gilman and Ross, 1995; Park *et al.*, 1997; Pinchuk *et al.*, 2000). However, estimation of the S-system parameters is too difficult for optimization with the conventional simple GA. While the use of real-coded GA (Tominaga *et al.*, 2000) was a possible alternative, it was suboptimal in that the convergence rates were low, and only a very small number of parameters, i.e. 12 parameters for 2 variables, were predictable. Furthermore, since it aimed mainly at reproducing given time-courses, it was rare that correct 'skeletal structures' (the smallest or almost smallest network structures in ones that can reproduce the same time-course) were obtained.

In the work presented here, we modified the basic method and achieved 3 remarkable improvements: (1) We improved an evaluation function of GA that aims at eliminating futile parameters by adding the sum of the absolute values of model parameters to the conventional error function. We show here that this resulted in an approximately 5-fold increase in the convergence rate and an approximately 1.5-fold increase in the optimization speed. (2) We employed a novel crossover method, the Simplex Crossover (SPX; Tsutsui *et al.*, 1999). We show that this improved the search ability, i.e. the number of predictable parameters and the optimization speed. (3) We

introduced a gradual optimization strategy. Since the GA is effective for global searches, we attempted to gradually identify parameters unnecessary for modeling. This strategy performs iterative identification of unnecessary parameters, fixes them to 0, and reoptimizes the other parameters.

By applying (2) and (3), we achieved an approximately 5-fold increase over the basic method in the number of predictable parameters. Moreover, we successfully inferred the dynamics of a small genetic network constructed with 5 variables (60 parameters) and feedback loops automatically from only time-course data of gene expressions.

BASIC METHOD

Here we briefly describe our previous modeling method using the GA for the S-system (Tominaga and Okamoto, 1998; Tominaga *et al.*, 2000).

S-system formalism

The S-system (Savageau, 1976; Voit, 2000) is a type of power-law formalism described as follows:

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}}, \tag{1}$$

where n is the number of state variables or reactants X_i (ex. concentration), and i, j ($1 \leq i, j \leq n$) are suffixes of state variables. The terms g_{ij} and h_{ij} are the interactive effect of X_j to X_i. The first and second terms represent all influences that increase and decrease X_i, respectively. In the biochemical engineering context, the non-negative parameters α_i and β_i are called rate constants, and the real-value exponents g_{ij} and h_{ij} are referred to as kinetic orders. Here, α_i, β_i, g_{ij}, and h_{ij} are parameters that must be estimated. For numerical integration of the S-system, there is a high-speed algorithm called 'Evaluation and Simulation of Synergistic Systems (ESSYNS)' (Irvine and Savageau, 1990).

Optimization of S-system parameters using GA

The model of the S-system, expressed by Equation (1), contains many real-number parameters (α_i, β_i, g_{ij} and h_{ij}). Their total number is $2n(n + 1)$, where n is the number of state variables. When the dynamic actions of the model are calculated by the numerical solution method from Equation (1), they cannot be determined sequentially; all must be determined simultaneously because the parameters affect others mutually. Moreover, since the number of parameters that must be determined is $O(n^2)$, optimization is almost impossible by analytical methods, e.g. a conjugate gradient method, if the number of parameters is large. We used the GA because it can determine many parameters simultaneously with high accuracy, and selected the real-coded GA (Jonikow and

Michalewicz, 1991) because it improves the optimization speed compared to the conventional binary GA.

Instead of the conventional binary- or Gray-expression, the real-coded GA uses the real numerical value expression as a gene. One individual expresses one model, i.e. one individual is expressed with a group of real-value parameters as $2n(n + 1)$ real-value parameters, representing α_i, β_i, g_{ij} and h_{ij}. As an evaluation function of the basic method to reproduce time-courses, the following relative squared error E was employed:

$$E = \sum_{i=1}^{n} \sum_{t=1}^{T} \left(\frac{X_i'(t) - X_i(t)}{X_i(t)} \right)^2, \qquad (2)$$

where n is the number of experimentally observable state variables, T is the number of sampling points of the experimental data, $X_i'(t)$ is the numerically calculated time-course at time t of a state variable X_i, and $X_i(t)$ represents the experimentally observed time-course at time t of X_i. For convenience, when $X_i(t)$ is 0 in the calculation of E, the denominator is set to 0.1. The fitness F of each individual is the reciprocal of E; the optimization strategy is to search for the individual that maximizes F. In the next section, we describe our improvement evaluation function.

