BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 7 2005, pages 1154–1163
doi:10.1093/bioinformatics/bti071

Systems biology

Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm

Shuhei Kimura[1,3,*], Kaori Ide[1,3], Aiko Kashihara[3], Makoto Kano[5], Mariko Hatakeyama[1,3], Ryoji Masui[3,6], Noriko Nakagawa[3,7], Shigeyuki Yokoyama[2,3,4,6], Seiki Kuramitsu[3,7] and Akihiko Konagaya[1,3]

[1]Bioinformatics Group and [2]Protein Research Group, RIKEN Genomic Sciences Center, 1-7-22 Suehiro-cho, Tsurumi, Yokohama 230-0045, Japan, [3]Structurome Group and [4]Cellular Signaling Laboratory, RIKEN Harima Institute at Spring-8, 1-1-1 Kohto, Mikazuki-cho, Sayo, Hyogo 679-5148, Japan, [5]Tokyo Research Laboratory, IBM Japan, 1623-14 Shimo-tsuruma, Yamato, Kanagawa 242-8502, Japan, [6]Department of Biophysics and Biochemistry, Graduate School of Science, the University of Tokyo, 7-3-1 Hongo, Bunkyo, Tokyo 113-0033, Japan and [7]Department of Biology, Graduate School of Science, Osaka University, Toyonaka, Osaka 560-0043, Japan

Received on June 16, 2004; revised on September 1, 2004; accepted on September 18, 2004
Advance Access publication October 28, 2004

ABSTRACT

Motivation: To resolve the high-dimensionality of the genetic network inference problem in the S-system model, a problem decomposition strategy has been proposed. While this strategy certainly shows promise, it cannot provide a model readily applicable to the computational simulation of the genetic network when the given time-series data contain measurement noise. This is a significant limitation of the problem decomposition, given that our analysis and understanding of the genetic network depend on the computational simulation.

Results: We propose a new method for inferring S-system models of large-scale genetic networks. The proposed method is based on the problem decomposition strategy and a cooperative coevolutionary algorithm. As the subproblems divided by the problem decomposition strategy are solved simultaneously using the cooperative coevolutionary algorithm, the proposed method can be used to infer any S-system model ready for computational simulation. To verify the effectiveness of the proposed method, we apply it to two artificial genetic network inference problems. Finally, the proposed method is used to analyze the actual DNA microarray data.

Contact: skimura@gsc.riken.jp

Supplementary information: See Bioinformatics Online.

INTRODUCTION

Advancement in technologies such as DNA microarrays allows us to measure gene expression patterns on a genomic scale, but to exploit these technologies we must find ways to extract useful information from the massive amount of data (Kwon *et al*., 2003). Among the possible solutions for extracting information, many researchers have taken an interest in the inference of genetic networks. The inference of genetic networks is a problem in which mutual interactions among genes are estimated using time-series data of gene expression patterns. The inferred model of the genetic network is conceived as an ideal tool to help biologists generate hypotheses and facilitate the design of their experiments. On another level, it may also shed light on the biological functions of genes.

The numerous models proposed to describe biochemical networks have ranged from simple Boolean networks to detailed sets of differential equations of an arbitrary form (Akutsu *et al*., 2000; Chen *et al*., 1999; D'haeseleer *et al*., 2000; Maki *et al*., 2001; Sakamoto and Iba, 2001; Vance *et al*., 2002; Weaver *et al*., 1999). One of the well-studied models among them, the S-system, possesses a rich structure capable of capturing various dynamics, and can be analyzed by several available methods (Savageau, 1976; Voit and Radivoyevitch, 2000). These advantages have led to the successful application of the S-system model to the analysis of biochemical networks (e.g. Shiraishi and Savageau, 1992; Voit and Radivoyevitch, 2000). The model is a set of non-linear differential equations of the form

$$\frac{\mathrm{d}X_i}{\mathrm{d}t} = \alpha_i \prod_{j=1}^{N} X_j^{g_{i,j}} - \beta_i \prod_{j=1}^{N} X_j^{h_{i,j}} \quad (i = 1, \ldots, N), \qquad (1)$$

where X_i is the state variable and N is the number of components in the network. In a genetic network, X_i is the expression level of the i-th gene and N is the number of genes in the network. 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.

The genetic network inference problem based on the S-system model is defined as an estimation problem of the S-system parameters. Several algorithms for the inference of S-system models of genetic networks have been proposed (Kikuchi *et al*., 2003; Morishita *et al*., 2003; Tominaga *et al*., 2000; Ueda *et al*., 2002). These algorithms estimate the S-system parameters (α_i, β_i, $g_{i,j}$ and $h_{i,j}$) using observed time-series data of gene expression patterns. Because the number of S-system parameters is proportional to the square of the number of network components, the algorithms must simultaneously estimate a large number of S-system parameters if they are to be used to infer large-scale network systems containing many

*To whom correspondence should be addressed.

network components. This is why inference algorithms based on the S-system model have only been applied to small-scale networks of less than five genes.

To resolve the high-dimensionality of the genetic network inference problem in the S-system model, a problem decomposition strategy, that divides the original problem into several subproblems, has been proposed (Maki *et al*., 2002; Kimura *et al*., 2003). This approach enables us to infer S-system models of large-scale genetic networks. However, when the given time-series data contain measurement noise, the inferred model cannot be used to computationally simulate a genetic network. Given that we depend on computational simulation for our analysis and understanding of the genetic network, this is viewed as an important disadvantage of the problem decomposition approach. To overcome the high-dimensionality of the genetic network inference problem, Voit and Almeida (2004) have proposed another approach that transforms the problem into a set of algebraic equations. However, the same disadvantage as the problem decomposition strategy still remains in their approach.

In this paper, we propose a new method to overcome this disadvantage. The proposed method solves the decomposed subproblems simultaneously using a cooperative coevolutionary algorithm (Potter and De Jong, 2000). All of the subproblems in this coevolutionary algorithm interact with each other through time-courses of gene expression levels. With this interaction, the proposed method can be used to infer any S-system model ready for computational simulation. To verify the effectiveness of this method, we apply it to two artificial genetic network inference problems containing 5 and 30 genes, respectively. Finally, the proposed method is used to analyze the actual DNA microarray data.

GENETIC NETWORK INFERENCE PROBLEM

Canonical problem definition

In general, the genetic network inference problem is formulated as a function optimization problem to minimize the following sum of the squared relative error (e.g. see Tominaga *et al*., 2000).

$$f = \sum_{i=1}^{N} \sum_{t=1}^{T} \left(\frac{X_{i,\mathrm{cal},t} - X_{i,\mathrm{exp},t}}{X_{i,\mathrm{exp},t}} \right)^2, \tag{2}$$

where $X_{i,\mathrm{exp},t}$ is an experimentally observed gene expression level at time t of the i-th gene, $X_{i,\mathrm{cal},t}$ is a numerically computed gene expression level acquired by solving a system of differential equations (1), N is the number of components in the network and T is the number of sampling points of observed data.

