Chem-Bio Informatics Journal, Vol. 11, pp.24-40 (2011)

Efficient Parameter Estimation for the Inference of S-system Models of Genetic Networks: Proposition of Further Problem Decomposition and Alternate Function Optimization

Shuhei Kimura[1*], Koki Matsumura[1] and Mariko Okada-Hatakeyama[2]

[1]*Graduate School of Engineering, Tottori University, 4-101, Koyama-minami, Tottori 680-8552, Japan*
[2]*Research Center for Allergy and Immunology, RIKEN, 1-7-22, Suehiro, Tsurumi, Yokohama 230-0045, Japan*

**E-mail: kimura@ike.tottori-u.ac.jp*

(Received November 29, 2010; accepted February 7, 2011; published online February 25, 2011)

Abstract

The problem decomposition strategy is a very efficient technique for the inference of S-system models of genetic networks. This strategy defines the inference of a genetic network consisting of *N* genes as *N* subproblems, each of which is a 2(*N*+1)-dimensional function optimization problem. Genetic networks made up of dozens genes can be analyzed with this strategy, though the computational cost in doing so remains quite high. In this study, we attempt to infer S-system models more efficiently by further dividing each 2(*N*+1)-dimensional subproblem into one (*N*+2)-dimensional problem and one (*N*+1)-dimensional problem. The subproblems are divided using the genetic network inference method based on linear programming machines (LPMs). Next, we propose a new method for estimating the S-system parameters by alternately solving the two divided problems. According to our experimental results, the proposed approach requires less than one-third of the time required by the original problem decomposition approach. Finally, we apply our approach to actual expression data from the bacterial SOS DNA repair system.

Key Words: Genetic network, S-system, Problem decomposition, Parameter estimation

Area of Interest: Bioinformatics and Bio Computing

1. Introduction

A genetic network is a functioning circuitry in living cells at the gene level. It can be considered as an abstract mapping of an actual biochemical network consisting of a number of components, such as genes, proteins, metabolites, and so on [11]. By analyzing genetic networks, investigators can gain insight into the working of biological systems. To take advantage of the possibilities, several researchers have closely studied techniques for inferring genetic networks. Numerous

models have thus been proposed to describe genetic networks, and numerous algorithms based on individual models have been proposed to infer genetic networks from observed time-series of gene expression levels (see, e.g., [1][6][11][30][31]).

The S-system model holds great promise as a framework for describing genetic networks, since this model possesses a rich structure capable of capturing various dynamics and can be analyzed by several available methods [28]. The model is a set of non-linear differential equations of the form

$$\frac{dX_n}{dt} = \alpha_n \prod_{m=1}^{N} X_m^{g_{n,m}} - \beta_n \prod_{m=1}^{N} X_m^{h_{n,m}}, \qquad (n = 1,2,\cdots,N), \tag{1}$$

where X_n is a state variable, N is the number of components in the system, and $\alpha_n, \beta_n, g_{n,m}$ and $h_{n,m}$ are model parameters. When we try to analyze genetic networks, X_n is the expression level of the n-th gene and N is the number of genes contained in the target network. A number of inference methods based on the S-system model have been proposed (e.g., [2][9][25][29]).

The inference methods based on the S-system model seek to estimate all $2N(N+1)$ model parameters, that produce time-series data consistent with the observed gene expression levels, i.e., $\alpha_n, \beta_n, g_{n,m}$ and $h_{n,m}$ $(m,n = 1,2,\cdots,N)$. In the canonical approach, this estimation problem is defined as a $2N(N+1)$-dimensional function optimization problem. Yet in an analysis of a large-scale genetic network consisting of many genes, the problem has too many dimensions to handle with function optimization algorithms alone. The calculation time required is prohibitive even for smaller-scale genetic networks. To infer a genetic network consisting of only 5 genes, for example, PEACE1, a program coded based on the canonical problem definition, reportedly took more than 10 hours on a PC cluster (Pentium III 933MHz×1040CPUs) [12].