Our previous GA for the S-system is described as follows:

(1) Initialization
Generate P individuals and assign uniform random numbers within the search space to each parameter of all individuals.

(2) Evaluation
Calculate fitness F for each individual. Keep the individual with the maximum fitness (the elite individual) separate and do not apply evolutionary operations (3)–(5). This represents the elite strategy. Complete optimization when the fitness of the elite individual is considered sufficiently high. Finish optimization when the number of generations reaches the upper limit $G_{\max}$.

(3) Selection
Select $(P - 1)$ individuals from the group of individuals based on probability determined by fitness. Replace the group of individuals with $(P-1)$ selected individuals and the elite individual, using ranking selection to choose these individuals. The method selects each individual according to the selection probability p_i

$$p_i = \frac{1}{P} \left(\eta^+ - (\eta^+ - \eta^-) \frac{i - 1}{P - 1} \right), \qquad (3)$$

where i is the fitness ranking of the individual, and η^+ and η^- are the control parameters of the selection procedure.

(4) Crossover
From the two selected parents, each matrix element of the offspring is alternately chosen from the corresponding elements of either parent at a probability.

(5) Mutation
Mutate all values according to a probability. Mutation is realized by adding a normal distribution random number with the average of 0 and the distribution of d. Set the distribution d to the initial value d_1. Change d to d_2 $(< d_1)$ when the fitness of the elite individual remains unchanged for G_1 generations. Next, change d to d_3 $(> d_1)$ when the fitness remains unchanged for G_2 generations. Then change d back to d_1 when the fitness remains unchanged for G_3 generations. If the fitness is improved, change d back to the base d_1. Multiply the mutation rate by k (> 1) when the fitness remains unchanged for G_k generations. However, fix it when it reaches the upper limit for the optimization $m_{\max}$. If the fitness is improved, change it back to the base m_0. Apply this mutation to all offspring parameters.

(6) Back to (2)
As there are many parameters for optimization that should be determined, set up suitable values by trial and error depending on the given problem.

PROPOSED IMPROVEMENTS

Because the basic method could predict only a very small number of parameters (Tominaga *et al.*, 2000) and the convergence rate was low, we applied our improvements; improved evaluation function of the GA, the crossover method, and our optimization strategy.

Pruning method

As the evaluation function of the basic method focuses on reproducing given time-courses, it converges to multiple local minima and rarely attains skeletal structures. Our aim is not only at the reproduction of given time-courses but also the detection of skeletal structures with the expectation of finding unknown pathways.

We propose the following evaluation function E

$$E = \sum_{i=1}^{n} \sum_{t=1}^{T} \left(\frac{X_i'(t) - X_i(t)}{X_i(t)} \right)^2 + cnT \left\{ \sum_{i,j} |g_{ij}| + \sum_{i,j,i \neq j} |h_{ij}| \right\}, \quad (4)$$

where c is the weighted coefficient that balances the two evaluation terms. The first term on the right-hand side of Equation (4) is the same evaluation function as in the basic method described in Equation (2). The second term is the sum of the absolute values of model

parameters g_{ij} and h_{ij}. Using the second term, skeletal structures are obtained. We call this the 'pruning term'. Our algorithm is comprised of the following two stages: As the first step, optimize using Equation (4) for G_f generations, and obtain rough skeletal structures while retaining optimization accuracy. As the second step, improve the optimization accuracy, i.e. the fitness value in the meaning of the relative error, keeping the skeletal structure using Equation (2) for G_s generations.

Following is a discussion of the meaning of improved evaluation function. The first term aims at the reproduction of given time-courses. The pruning term, on the other hand, expresses the futility of model parameters. Since the value of the term is small when the sum of the absolute value is small, the pruning term expresses that it is better for almost all parameters to be 0. While the basic method assumes all combinations so as not to decrease possibility, most actual biological networks are sparse, and the S-system precondition prepares many futile parameters. Parameters with value 0 do not influence the other state variables, so that the corresponding pathway can be cut off in the pathway diagram. The pruning term realizes this purpose using mathematical equations. Moreover, under the concept of regularization, the pruning term is equivalent to the Laplacian regularization term (Williams, 1995). Similarly, the sum of the squared parameter value can also be considered. It is equivalent to the Gaussian regularization term (Mackay, 1995). However, parameter values do not truly become 0 by the Gaussian regularization term; for this reason we employ the Laplace regularization term. Our improved technique can prevent over-fitting and improve the generalization ability because it not only reproduces time-courses but also tries to minimize and simplify the network structure by applying the pruning method. It is effective for noisy real-world data that have few sampling points. In this case study we investigated whether skeletal structures can be obtained using a regularization term. We propose that it is better to choose regularization terms flexibly, depending on the given problems.