Since $2N(N+1)$ S-system parameters must be determined in order to solve the set of differential equations (1), this function optimization problem is $2N(N+1)$-dimensional. This problem is too high-dimensional for non-linear function optimizers in cases where we try to infer S-system models of large-scale genetic networks containing many network components (Maki *et al*., 2001).

Decomposition of the problem

Because of the high-dimensionality, function optimizers have difficulty inferring S-system models of large-scale genetic networks. To resolve the high-dimensionality, Maki *et al*. (2002) proposed the strategy of dividing the genetic network inference problem into several subproblems. In this strategy, each subproblem corresponds to each gene. The objective function of the subproblem corresponding to the i-th gene is

$$f_i = \sum_{t=1}^{T} \left(\frac{X_{i,\mathrm{cal},t} - X_{i,\mathrm{exp},t}}{X_{i,\mathrm{exp},t}} \right)^2, \tag{3}$$

where $X_{i,\mathrm{cal},t}$ is a numerically computed gene expression level at time t of the i-th gene, as described in the previous subsection. In contrast to the previous subsection, however, $X_{i,\mathrm{cal},t}$ is obtained by solving the following differential equation:

$$\frac{\mathrm{d}X_i}{\mathrm{d}t} = \alpha_i \prod_{j=1}^{N} Y_j^{g_{i,j}} - \beta_i \prod_{j=1}^{N} Y_j^{h_{i,j}}, \tag{4}$$

where

$$Y_j = \begin{cases} X_j, & \text{if } j = i, \\ \hat{X}_j, & \text{otherwise.} \end{cases} \tag{5}$$

$\hat{X}_j$ is an estimated time-course of the j-th gene expression level acquired not by solving a differential equation, but by making a direct estimation from the observed time-series data. We can obtain $\hat{X}_j$s using an interpolation technique such as a spline interpolation (Press *et al*., 1995) or a local linear regression (Cleveland, 1979).

Equation (4) is solvable when $2(N+1)$ S-system parameters (i.e. $\alpha_i, \beta_i, g_{i,1}, \ldots, g_{i,N}, h_{i,1}, \ldots, h_{i,N}$) are given. Thus, the problem decomposition strategy divides a $2N(N+1)$-dimensional network inference problem into N subproblems that are $2(N+1)$-dimensional.

Use of a priori knowledge

The genetic network inference problem based on the S-system model may have multiple optima because the degree-of-freedom of the model is high and the observed time-series data are usually polluted by the measurement error. To increase the probability of inferring a correct S-system model, we introduced a priori knowledge of the genetic network into the objective function (Kimura *et al*., 2003).

Genetic networks are known to be sparsely connected (Thieffry *et al*., 1998). When an interaction between two genes is clearly absent, the S-system parameter values corresponding to the interaction (i.e. kinetic orders; $g_{i,j}$ and $h_{i,j}$) are zero. Kikuchi *et al*. (2003) incorporated this knowledge into the objective function using a penalty term named the pruning term. This turns out to be an imperfect solution, however, since the pruning term pushes all of the kinetic orders down to zero, a condition that may make prevent the model from finding the existing interactions. To avoid this, we incorporated the knowledge into the objective function (3) by using another penalty term, as shown below (Kimura *et al*., 2003).

$$F_i = \sum_{t=1}^{T} \left(\frac{X_{i,\mathrm{cal},t} - X_{i,\mathrm{exp},t}}{X_{i,\mathrm{exp},t}} \right)^2 + c \sum_{j=1}^{N-I} (|G_{i,j}| + |H_{i,j}|), \tag{6}$$

where $G_{i,j}$ and $H_{i,j}$ are given by rearranging $g_{i,j}$ and $h_{i,j}$, respectively, in descending order of their absolute values (i.e. $|G_{i,1}| \leq |G_{i,2}| \leq \cdots \leq |G_{i,N}|$ and $|H_{i,1}| \leq |H_{i,2}| \leq \cdots \leq |H_{i,N}|$). The variable c is a penalty coefficient and I is a maximum indegree. The maximum indegree determines the maximum number of genes that directly affect the i-th gene.

The penalty term is the second term on the right-hand side of Equation (6). This term forces most of the kinetic orders down to zero. In other words, when the penalty term is applied, most of the genes are disconnected from each other. However, when the number of genes that directly affect the i-th gene is smaller than the maximum indegree I, the term does not penalize. Thus, the optimum solutions to the objective functions (3) and (6) are identical when the number of interactions that affect the focused (i-th) gene is lower than the maximum indegree. In this paper, we use Equation (6) as an objective function that should be minimized.

PROPOSED METHOD

Concept

The problem decomposition strategy proposed by Maki *et al*. (2002) enables us to infer large-scale genetic networks. To solve the subproblems decomposed by this strategy, as mentioned above, the estimated time-courses of the gene expression levels, $\hat{X}_j$s, must be given. In the problem decomposition strategy, $\hat{X}_j$s are estimated directly from the observed time-series data using some interpolation method, and are not updated through the search. If $\hat{X}_j$s are correctly estimated, optimum solutions obtained from the problem decomposition approach and the canonical (non-decomposed) approach completely coincide with each other. However, when the given time-series data contain measurement noise, it is often difficult for us to estimate $\hat{X}_j$s correctly. When incorrect $\hat{X}_j$s are used, the optimum solutions of the decomposed subproblems do not always coincide with that of the non-decomposed problem. This means that the parameters obtained by solving the subproblems do not always provide us with a model [i.e. a set of differential equations (1)] that fits into the observed data. As such, in the problem decomposition approach, the inferred model is not yet suitable for the computational simulation of genetic networks.

In the subproblem corresponding to the i-th gene, the time-course of the i-th gene expression level is calculated by solving the differential equation (4). When optimizing the i-th subproblem, the function optimizer searches for the S-system parameters which make the calculated expression time-course of the i-th gene fits into the observed data. Therefore, the calculated time-courses obtained by solving the subproblems are the most suitable for $\hat{X}_j$s. If we can always use the calculated time-courses of the gene expression levels as $\hat{X}_j$s, optimizing the subproblems should give the model that fits into the observed data.

In order to use the time-courses of the gene expression levels obtained by solving the subproblems as $\hat{X}_j$s, we can use the cooperative coevolutionary approach (Liu *et al*., 2001; Potter and De Jong, 2000), an extension of the evolutionary algorithm (Holland, 1975). It consists of several subpopulations, each of which contains competing individuals (candidate solutions) for each subproblem. The subpopulations are genetically isolated, i.e. individuals mate only with other members of their subpopulation. These subpopulations interact with each other only when the fitness values (objective values) are calculated. In this paper, the subpopulations interact with each other only through the gene expression time-courses, i.e. when the proposed method solves the i-th subproblem, the calculated expression time-courses of the other genes, which are obtained from the best individuals of the other subproblems at the previous generation, are used as $\hat{X}_j$s (Fig. 1).