The problem decomposition strategy has been proposed as a means to overcome the high dimensionality in the canonical problem definition [14][19]. This technique divides the $2N(N+1)$-dimensional parameter estimation problem into N subproblems, each of which is a $2(N+1)$-dimensional function optimization problem. This strategy considerably reduces the computational cost required for the estimation of the S-system parameters. On the basis of the problem decomposition strategy, for example, we have proposed the coevolutionary method [15]. This method, running on a PC cluster (Pentium III 933MHz×8CPUs), inferred a genetic network consisting of 5 genes in only about 89.0 minutes.

With the problem decomposition strategy, we can obtain S-system models of genetic networks made up of several dozens of genes. Yet, the computational cost required for the inference of S-system models of genetic networks by this approach is still quite high. In this study, we decrease the computational cost by further dividing each $2(N+1)$-dimensional subproblem into one $(N+2)$-dimensional problem and one $(N+1)$-dimensional problem by a genetic network inference method using linear programming machines [16][17]. Next, based on our problem definition, we propose a new inference method for estimating the S-system parameters by alternately optimizing the two divided problems. Finally, we confirm the effectiveness of the proposed approach through numerical experiments.

This is an extension of our former conference paper [17]. While the former version only proposed a technique that transforms each $2(N+1)$-dimensional S-system parameter estimation problem into an $(N+2)$-dimensional problem, the current study divides each $2(N+1)$-dimensional subproblem into one $(N+2)$-dimensional problem and one $(N+1)$-dimensional problem. In addition, in order to obtain more reasonable S-system parameters, this study proposes a method to optimize the two divided problems alternately.

2. The inference method using LPMs

In this study, we apply a genetic network inference method using linear programming machines (LPMs) [16] to further divide each subproblem defined by the problem decomposition approach [14][19]. In this section, we therefore begin by describing this inference method. Hereafter, we refer to it as the LPM-based inference method.

2.1 Genetic network inference as a series of discrimination tasks

The LPM-based inference method defines the inference of a genetic network consisting of N genes as N discrimination tasks, each corresponding to one of the genes. The goal of the discrimination task is to establish a classification rule (classifier) from the given training examples. By solving the n-th discrimination task corresponding to the n-th gene, we obtain a classifier that can predict the sign of the time derivative of the expression level of the n-th gene. By analyzing the obtained classifier, we can infer genes that regulate the n-th gene. This study, on the other hand, utilizes it to estimate the S-system parameters efficiently, as described in the sections 3 and 4.

The training examples of the n-th discrimination task are

$$\left(\mathbf{X}|_{t_1}, Y_{t_1}\right), \left(\mathbf{X}|_{t_2}, Y_{t_2}\right), \cdots, \left(\mathbf{X}|_{t_K}, Y_{t_K}\right),$$

where $\mathbf{X}|_{t_k} = (X_1|_{t_k}, X_2|_{t_k}, \cdots, X_N|_{t_k})$ gives the expression levels of all of the genes at time t_k, Y_{t_k} is the label of the class to which $\mathbf{X}|_{t_k}$ belongs, and K is the number of the training examples. To construct the training examples of the n-th discrimination task, we must specify both the gene expression levels and their class labels. The gene expression levels can be measured using technologies such as DNA microarrays. The class labels are determined according to the following rules: (i) Assign `plus' to the class label $Y_{t_k}$ when the estimated derivative of the expression level of the $n$-th gene at time $t_k$ exceeds $\sigma$ (i.e., $\left.\frac{dX_n}{dt}\right|_{t_k} \geq \sigma$), where $\sigma$ (> 0) is a threshold; (ii) Similarly, assign `minus' to Y_{t_k}, when $\left.\frac{dX_n}{dt}\right|_{t_k} \leq -\sigma$; (iii) Otherwise, assign `zero' to Y_{t_k}. The time derivatives of the expression levels of the n-th gene, that are required for the assignment of the class labels of the training examples, are estimated directly from the observed time-series of the gene expression levels using some smoothing technique, such as the spline interpolation [21], the local linear regression [5], the neural network [29], or the modified Whittaker's smoother [27]. Note that, while these techniques are generally used to smooth noisy data, most of them allow us to compute the derivatives of the smoothed curves.

2.2 Learning of classifiers

One classifier obtained by the LPM-based inference method corresponds to one gene. To analyze a genetic network consisting of N genes, therefore, the inference method must obtain N classifiers. In this subsection, we describe a way to construct the n-th classifier corresponding to the n-th gene.