Simplex crossover

Tsutsui *et al.* (1999) proposed SPX, and Higuchi *et al.* (2000) showed that it improved the optimization speed. We used the SPX to improve the optimization speed compared with the basic method.

SPX operations are as follows:

(1) Choose m parents P_k $(k = 1, 2, \ldots, m)$ according to the generational model used and calculate their center of gravity G

$$G = \frac{1}{m} \sum_{k=1}^{m} P_k. \qquad (5)$$

(2) Generate random numbers r_k

$$r_k = u^{\frac{1}{k+1}}, \quad (k = 1, 2, \ldots, m - 1), \qquad (6)$$

where u is a uniform random number $\in [0, 1]$.

(3) Calculate x_k and C_k

$$x_k = G + \varepsilon(P_k - G), \quad (k = 1, 2, \ldots, m) \qquad (7)$$

$$C_k = \begin{cases} 0, & (k = 1) \\ r_{k-1}(x_{k-1} - x_k + C_{k-1}), & (k = 2, 3, \ldots, m) \end{cases} \qquad (8)$$

where ε is the expansion rate, a control parameter of SPX.

(4) Generate an offspring C

$$C = x_m + C_m. \qquad (9)$$

SPX has the following features (Tsutsui *et al.*, 1999):

(1) Since a simplex is basically independent of coordinate systems, the SPX can inherit this independence.

(2) Since offspring vector values are uniformly sampled around m parent vector values, they inherit the characteristics of parents and sampling can be assumed to reflect a certain linkage among the parameters.

(3) As a result, SPX balances between exploration and exploitation in generating offspring, and it works well on functions with multimodarity and/or epistasis among the parameters.

(4) Since the uniform sampling in a simplex can be performed with simple procedures, SPX is a simple and non-time consuming operator.

Gradual optimization strategy

Although it is difficult to optimize all the parameters of the S-system at once, parameters of comparatively lower importance, which become almost 0, are detectable. If these values are fixed to 0 and optimization is again done from the beginning, more parameters of lower importance are detected. The method of iteratively performing this procedure is named the 'gradual optimization strategy'.

The rule for judging unnecessary parameters in the gradual optimization strategy is as follows: If the trials are repeated with different initial values, many solution candidates are obtained as local minima. Therefore, it is difficult to determine deleted parameters automatically. One solution is to use the result of best fitness in multiple trials. However, this may delete necessary parameters. To avoid this, we performed double optimization using multiple elite individuals over a threshold from different trials. In fact, although there were multiple local minima, the essential parameters remained intact. Double optimization could automatically detect the essential parameters by optimizing multiple local minima once again. This avoided the deletion of necessary parameters based on one elite individual obtained by the first trial [see (P1) below].

Table 1. Parameters that determine the dynamic action of the S-system. These values were determined artificially in order to realize the network of Fig. 1 as a case study. For details, refer to (Hlavacek and Savageau, 1996)

i	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}
1	5.0	0.0	0.0	1.0	0.0	-1.0	10.0	2.0	0.0	0.0	0.0	0.0
2	10.0	2.0	0.0	0.0	0.0	0.0	10.0	0.0	2.0	0.0	0.0	0.0
3	10.0	0.0	-1.0	0.0	0.0	0.0	10.0	0.0	-1.0	2.0	0.0	0.0
4	8.0	0.0	0.0	2.0	0.0	-1.0	10.0	0.0	0.0	0.0	2.0	0.0
5	10.0	0.0	0.0	0.0	2.0	0.0	10.0	0.0	0.0	0.0	0.0	2.0

The procedure of our gradual optimization strategy is as follows:

(P1) Obtain skeletal structures by trials with different initial values using the GA for the S-system. Collect the higher-rank individuals. In our experience, they have essential and common links.