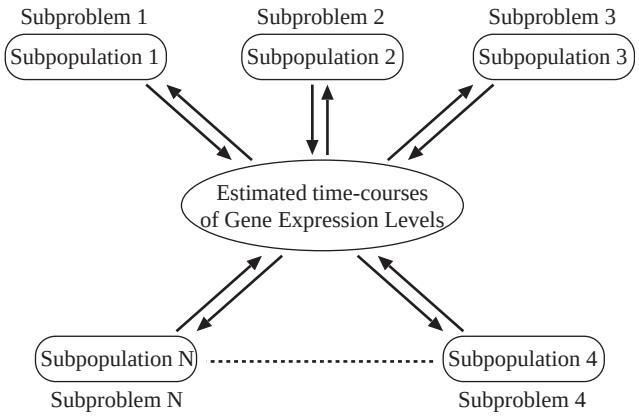

Fig. 1. The cooperative evolutionary model in this paper.

Algorithm

On the basis of the concept described above, we propose a new cooperative coevolutionary algorithm for inferring genetic networks. The following is an algorithm of the proposed method.

(1) *Initialize.* Generate N subpopulations, where N is the number of components in the genetic network. As an initial guess, estimate the gene expression time-courses from the observed time-series data. Set Generation $= 0$.

(2) *Execution of function optimization.* Execute one cycle of a function optimization algorithm on each subpopulation. In this paper, we use GLSDC (Kimura and Konagaya, 2003) as the function optimizer.

(3) *Update of estimated gene expression time-courses.* Update all of the estimated gene expression time-courses using the best individuals of the subpopulations. The updated gene expression time-courses are used as $\hat{X}_j$s in the next generation.

(4) Stop if halting criteria are satisfied. Otherwise, Generation $\leftarrow$ Generation $+1$ and go to Step 2.

Each of these steps is described below in greater detail.

Initialize N subpopulations, each of which corresponds to one subproblem, are generated. Each subpopulation contains n_p individuals which are randomly created. At the same time, initial estimations of time-courses of gene expression levels, $\hat{X}_j$s, are made directly from the observed time-series data. In this paper, the local linear regression (Cleveland, 1979) is used to estimate the time-courses.

Execution of function optimization Any type of function optimizer can be applied to the decomposed subproblem. In this study, we decided to adopt GLSDC, an evolutionary algorithm successfully applied to the genetic network inference problem as a function optimizer by Kimura *et al*. (2003, 2004).

In this step, one cycle (generation) of GLSDC is performed on each subpopulation. When the algorithm calculates the fitness value of each individual in each subpopulation, the differential equation (4) is solved using the estimated time-courses of the gene expression levels, $\hat{X}_j$s . An initial gene expression level (an initial state value for the differential equation) is required together with the S-system parameters at this time. In this study, the initial gene expression level

Table 1. S-system parameters of the small-scale target model

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
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

of the i-th gene was obtained from its estimated gene expression time-course, i.e. the value of $\hat{X}_i(0)$ was used for $X_{i,\mathrm{cal},0}$.

Update of estimated gene expression time-courses Next, we calculate the time-courses of the gene expression levels obtained from the best individuals of the subpopulations, each of which is given as a solution of the differential equation (4). The old gene expression time-courses are then updated to these calculated time-courses. The updated gene expression time-courses are used as $\hat{X}_j$s in the next generation.

When we calculate the time-courses of the gene expression levels, the initial levels of the gene expression are required. Since the noise in the actual time-series data corrupts the values of the initial gene expression levels, we should estimate these values together with the S-system parameters. However, the simultaneous estimation of the initial gene expression levels and S-system parameters increases the dimensionality of the function optimization problem, creating a condition inconvenient for function optimizers. To avoid this problem, we use an alternate method for estimation (Kimura *et al*., 2004).

In this step, the initial levels of the gene expression are adjusted to fit the new calculated gene expression time-courses into the observed time-series data, before the gene expression time-courses are updated. The adjustment of the initial gene expression level of the i-th gene is formulated as a one-dimensional function minimization problem. This is because the initial gene expression level of the i-th gene is a unique variable and all of the S-system parameters are fixed to the values of the best individual. The objective function of this adjustment problem is

$$F_i^{\mathrm{adj}} = \sum_{t=1}^{T} \gamma^{t-1} \left(\frac{X_{i,\mathrm{cal},t} - X_{i,\mathrm{exp},t}}{X_{i,\mathrm{exp},t}} \right)^2, \qquad (7)$$

where $X_{i,\mathrm{cal},t}$ is acquired by solving the differential equation (4) and $\gamma (0 \leq \gamma \leq 1)$ is a discount parameter. Since the fixed model parameters obtained from the best individual are not always optimal, the calculated gene expression time-course may differ greatly from the actual time-course. When the estimated time-course is incorrect, the algorithm should not fit the time-course, especially the latter half of it, into the observed data. Therefore, in this study, we introduce a discount parameter γ.

A golden section search (Press *et al*., 1995) is used to solve the one-dimensional function minimization problem described above. When multiple sets of time-series data are given as the observed data, the one-dimensional search is applied to all of the sets. After the adjustment, the new calculated gene expression time-courses are substituted for the old ones, and they are used as $\hat{X}_j$s in the next generation.

NUMERICAL EXPERIMENTS

To show the effectiveness of the proposed method, we applied it to two artificial genetic network inference problems. Then, it was used to analyze the actual DNA microarray data.

Experiment 1: noise-free environment

In this experiment, we confirm that the proposed method has an ability to infer a correct S-system model of the genetic network when a sufficient amount of noise-free data is given.

Experimental setup As a target genetic network, we used a small-scale S-system model with the parameters listed in Table 1 (Kikuchi *et al*., 2003). This model consists of five network components ($N = 5$).

If an insufficient amount of time-series data is given as observed gene expression patterns, the high degree-of-freedom of S-system models ensures that many candidate solutions will be found. In this experiment, however, we used a sufficient amount of time-series data to enhance our chances of finding the correct solution. Specifically, we used 15 sets of noise-free time-series data, each covering all five genes. The sets of time-series data were obtained by solving the set of differential equations (1) on the target model. The initial values of these sets were generated randomly (listed in Table 2). In a practical application, these sets of time-series data could be obtained by actual biological experiments under different experimental conditions. A total of 11 sampling points for the time-series data were assigned on each gene in each set. Thus, the observed time-series data for each gene consisted of $15 \times 11 = 165$ sampling points. In this experiment, $2 \times 5 \times (5 + 1) = 60$ S-system parameters and $15 \times 5 = 75$ levels of initial gene expression should be estimated.