2.2.1 Discriminant function

To solve the n-th discrimination task corresponding to the n-th gene, the LPM-based inference

method uses the following discriminant function.

$$f(\mathbf{X}) = b + \sum_{m=1}^{N} w_m \log X_m \,, \tag{2}$$

where the input vector $\mathbf{X} = (X_1, X_2, \cdots, X_N)$ denotes the expression levels of all of the genes at a certain time, and the vector $\mathbf{w} = (w_1, w_2, \cdots, w_N)$ and b are the parameters. When $f(\mathbf{X}) > 0$, $f(\mathbf{X}) < 0$ and $f(\mathbf{X}) = 0$, the method concludes that the input vector $\mathbf{X}$ belongs to the classes `plus', `minus' and `zero', respectively. The purpose of the n-th discrimination task is therefore to find the parameters $\mathbf{w}$ and b that enable us to classify the training examples correctly (see Figure 1). This study uses these parameters to decompose the $2(N+1)$-dimensional S-system parameter estimation problems, as described in the section 3.

2.2.2 Parameter estimation

The LPM-based inference method estimates the parameters $\mathbf{w}$ and b of the discriminant function (2) by solving the following constrained function minimization problem.

$$\underset{\mathbf{w},b,\xi_i^+,\xi_i^-,\eta_i^+,\eta_i^-}{\text{minimize}} \sum_{m=1}^{N} |w_m| + C_1 \sum_{i=1}^{K^+} \xi_i^+ + C_1 \sum_{i=1}^{K^-} \xi_i^- + C_2 \sum_{i=1}^{K^0} \left(\eta_i^+ + \eta_i^-\right), \tag{3}$$

subject to

$$\begin{cases} b + \sum_{m=1}^{N} w_m \log X_{m,i}^+ \geq 1 - \xi_i^+, & \left(i = 1,2,\cdots,K^+\right), \\ \xi_i^+ \geq 0, & \left(i = 1,2,\cdots,K^+\right), \\ b + \sum_{m=1}^{N} w_m \log X_{m,i}^- \leq -1 + \xi_i^-, & \left(i = 1,2,\cdots,K^-\right), \\ \xi_i^- \geq 0, & \left(i = 1,2,\cdots,K^-\right), \\ b + \sum_{m=1}^{N} w_m \log X_{m,i}^0 \leq \eta_i^+, & \left(i = 1,2,\cdots,K^0\right), \\ \eta_i^+ \geq 0, & \left(i = 1,2,\cdots,K^0\right), \\ b + \sum_{m=1}^{N} w_m \log X_{m,i}^0 \geq -\eta_i^-, & \left(i = 1,2,\cdots,K^0\right), \\ \eta_i^- \geq 0, & \left(i = 1,2,\cdots,K^0\right), \end{cases}$$

where $\xi_i^+, \xi_i^-, \eta_i^+$ and η_i^- are slack variables, and C_1 and C_2 are constant parameters. K^+, K^- and K^0 are the numbers of the training examples labeled `plus', `minus' and `zero', respectively, and $X_{m,i}^+, X_{m,i}^-$ and $X_{m,i}^0$ are the $i$-th expression levels of the $m$-th gene contained in the training examples belonging to the classes `plus', `minus' and `zero', respectively.

This parameter estimation problem was designed based on the learning algorithm for the linear programming machine (LPM) [10], a variant of the support vector machine (SVM). Therefore, the LPM-based inference method seeks to find a hyperplane $f(\mathbf{X}) = 0$ that separates a set of the `plus' examples from a set of the `minus' examples (see Figure 1). To determine the separating hyperplane $f(\mathbf{X}) = 0$, the inference method also uses the training examples labeled `zero'. With the penalty term $C_2 \sum_{i=1}^{K^0} \left(\eta_i^+ + \eta_i^-\right)$, the method seeks to obtain the separating hyperplane positioned as close as possible to the `zero' examples. We can easily solve this problem by converting it into a linear programming problem.