(P2) Use the higher-rank individuals as the subsequent initial individual groups. Apply the GA for the S-system to them and obtain the elite solution by double optimization; this detects common parameters from the multiple local minima.

(P3) From the result of (P2), fix parameters judged unnecessary to 0, and return to (P1). In this way, the optimization procedure gradually becomes simpler. The process is finished when no more parameters are judged unnecessary.

A problem with the basic method lies in the fact that the S-system contains many futile parameters with a value of 0 whose optimization is difficult. An important point in our technique is that a given problem is divided into several simple problems and the importance of each parameter gradually becomes clear.

EXPERIMENTAL RESULTS

To confirm the validity of our improved algorithm, we conducted an experiment, the modeling of the dynamics of the small-scale gene network shown in Figure 1 (Hlavacek and Savageau, 1996) as a case study. We described the system shown in Figure 1 using the S-system notation of Equation (1) with the numerical parameters shown in Table 1.

Our computational experiment used 50 time-course data. They were artificially prepared using the values of Tables 1 and 2. An example of the data is shown in Figure 2. Other data are calculable using the values of Tables 1 and 2 with the same sampling points.

The conditions of our experiments were as follows: $n = 5$, $T = 10$, $P = 65$, $G_{\max} = 35\,000$, α_i and $\beta_i \in [0, 15.0]$, g_{ij} and $h_{ij} \in [-3.0, 3.0]$, $\eta^+ = 2.0$, $\eta^- = 0.0$, $G_1 = 10$, $G_2 = 5$, $G_3 = 5$, $d_1 = 3.0$, $d_2 = 0.375$ and $d_3 = 15.0$ for α_i and β_i, $d_1 = 1.2$, $d_2 = 0.15$ and $d_3 = 6.0$ for g_{ij} and h_{ij}, $m_0 = 0.01$, $G_k = 2$, $k = 1.01$,
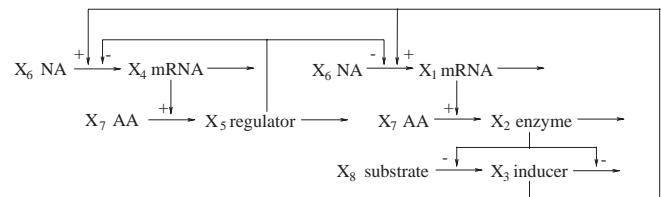

Fig. 1. Genetic network used in our computational experiments (Hlavacek and Savageau, 1996). This is a typical gene interaction system consisting of two genes (genes 1 and 4). X_1 is an mRNA produced from gene 1, X_2 is an enzyme protein it produces, and X_3 is an inducer protein catalyzed by X_2. X_4 is an mRNA produced from gene 4 and X_5 is a regulator protein it produces. Positive feedback from the inducer protein X_3 and negative feedback from the regulator protein X_5 are assumed in the mRNA production processes of genes 1 and 4. This model has been developed to analyze the interaction of regulator and effector genes. In our study, this model was used an example that is well-studied and has feedback loops.

Table 2. 10 sets of initial concentrations used in our computational experiments. These values were also prepared artificially

Trial	X_1	X_2	X_3	X_4	X_5
1	0.70	0.12	0.14	0.16	0.18
2	0.10	0.70	0.14	0.16	0.18
3	0.10	0.12	0.70	0.16	0.18
4	0.10	0.12	0.14	0.70	0.18
5	0.10	0.12	0.14	0.16	0.70
6	0.70	0.70	0.14	0.16	0.70
7	0.10	0.70	0.70	0.16	0.18
8	0.10	0.12	0.70	0.70	0.18
9	0.10	0.12	0.14	0.70	0.70
10	0.70	0.12	0.14	0.16	0.70