As the proposed method is based on the stochastic search algorithm, we should perform multiple runs by changing the seed for pseudo random number in order to confirm its performance. Therefore, five runs were carried out. In each run, the proposed method produces one candidate solution. Each run was continued until the number of generations reached 75. The search regions of the parameters were [0.0, 20.0] for α_i and β_i, and [−3.0, 3.0] for $g_{i,j}$ and $h_{i,j}$. As the observed initial gene expression levels should be close to true ones even when they are polluted by measurement error, the search regions of the initial gene expression levels were set to ±30% of the observed ones (i.e. $[0.7X_{i,\mathrm{exp},0}, 1.3X_{i,\mathrm{exp},0}]$). The maximum indegree I was 5, the penalty coefficient c was 1.0, and the discount parameter γ was 0.75. In this paper, we used the following GLSDC parameters; the population size n_p is $3n$, where n is the dimension of the search space of each subproblem; the number of children generated by the crossover per selection n_c is 10; and the number of applied

Table 2. Fifteen sets of the initial gene expression levels used in the experiment with the small-scale target model

Set	X_1	X_2	X_3	X_4	X_5
1	1.655967E+00	1.868416E+00	1.032173E−01	2.730268E−01	1.562687E+00
2	7.862766E−01	5.474855E−01	9.287958E−01	3.894443E−01	9.344040E−01
3	3.468547E−01	1.994981E+00	1.532913E+00	1.761393E+00	1.264981E+00
4	8.020131E−01	8.949262E−01	3.135082E−01	7.610533E−02	1.269706E+00
5	9.590725E−01	2.805737E−01	5.507401E−01	1.694232E+00	5.744767E−01
6	3.992936E−01	1.849408E+00	2.912736E−01	1.144217E+00	9.988814E−01
7	1.055713E−02	5.114093E−02	8.495855E−01	1.740444E+00	1.969969E−01
8	1.489803E+00	9.168820E−01	1.707836E+00	1.827741E+00	2.824051E−01
9	1.842769E−01	1.589055E+00	6.668454E−01	4.727903E−01	1.265678E+00
10	1.285646E+00	8.995862E−01	1.994967E−01	8.811659E−01	1.723054E+00
11	1.336863E−01	4.233753E−01	4.168260E−01	4.823942E−01	5.539923E−01
12	1.652500E+00	1.744966E+00	3.904404E−01	1.584671E+00	4.339247E−01
13	1.562800E+00	1.164151E+00	1.391469E+00	6.808265E−01	1.090292E+00
14	3.271505E−01	1.147837E+00	1.576167E−01	8.645541E−01	2.591408E−01
15	5.522177E−01	4.220327E−01	1.084436E+00	1.994388E+00	1.050098E+00

Table 3. A sample of estimated S-system parameters in the experiment with the small-scale target model

i	α_i	$g_{i,1}$	$g_{i,2}$	$g_{i,3}$	$g_{i,4}$	$g_{i,5}$	β_i	$h_{i,1}$	$h_{i,2}$	$h_{i,3}$	$h_{i,4}$	$h_{i,5}$
1	4.917	−0.009	−0.003	1.019	−0.017	−1.014	9.922	2.021	−0.009	0.002	−0.009	−0.009
2	10.030	1.995	0.002	−0.002	0.006	−0.001	10.026	0.002	1.995	−0.002	0.002	0.000
3	9.851	−0.005	−0.991	−0.004	−0.003	0.002	9.835	−0.004	−0.993	2.036	−0.010	0.002
4	8.020	−0.007	0.006	2.000	−0.002	−0.998	10.054	0.001	0.003	0.008	1.988	0.007
5	9.875	−0.002	0.003	0.018	2.015	−0.020	9.892	0.004	0.002	0.008	−0.010	2.017

Table 4. A sample of estimated initial gene expression levels in the experiment with the small-scale target model

Set	X_1	X_2	X_3	X_4	X_5
1	1.656888E+00	1.868827E+00	1.031426E−01	2.727441E−01	1.563031E+00
2	7.863679E−01	5.474571E−01	9.291561E−01	3.898476E−01	9.349235E−01
3	3.468950E−01	1.995085E+00	1.532417E+00	1.760597E+00	1.264096E+00
4	8.021380E−01	8.953308E−01	3.134594E−01	7.608557E−02	1.270594E+00
5	9.604875E−01	2.802652E−01	5.510681E−01	1.693792E+00	5.739783E−01
6	3.992472E−01	1.850007E+00	2.912282E−01	1.143535E+00	9.994764E−01
7	1.055016E−02	5.123888E−02	8.495751E−01	1.740938E+00	1.969536E−01
8	1.489976E+00	9.178903E−01	1.709318E+00	1.825659E+00	2.825914E−01
9	1.841744E−01	1.588337E+00	6.678721E−01	4.723989E−01	1.265056E+00
10	1.284448E+00	8.998418E−01	1.996619E−01	8.810286E−01	1.723033E+00
11	1.336120E−01	4.231231E−01	4.167611E−01	4.827125E−01	5.529668E−01
12	1.651859E+00	1.743927E+00	3.905919E−01	1.582865E+00	4.336105E−01
13	1.563669E+00	1.163820E+00	1.392009E+00	6.809762E−01	1.090532E+00
14	3.271675E−01	1.148089E+00	1.576677E−01	8.633683E−01	2.593822E−01
15	5.524893E−01	4.221964E−01	1.083833E+00	1.992993E+00	1.049187E+00

the converging operations N_0 is n_p. The experiments were executed in parallel on a PC cluster (Pentium III 933 MHz × 8 CPUs).

In order to reduce the computational cost, we applied a structure skeletalizing technique (Tominaga *et al*., 2000). This technique assigns a value of zero to the kinetic orders ($g_{i,j}$ and $h_{i,j}$), whose absolute values are less than the given threshold δ_s. Structure skeletalizing reduces the computational cost because the exponential calculation of Equation (4) can be omitted when the kinetic orders are zero. In this paper, the given threshold δ_s was 1.0×10^{-3}.