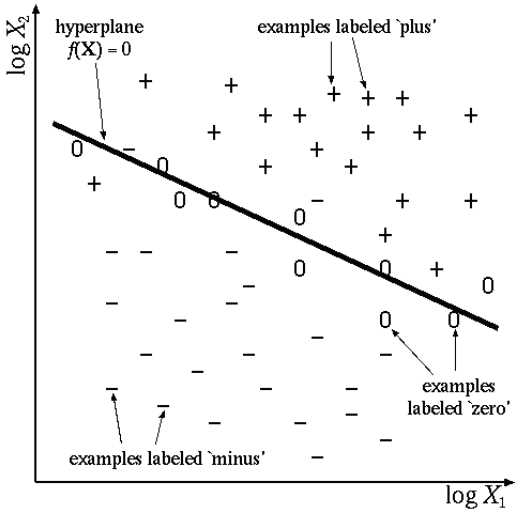

Figure 1. A classification using the discriminant function (2)
A bold line is a hyperplane $f(\mathbf{X}) = 0$. Plus, minus and zero symbols represent the training examples labeled `plus', `minus' and `zero', respectively.

2.3 Relation to S-system models

The discriminant function (2) is simple, as it is linear with respect to the logarithms of the components of the input vector **X**. The LPM-based inference method is effective, however, in analyzing genetic networks, as the following equations relate it with the S-system model (see [16][17]).

$$b = d\left(\log \alpha_n - \log \beta_n\right), \tag{4}$$

$$\mathbf{w} = d\left(\mathbf{g}_n - \mathbf{h}_n\right). \tag{5}$$

where $\mathbf{w} = (w_1, w_2, \cdots, w_N)$ and b are the parameters of the discriminant function (2), that are obtainable by solving the n-th discrimination task, α_n , β_n , $\mathbf{g}_n = (g_{n,1}, g_{n,2}, \cdots, g_{n,N})$ and $\mathbf{h}_n = (h_{n,1}, h_{n,2}, \cdots, h_{n,N})$ are the S-system parameters, and d (>0) is a single constant parameter.

While the LPM-based inference method uses the parameters **w** and b of the discriminant function (2) to infer interactions between genes, the equations above allow us to use these parameters for the estimation of the S-system parameters, as described in the next section.

3. Parameter estimation problems for S-system models

The problem decomposition strategy defines the inference of a genetic network consisting of N genes as N individual 2(N+1)-dimensional function optimization problems [14][19]. In this study, we further divide each 2(N+1)-dimensional subproblem into (N+2)-dimensional and (N+1)-dimensional problems. This section describes a way to divide the n-th subproblem corresponding to the n-th gene into two problems.

3.1 Problem decomposition strategy

The canonical approach formulates the inference of an S-system model of a genetic network containing N genes as a 2N(N+1)-dimensional function optimization problem. The problem

decomposition strategy has been proposed as a means of overcoming the difficulty in solving a high-dimensional problem in the canonical problem definition [14][19].

The problem decomposition strategy divides the genetic network inference problem into N subproblems, each corresponding to one of the genes. In the n-th subproblem corresponding to the n-th gene, we estimate the S-system parameters α_n, β_n, $\mathbf{g}_n = (g_{n,1}, g_{n,2}, \cdots, g_{n,N})$ and $\mathbf{h}_n = (h_{n,1}, h_{n,2}, \cdots, h_{n,N})$ by minimizing the following objective function.

$$F_n(\alpha_n, \beta_n, \mathbf{g}_n, \mathbf{h}_n) = \sum_{k=1}^{K} \left(\frac{X_n^{\mathrm{cal}}\big|_{t_k} - X_n^{\exp}\big|_{t_k}}{X_n^{\mathrm{cal}}\big|_{t_k}} \right)^2 + c \sum_{m=1}^{N-I} \left(|G_{n,m}| + |H_{n,m}| \right). \tag{6}$$

$G_{n,m}$ and $H_{n,m}$ in the function (6) are obtained by rearranging $g_{n,m}$ and $h_{n,m}$, respectively, in descending order of their absolute values (i.e., $|G_{n,1}| \leq |G_{n,2}| \leq \cdots \leq |G_{n,N}|$ and $|H_{n,1}| \leq |H_{n,2}| \leq \cdots \leq |H_{n,N}|$). N is the number of genes in the network, K is the number of sampling points of the observed gene expression data, c is a penalty coefficient, and I is a maximum indegree. The maximum indegree determines the maximum number of genes that directly regulate the n-th gene. $X_n^{\exp}|_{t_k}$ is an experimentally measured expression level of the n-th gene at time t_k, and $X_n^{\mathrm{cal}}|_{t_k}$ is a numerically calculated expression level acquired by solving the following differential equation.