$m_{\max} = 0.5$, $G_f = 5000$, $G_s = 30\,000$, $\varepsilon = 1$, and $m = 65$. The applied values of c are shown in Table 3. The computer environment was as follows: AIST CBRC Magi Cluster with 1040 CPUs and Pentium III 933 MHz. Our algorithms were implemented in C language. The time required for one loop was approximately 10 h.
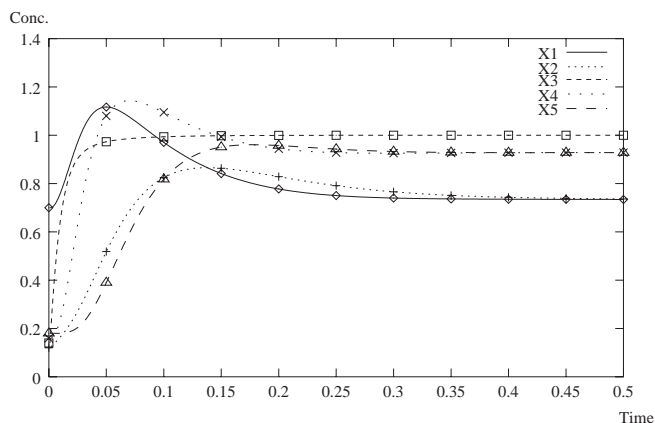
Fig. 2. Applied time-courses of trial 1. From each time-course, 10 points were sampled for optimization; dots denote sampling points. These points were selected artificially as a case study. This system is highly stable that converges on the same values at Time = 0.5. It diverges when it strays from the convergence domain.

Table 3. Applied values of c in each application. These values are determined over 50 trials started with different random seeds

Loop	P1	P2
1	0.15	0.19
2	0.07	0.15
3	0.01	0.11
4	0.005	0.07
5	0.003	0.01
6	0.001	0.001
7	0.0007	0.0009

Parameters estimated by PEACE1 are shown in Table 4. We found that dynamics modeling was attained. This result was obtained by applying the gradual optimization strategy 7 times. The number of parameters inferred to have value 0 was 17, 23, 26, 30, 33, 36 and 36 at the sequential iterations. The first number, 17, reflects the performance of SPX without application of the gradual optimization strategy; the correct number is 37 as shown in Table 1. The results in Table 5 show that both, the convergence rate for obtaining skeletal structures and the optimization speed, were improved by using our technique. The pruning method is equivalent to having the foreknowledge that general genetic network structure are sparse by giving an evaluation function that aims at skeletal structures. Not only was the ability to acquire skeletal structures increased, but the convergence rate and the optimization speed were also improved, and solutions with high fitness values were obtained. This shows that our technique is also useful for raising the convergence

rate and the optimization speed. Compared to the basic method $c = 0$, the convergence rate using our technique was increased about 5-fold and the number of generations for convergence was decreased by about 1.5-times in cases where the optimal coefficient was $c = 0.15$. The most suitable values of c in each application are presented in Table 3. Table 5 shows that if the value of c is too large, the convergence rate falls because all parameter values become 0. On the other hand, if the value is too small, the effect decreases and becomes equivalent to the basic method. Based on these considerations, we expect that a certain optimal value exists for c.

DISCUSSIONS

Not only the improvement of the evaluation function but also the introduction of the SPX were important factors in this work. By the crossover of the basic method, prediction was limited to 12 parameters for 2 variables (Tominaga *et al.*, 2000). However, introduction of the SPX expanded the search ability to 60 parameters for 5 variables. We previously tested the effects of the improvement of the evaluation function using the crossover of the basic method (Kikuchi *et al.*, 2001) and found that although the convergence rate improved, as was the case in the present study, it was difficult to increase the number of predictable parameters. Based on the data presented here, we conclude that adding SPX and the gradual optimization strategy contributed to the increase in the number of predictable parameters, and that the pruning method played a role in the observed improvement of the convergence rate and the optimization speed.

When the application of our method to actual biotechnology systems is considered, the acquisition of multiple time-courses becomes a problem in our strategy. In fact, we observed no convergence when only a few time-courses were given as input data, indicating that the restriction condition was too weak. One simple solution we chose here is to use multiple time-courses with the same dynamics and different initial concentrations.

The search performance can be expected to further improve, if the range of each parameter value is restricted by biological knowledge. Robustness, i.e. the ability to bear noises, was not addressed in this study. Since we generated the time-courses artificially, we did not consider the observation noise. However, we posit that decreasing the number of model parameters by the pruning method will be effective. As over-fitting worsens generalization ability, we will investigate the relationship between our technique and robustness. At present, we suggest that the number of sampling points and the optimization time are not unrealistic if our method is applied to real problems.