Result Tables 3 and 4 show the samples of the S-system parameters and the initial gene expression levels, respectively, estimated by

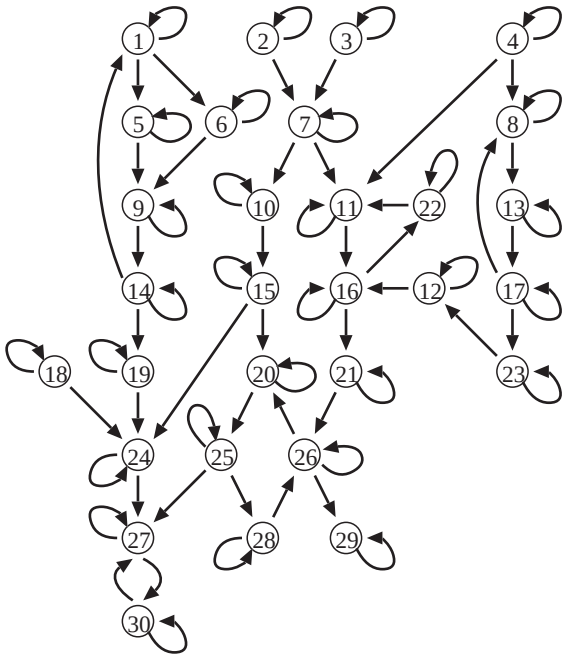

Fig. 2. The target genetic network used in Experiment 2.

the proposed method. As the tables show, our method was unable to estimate the parameter values with perfect precision. Notwithstanding, the values were precise enough to biologically interpret the network. The sum of the squared relative error between the time-courses produced by the inferred model and the given time-series data, i.e. the value of the function (2), averaged about $2.08 \times 10^{-3} \pm 0.77 \times 10^{-3}$.

In this experiment, we confirmed the effectiveness of the proposed method by estimating both the initial gene expression levels and the S-system parameters. In practice, however, there is no need to estimate the initial levels of the gene expression when the observed data seem to contain no measurement error. When the initial gene expression levels do not need to be estimated, the estimated parameters will be more precise since the problem contains fewer unknown parameters.

Our method running on the PC cluster (Pentium III 933 MHz × 8 CPUs) required ∼89.0 min to solve this problem. This is far less computing time than that required by Predictor by Evolutionary Algorithms and Canonical Equations 1 (PEACE1) proposed by Kikuchi *et al*. (2003). PEACE1 running on a PC cluster (Pentium III 933 MHz × 1040 CPUs) reportedly took more than 10 h to estimate the S-system parameters.

Experiment 2: noisy environment

Next, we test the performance of our method in a real-world setting by conducting the experiment with the sets of noisy time-series data.

Experimental setup A large-scale S-system model containing 30 genes ($N = 30$) was used as a target model. The network structure and the S-system parameters of the model are given in Figure 2 and Table 5, respectively (Maki *et al*., 2001). The observed gene expression data included 20 sets of time-series data, each covering all 30 genes. The sets of time-series data began from randomly

Table 5. S-system parameters of the large-scale target model

α_i	1.0
β_i	1.0
$g_{i,j}$	$g_{1,14} = -0.1$, $g_{5,1} = 1.0$, $g_{6,1} = 1.0$, $g_{7,2} = 0.5$, $g_{7,3} = 0.4$, $g_{8,4} = 0.2$, $g_{8,17} = -0.2$, $g_{9,5} = 1.0$, $g_{9,6} = -0.1$, $g_{10,7} = 0.3$, $g_{11,4} = 0.4$, $g_{11,7} = -0.2$, $g_{11,22} = 0.4$, $g_{12,23} = 0.1$, $g_{13,8} = 0.6$, $g_{14,9} = 1.0$, $g_{15,10} = 0.2$, $g_{16,11} = 0.5$, $g_{16,12} = -0.2$, $g_{17,13} = 0.5$, $g_{19,14} = 0.1$, $g_{20,15} = 0.7$, $g_{20,26} = 0.3$, $g_{21,16} = 0.6$, $g_{22,16} = 0.5$, $g_{23,17} = 0.2$, $g_{24,15} = -0.2$, $g_{24,18} = -0.1$, $g_{24,19} = 0.3$, $g_{25,20} = 0.4$, $g_{26,21} = -0.2$, $g_{26,28} = 0.1$, $g_{27,24} = 0.6$, $g_{27,25} = 0.3$, $g_{27,30} = -0.2$, $g_{28,25} = 0.5$, $g_{29,26} = 0.4$, $g_{30,27} = 0.6$, other $g_{i,j} = 0.0$
$h_{i,j}$	1.0 if $i = j$, 0.0 otherwise

generated initial values in $[0.0, 2.0]$ and were obtained by solving the set of differential equations (1) on the target model. We added 10% Gaussian noise to the time-series data in order to simulate the measurement noise that often corrupts the observed data obtained from actual measurements of gene expression patterns. A total of 11 sampling points for the time-series data were assigned on each gene in each set. In this experiment, $2 \times 30 \times (30 + 1) + 30 \times 20 = 2460$ parameters should be estimated.

Five runs were carried out. As the performance of the algorithm seems to depend on the given data, different sets of time-series data, generated randomly, were used in each run. The search regions were $[0.0, 3.0]$ for α_i and β_i, $[-3.0, 3.0]$ for $g_{i,j}$ and $h_{i,j}$, and $[0.7X_{i,\mathrm{exp},0}, 1.3X_{i,\mathrm{exp},0}]$ for the initial levels of the gene expression. The experiments were executed in parallel on a PC cluster (Pentium III 933 MHz × 32 CPUs). All of the other experimental conditions were the same as those used in the experiment conducted in the noise-free environment described above.

To confirm the effectiveness of the coevolutionary approach, we compared the results to those of a non-coevolutionary method that did not consider the interactions between decomposed subproblems (Kimura *et al*., 2004). This paper refers to this non-coevolutionary method as the problem decomposition approach.

Result Figure 3 shows the calculated gene expression time-courses obtained from the methods with and without the coevolution. The calculated time-courses obtained by solving the set of Equations (1) and (4), respectively, are shown in the figure. As shown in Figure 3A, when the proposed coevolutionary approach was applied, the time-course obtained by solving the set of equations (1) was almost identical to that obtained by solving Equation (4). On the contrary, the calculated time-courses of the problem decomposition approach differed greatly (Fig. 3B).

When inferring S-system models of genetic networks, both approaches use the differential equation (4) to calculate time-courses of gene expression levels. In Equation (4), however, the perturbation in the i-th gene does not affect the expression levels of the other genes. Therefore, Equation (4) is not a suitable model to help biologists generate hypotheses and facilitate the design of their experiments, while it is a useful model for inferring genetic networks. When we analyze the inferred genetic network, the set of equations (1) must be used as the model for computational simulation. From this point of view, the problem decomposition approach does not produce a suitable model for computational simulation, since the model does not always fit into

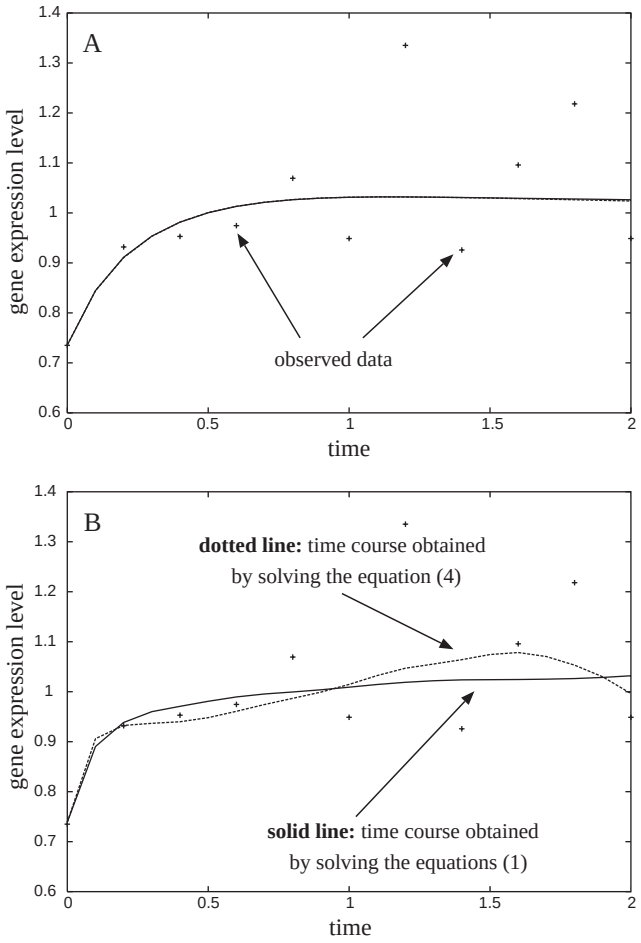

Fig. 3. Samples of calculated time-courses obtained from (**A**) the proposed coevolutionary approach and (**B**) the problem decomposition approach. Solid line: the solution of the set of differential equations (1) where the estimated values are used as the model parameters. Dotted line: time-course obtained at the end of the search, i.e. the solution of the differential equation (4). Plus symbol: noisy time-series data given as the observed data.

the observed data. As the time-courses obtained from Equation (1) are almost identical to those obtained from Equation (4), the proposed approach provides us with a suitable model. The sum of the squared relative error between the given data and the calculated time-courses of the model inferred by the proposed method was always smaller than that obtained from the problem decomposition approach in this experiment. The sums of squared relative error obtained from the methods with and without the coevolution averaged about 27.72 $\pm$ 0.68 and 28.18 $\pm$ 0.76, respectively.

Typical results obtained from the methods with and without the coevolution are shown in Figures 4 and 5, respectively. Inferred interactions for the 8th, 16th and 24th subproblems are shown. As the results show, both methods failed to infer some of the interactions present in the target model, and they inferred many erroneous interactions that had absolute parameter values too large to ignore. Weakly interactions were, especially, difficult to be correctly inferred, e.g. both methods often failed to infer interactions corresponding to $g_{1,14}$, $g_{24,18}$ and $g_{26,28}$. In addition, an interaction represented as $g_{i,j}$ was sometimes inferred as that of $h_{i,j}$. The failure to infer the correct interactions, however, does not seriously hinder our investigation, as the inferred model is intended mainly for use by biologists as a tool for generating hypotheses and facilitating the design of experiments. The necessary interactions that were not correctly inferred should be added, and the erroneous interactions should be removed in either of two ways, i.e. by using more sets of time-series data obtained from additional biological experiments or by using further a priori knowledge about the genetic network.

The model inferred by the proposed method contained 58.4 ± 2.1 true-positive interactions and 241.6 ± 2.1 false-positive interactions on average. The number of the inferred interactions corresponded to the maximum indegree I. Our method failed to infer an average of 9.6 ± 2.1 interactions that were present in the target model (i.e. the number of false-negative interactions was 9.6 ± 2.1). On the contrary, in the experiment using the problem decomposition approach, the numbers of true-positive false-positive and false-negative interactions averaged 57.6 ± 2.3, 242.4 ± 2.3 and 10.4 ± 2.3, respectively. To solve this problem, the proposed coevolutionary method required $\sim$57.8 h on the PC cluster (Pentium III 933 MHz $\times$ 32 CPUs). The computational time that the problem decomposition approach required for optimizing each subproblem averaged $\sim$57.5 h on a single-CPU personal computer (Pentium III 933 MHz), and the subproblems were optimized simultaneously on the PC cluster.

The experimental results suggest that our coevolutionary approach slightly improves the probability of inferring the correct interactions. In order to confirm the improvement, we performed a number of other experiments using different amount of time-series data. Figure 6 shows the number of false-negative interactions in each. In all of the experiments, the proposed method slightly reduced the number of false-negative interactions, i.e. it enhanced the probability of finding the correct interactions. This may be because the proposed method updates the estimated gene expression time-courses, $\hat{X}_j$s. In this study, the algorithm uses $\hat{X}_j$s to solve the decomposed subproblems. Therefore, $\hat{X}_j$s must be precise if the probability of finding the correct interactions is to be improved. Because the proposed coevolutionary approach updates $\hat{X}_j$s, their precision may be improved through searches.

In the proposed coevolutionary method, the improvement in a performance of finding correct interactions was slight. However, it should be noted that the same time-series data were given to both methods as the observed ones. As the proposed method extracts correcter information from the data, the inferred model is more reasonable to analyze genetic networks.

Experiment 3: analysis of actual data

The proposed method enables us to infer large-scale genetic networks containing dozens of genes. However, when attempting to analyze actual DNA microarray data, many hundreds or thousands of genes must be handled. This task lies far beyond the powers of the proposed coevolutionary method. One possible strategy to improve its inference capability is to use any clustering technique to identify genes with similar expression patterns and group them together (D'haeseleer *et al*., 2000; Eisen *et al*., 1998). By treating groups of similar genes as single-network components, the proposed coevolutionary method is capable of analyzing systems containing many hundreds of genes. In this study, we combined the proposed