To analyze such perturbations in reality, quantitative analysis of biological dynamics will be required. Although

Table 4. Parameters estimated by PEACE1. The relative error of time-courses produced by this matrix is 0.54%

i	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}
1	5.9	0.0	0.0	0.9	0.0	−0.9	10.6	1.7	0.0	0.0	0.0	0.0
2	10.0	2.1	0.0	0.0	0.0	0.0	10.2	0.0	2.1	0.0	0.0	0.0
3	9.6	0.0	−0.9	0.0	0.0	0.0	9.7	0.0	−0.9	2.3	0.0	0.0
4	9.4	0.0	0.0	1.9	0.0	−0.9	11.5	0.0	0.0	0.0	1.8	0.0
5	10.2	0.0	0.0	0.0	2.1	0.0	10.2	0.0	0.0	0.7	0.0	1.9

Table 5. Changes in the convergence rate and the number of generations for convergence in attaining a 5% relative error at the first application according to structure parameter c. Here, since $c = 0$ is equivalent to not calculating the second term, this case reflects the basic method. $c = 0.15$ was a suitable value in this study

c	Success rate	No. of generations
0	**14%**	**9537**
0.03	38%	6619
0.09	58%	6488
0.15	**72%**	**6346**
0.21	60%	6927
0.27	66%	6439
0.33	54%	6742
0.39	70%	6520
0.45	66%	6757
0.51	64%	7166
0.57	44%	7574
0.63	36%	7789

conventional methods have aimed mainly at pathway-finding (D'haeseleer *et al.*, 2000), the future research focuses more on dynamic changes and interactions among biological objects. In this light, our work is considered a good primer for the reconstruction of complex dynamics. For example, our result can be imported to a kinetic cell simulation system, such as the E-CELL system (Tomita *et al.*, 1999), to see the network behavior under different perturbations. Although the ability of PEACE1 is still short for inferring the dynamics of a large network, the basic approach can be integrated with other systems to enlarge the network size to be predicted.

AIGNET (Algorithms for Inference of Genetic Networks; Maki *et al.*, 2001) is a system for large-scale genetic network modeling. It represents a genetic network as a directed graph where nodes and edges correspond to genes and their relations, respectively. The network prediction is based on the change in gene relations between normal (wild-type) and abnormal conditions, i.e. deletion or forcible expression of a particular gene. The gene relation for each disruption experiment is expressed by a matrix in which each element represents the real-value intensity of a gene expression. Since

AIGNET conforms to Boolean networks, the top of the loop cannot be determined, and loop structures cannot be inferred. Our PEACE1 can be applied for the modeling of loop structures as an additional component of AIGNET. AIGNET is useful for obtaining a rough skeletal structure and a large-scale network. PEACE1, on the other hand, can be used to infer loop structures and estimate detailed kinetics. We propose that in combination, PEACE1 and AIGNET can be used effectively for detailed large-scale network prediction such as TCA cycle.

CONCLUDING REMARKS

The basic method using the GA and the S-system can infer skeletal structures rarely and can predict only a very small number of parameters because it aims at reproducing given time-courses and employs a simple genetic operation strategy. We reported our improvement of an evaluation function that eliminates futile model parameters. Since it is aimed at a simple structure that is equal to an objective structure, it forces model parameters into desirable values. As a result, the convergence rate increased about 5-fold and the optimization speed increased about 1.5-fold. In addition, we employed a novel crossover SPX and a gradual optimization strategy. Using this method, we showed that the number of predictable parameters rose about 5-fold. We successfully and automatically inferred the dynamics of a small genetic network constructed with 60 parameters for 5 variables and the feedback loops from only time-course data of gene expression.

To achieve the goal of predicting large-scale networks, we propose that PEACE1 be included in a module of AIGNET, and be applied to actual genetic network prediction and kinetic cell simulation. Moreover, as the predicted dynamics are expressed as the S-system formalism, we suggest that steady-state evaluation, control analysis and sensitivity analysis be applied to the results thus obtained.

REFERENCES

Akutsu,T., Miyano,S. and Kuhara,S. (2000) Algorithms for identifying Boolean networks and related biological networks based on matrix multiplication and fingerprint function. *J. Comput. Biol.*, **7**, 331–343.

Alves,R. and Savageau,M.A. (2000) Systemic properties of ensembles of metabolic networks: application of graphical and statistical methods to simple unbranched pathways. *Bioinformatics*, **16**, 534–547.

Arkin,A., Ross,J. and McAdams,H.H. (1998) Stochastic kinetic analysis of developmental pathway bifurcation in phage λ-infected *Escherichia coli* cells. *Genetics*, **149**, 1633–1648.