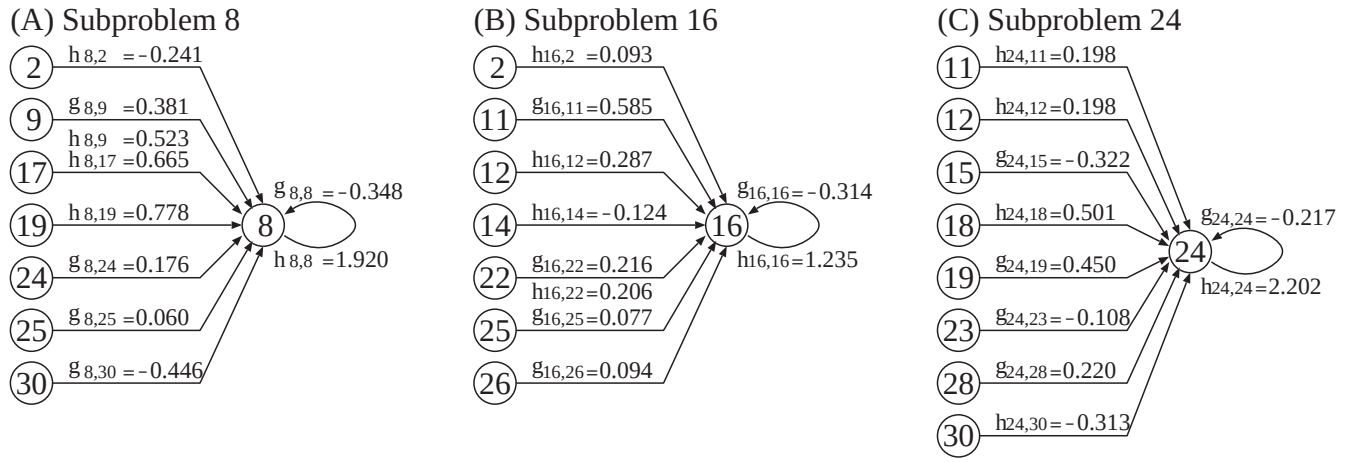

Fig. 4. Samples of the interactions inferred by the proposed method. The figure shows the results for (**A**) the 8th subproblem, (**B**) the 16th subproblem and (**C**) the 24th subproblem.

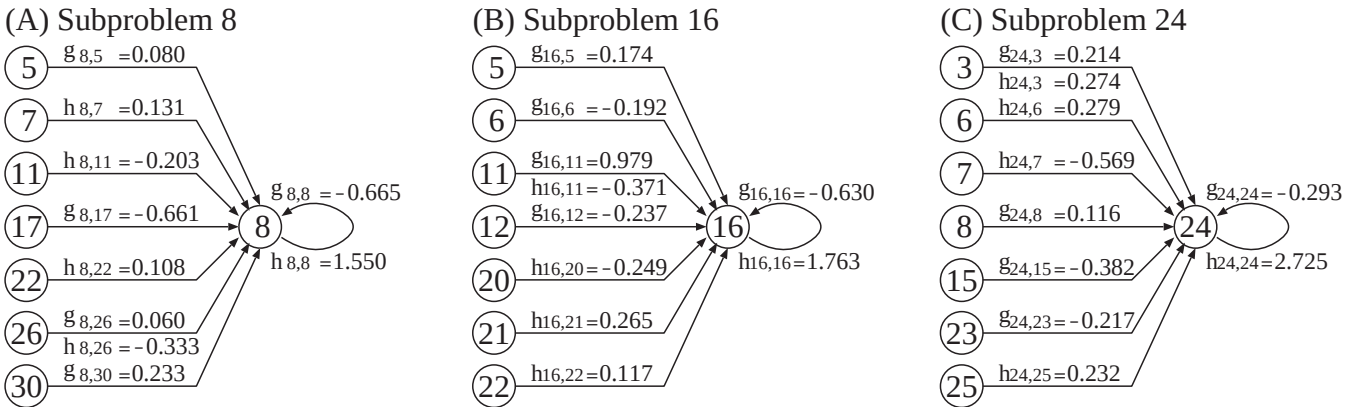

Fig. 5. Samples of the interactions inferred by the problem decomposition approach.

method with the clustering technique proposed by Kano *et al*. (2003). The combined method was then used to analyze cDNA microarray data of *Thermus thermophilus* HB8 strains.

Two sets of cDNA microarray time-series data, i.e. wild-type and *UvrA* gene disruptant, were observed. Each sets of the data were measured at 14 time points. The clustering technique used in this study grouped 612 putative open reading frames (ORFs) included in the data into 24 clusters. As we treated the disrupted gene, *UvrA*, as single-network component, the target system consisted of $24 + 1 = 25$ network components. The time-series data of each cluster was given by averaging the expression patterns of ORFs included in the cluster. A total of 10 runs were carried out. The maximum indegree I was 3. The search regions were $[0.0, 5.0]$ for α_i and β_i, $[-3.0, 3.0]$ for $g_{i,j}$ and $h_{i,j}$, and $[0.7X_{i,\mathrm{exp},0}, 1.3X_{i,\mathrm{exp},0}]$ for the initial levels of the gene expression. The experiments were executed in parallel on a PC cluster (Pentium III 933 MHz $\times$ 32 CPUs). All of the other experimental conditions were the same as those in previous experiments.

Figure 7 shows the core network structure where the interactions were inferred by the proposed method more than nine times within 10 runs. As the amount of observed data was insufficient, the inferred network model seems to contain many erroneous interactions. However, some reasonable interactions were also inferred. Many ORFs contained in the clusters 6, 7, 10, 15, 16, 19 and 22 are annotated to be concerned with 'Energy metabolism', and these clusters were relatively located near from each other in the inferred model. The figure shows the clusters 12 and 23 were also located near from the clusters of 'Energy metabolism'. However, a few ORFs contained in the clusters 12 and 23 are annotated to be concerned with 'Energy metabolism'. This fact suggests that some of hypothetical and unknown ORFs included in the clusters 12 and 23 may work for 'Energy metabolism' or related functions.

In this experiment, as the amount of the measured time-series data was insufficient, it is hard to extract many suggestions from the inferred network. To obtain more meaningful results, we are now planning additional biological experiments.

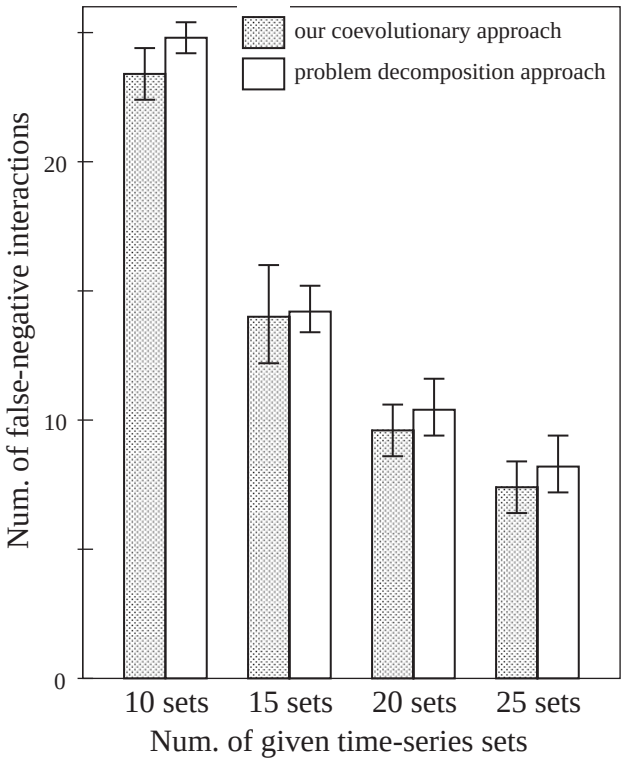

Fig. 6. The number of false-negative interactions in experiments using different amount of time-series data.