D'haeseleer,P., Liang,S. and Somogyi,R. (2000) Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, **16**, 707–726.

Gilman,A. and Ross,J. (1995) Genetic-algorithm selection of a regulatory structure that directs flux in a simple metabolic model. *Biophys. J.*, **69**, 1321–1333.

Goldberg,D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA.

Higuchi,T., Tsutsui,S. and Yamamura,M. (2000) Theoretical analysis of simplex crossover for real-coded genetic algorithms. In *Proceedings of the International Conference on Parallel Problem Solving from Nature*. pp. 365–374.

Hlavacek,W.S. and Savageau,M.A. (1996) Rules for coupled expression of regulator and effector genes in inducible circuits. *J. Mol. Biol.*, **255**, 121–139.

Holter,N.S., Maritan,A., Cieplak,M., Fedoroff,N.V. and Banavar,J.R. (2001) Dynamic modeling of gene expression data. *Proc. Natl Acad. Sci. USA*, **98**, 1693–1698.

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.

Jong,H.D. (2002) Modeling and simulation of genetic regulatory systems: a literature review. *J. Comput. Biol.*, **9**, 67–103.

Jonikow,C.Z. and Michalewicz,Z. (1991) An experimental comparison of binary and floating point representations in genetic algorithms. In *Proceedings of the International Conference on Genetic Algorithms*. pp. 31–38.

Kikuchi,S., Tominaga,D., Arita,M. and Tomita,M. (2001) Genetic networks prediction using S-system optimized by GA. In *Proceedings of IPSJ meetings*. pp. MPS–37–4 (in Japanese).

Mackay,D.J.C. (1995) Probable networks and plausible prediction—a review of practical Bayesian methods for supervised neural networks. *Network: Computation in Neural Systems*, **6**, 469–505.

Maki,Y., Tominaga,D., Okamoto,M., Watanabe,S. and Eguchi,Y. (2001) Development of a system for the inference of large scale genetic networks. In *Proceedings of the Pacific Symposium on Biocomputing*. pp. 446–458.

Park,L.J., Park,C.H., Park,C. and Lee,T. (1997) Application of genetic algorithms to parameter estimation of bioprocesses. *Med. Biol. Eng. Comput.*, **35**, 47–49.

Pinchuk,J., Brown,W.A., Hughes,S.M. and Cooper,D.G. (2000) Modeling of biological process using self-cycling fermentation and genetic algorithm. *Biotechnol. Bioeng.*, **67**, 19–24.

Reinitz,J. and Sharp,D.H. (1995) Mechanism of *eve* stripe formation. *Mech. Dev.*, **49**, 133–158.

Savageau,M.A. (1976) *Biochemical System Analysis: a Study of Function and Design in Molecular Biology*. Addison-Wesley, Reading, MA.

Shiraishi,F. and Savageau,M.A. (1992) The tricarboxylic acid cycle in *Dictyostelium discoideum*. *J. Biol. Chem.*, **267**, 22912–22943.

Tominaga,D. and Okamoto,M. (1998) Design of canonical model complex nonlinear dynamics. In *Proceedings of the International Conference on Computer Applications in Biotechnology*. pp. 85–90.

Tominaga,D., Koga,N. and Okamoto,M. (2000) Efficient numerical optimization algorithm based on genetic algorithm for inverse problem. In *Proceedings of the Genetic and Evolutionary Computation Conference*. pp. 251–258.

Tomita,M., Hashimoto,K., Takahashi,K., Shimizu,T.S., Matsuzaki,Y., Miyoshi,F., Saito,K., Tanida,S., Yugi,K., Venter,J.C. and Hutchison,C.A. (1999) E-CELL: Software environment for whole-cell simulation. *Bioinformatics*, **15**, 72–84.

Tsutsui,S., Yamamura,M. and Higuchi,T. (1999) Multi-parent recombination with simplex crossover in real coded genetic algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference*. pp. 657–664.

Voit,E.O. (2000) *Computational Analysis of Biochemical Systems*. Cambridge University Press.

Voit,E.O. and Radivoyevitch,T. (2000) Biochemical systems analysis of genome-wide expression data. *Bioinformatics*, **16**, 1023–1037.

Williams,P.M. (1995) Bayesian regularization and pruning using a Laplace prior. *Neural Computat.*, **7**, 117–143.