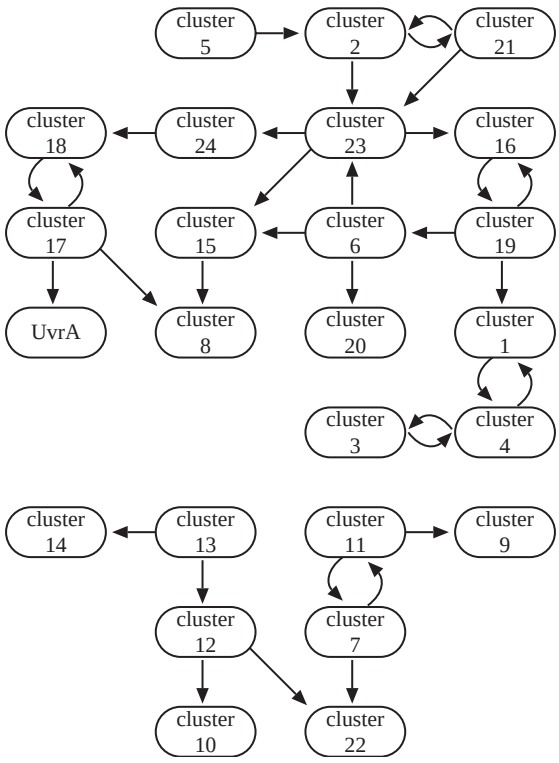

Fig. 7. The inferred genetic network.

CONCLUSION

In this paper, we proposed a new method for inferring the S-system models of large-scale genetic networks. The proposed method uses the problem decomposition strategy to divide the genetic network inference problem into several subproblems. The decomposed subproblems are then solved simultaneously using the cooperative coevolutionary algorithm. Because the decomposed subproblems interact with each other through their calculated gene expression time-courses, the inferred model can be used in the computational simulation. This feature is important because the computational simulation provides us with a better understanding of genetic networks. Through numerical experiments, we showed that the proposed method slightly enhanced the probability of finding the correct interactions of a network. Updating the gene expression time-courses also seems to enhance the probability of inferring a correct network structure. Finally, to analyze actual DNA microarray data, we combined the proposed coevolutionary method with the clustering technique.

ACKNOWLEDGEMENTS

We thank Dr S. Kuhara and Dr K. Tashiro in Kyushu University for supervising cDNA microarray experiments.

REFERENCES

Akutsu,T., Miyano,S. and Kuhara,S. (2000) Inferring qualitative relations in genetic networks and metabolic pathways. *Bioinformatics*, **16**, 727–734.

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

Cleveland,W.S. (1979) Robust locally weight regression and smoothing scatterplots. *J. Am. Stat. Assoc.*, **79**, 829–836.

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

Eisen,M.B., Spellman,P.T., Brown,P.O. and Botstein D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc. Natl Acad. Sci., USA*, **95**, 14863–14868.

Holland,J.H. (1975) *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, Ann Arbor, MI.

Kano,M., Nishimura,K., Tsutsumi,S., Aburatani,H., Hirota,K. and Hirose,M. (2003) Cluster overlap distribution map: visualization for gene expression analysis using immersive projection technology. *Presence Teleoper. Virt. Environ.*, **12**, 96–109.

Kikuchi,S., Tominaga,D., Arita,M., Takahashi,K. and Tomita,M. (2003) Dynamic modeling of genetic networks using genetic algorithm and S-system. *Bioinformatics*, **19**, 643–650.

Kimura,S. and Konagaya,A. (2003) High dimensional function optimization using a new genetic local search suitable for parallel computers. In *Proceedings of the 2003 Conference on Systems, Man and Cybernetics*, Washington, DC, USA, pp. 335–342.

Kimura,S., Hatakeyama,M. and Konagaya,A. (2003) Inference of S-system models of genetic networks using a genetic local search. In *Proceedings of the 2003 Congress on Evolutionary Computation*, Canberra, Australia, pp. 631–638.

Kimura,S., Hatakeyama,M. and Konagaya,A. (2004) Inference of S-system models of genetic networks from noisy time-series data. *Chem-Bio Informatics J.*, **4**, 1–14.

Kwon,A.T., Hoos,H.H. and Ng,R. (2003) Inference of transcriptional regulation relationships from gene expression data. *Bioinformatics*, **19**, 905–912.

Liu,Y., Yao,X., Zhao,Q. and Higuchi,T. (2001) Scaling up fast evolutionary programming with cooperative coevolution. In *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, Korea, pp. 1101–1108.

Maki,Y., Tominaga,D., Okamoto,M., Watanabe,S. and Eguchi,Y. (2001) Development of a system for the inference of large scale genetic networks. *Proc. Pac. Symp. Biocomput.*, **6**, 446–458.

Maki,Y., Ueda,T., Okamoto,M., Uematsu,N., Inamura,Y. and Eguchi,Y. (2002) Inference of genetic network using the expression profile time course data of mouse P19 cells. *Genome Inform.*, **13**, 382–383.

Morishita,R., Imade,H., Ono,I., Ono,N. and Okamoto,M. (2003) Finding multiple solutions based on an evolutionary algorithm for inference of genetic networks by S-system. In *Proceedings of the 2003 Congress on Evolutionary Computation*, Canberra, Australia, pp. 615–622.

Potter,M.A. and De Jong,K.A. (2000) Cooperative coevolution: an architecture for evolving coadapted subcomponents. *Evol. Comput.*, **8**, 1–29.

Press,W., Teukolsky,S., Vetterling,W. and Flannery,B. (1995) *Numerical Recipes in C*, 2nd edn. Cambridge University Press, Cambridge, UK.

Sakamoto,E. and Iba,H. (2001) Inferring a system of differential equations for a gene regulatory network by using genetic programming. In *Proceedings of the 2001 Congress on Evolutionary Computation*, Seoul, Korea, pp. 720–726.

Savageau,M.A. (1976) *Biochemical Systems 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**, 22912–22918.

Thieffry,D., Huerta,A.M., Pérez-Rueda,E. and Collado-Vides,J. (1998) From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in *Escherichia coli*. *BioEssays*, **20**, 433–440.

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*, Las Vegas, Nevada, USA, pp. 251–258.

Ueda,T., Ono,I. and Okamoto,M. (2002) Development of system identification technique based on real-coded genetic algorithm. *Genome Inform.*, **13**, 386–387.

Vance,W., Arkin,A. and Ross,J. (2002) Determination of causal connectivities of species in reaction networks. *Proc. Natl Acad. Sci., USA*, **99**, 5816–5821.

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

Voit,E.O. and Almeida,J. (2004) Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics*, **20**, 1670–1681.

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

Weaver,D.C., Workman,C.T. and Stormo,G.D. (1999) Modeling regulatory networks with weight matrices. *Proc. Pac. Symp. Biocomput.*, **4**, 112–123.