$$\frac{dX_n}{dt} = \alpha_n \prod_{m=1}^{N} Y_m^{g_{n,m}} - \beta_n \prod_{m=1}^{N} Y_m^{h_{n,m}} , \tag{7}$$

where

$$Y_m = \begin{cases} X_m, & (\text{if } m = n), \\ \hat{X}_m, & (\text{otherwise}). \end{cases}$$

$\hat{X}_m$, an estimated expression level of the m-th gene, is obtained not by solving any differential equation, but by making a direct estimation from the observed time-series data. In this study, we obtain $\hat{X}_m$ by the same smoothing technique described in the section 2.1.

The first term on the right hand side of the objective function (6) is the sum of the squared relative errors between the observed gene expression levels and the numerically calculated expression levels. The second term, $c\sum_{m=1}^{N-I}\left(|G_{n,m}| + |H_{n,m}|\right)$, is a penalty term that forces most of the exponential parameters $g_{n,m}$'s and $h_{n,m}$'s down to zero. In the S-system model, $g_{n,m}$ and $h_{n,m}$ correspond to the regulations of the n-th gene from the m-th gene. Thus, when this penalty term is applied, most of the genes are disconnected from each other. If, however, the number of genes that directly regulate the n-th gene is lower than the maximum indegree I, this term imposes no penalty. The penalty term embodies a priori knowledge that genetic networks are sparsely connected.

The objective function (6) is $2(N+1)$-dimensional. Therefore, the problem decomposition strategy divides the $2N(N+1)$-dimensional genetic network inference problem into N individual $2(N+1)$-dimensional subproblems.

3.2 Further problem decomposition

In this subsection, we propose a new technique to divide each $2(N+1)$-dimensional subproblem into one $(N+2)$-dimensional problem and one $(N+1)$-dimensional problem. To decompose the subproblems, the proposed technique uses the parameters **w** and b obtained by the LPM-based inference method, as described below.

3.2.1 Estimation of the parameters α_n, $\mathbf{g}_n$ and s

The objective function of the (N+2)-dimensional problem [17] is

$$G_n(\alpha_n, \mathbf{g}_n, s) = \sum_{k=1}^{K} \left(\frac{X_n^{\mathrm{cal}}\big|_{t_k} - X_n^{\mathrm{exp}}\big|_{t_k}}{X_n^{\mathrm{cal}}\big|_{t_k}} \right)^2 + c \sum_{m=1}^{N-I} \left(\left| G_{n,m} \right| + \left| H_{n,m} \right| \right). \tag{8}$$

By minimizing this function, we estimate the S-system parameters α_n and $\mathbf{g}_n$, and an additional parameter s. Note that the form of this function is identical to that of the function (6). To compute the objective value of this function, therefore, we must also give the S-system parameters β_n and $\mathbf{h}_n$. These parameters are estimated directly from α_n, $\mathbf{g}_n$ and s according to the following equations.

$$\beta_n = \alpha_n \exp(-b/d), \tag{9}$$

$$\mathbf{h}_n = \mathbf{g}_n - \mathbf{w}/d, \tag{10}$$

where $\mathbf{w} = (w_1, w_2, \cdots, w_N)$ and b are the parameters obtained by solving the n-th discrimination task mentioned in the section 2, and α_n and $\mathbf{g}_n$ are the S-system parameters given as the inputs of the function G_n. Because d should be positive, this study estimates it in a logarithmic space, i.e., $d = \exp(s)$. The equations (9) and (10) are derived from the equations (4) and (5), respectively.

As mentioned above, the parameters α_n, $\mathbf{g}_n$ and s are estimated in the optimization problem of the function (8). This makes it an (N+2)-dimensional function minimization problem. The optimization of the function (8) only allows the direct estimation of the S-system parameters α_n and $\mathbf{g}_n$, but we can obtain β_n and $\mathbf{h}_n$ from the equations (9) and (10). We thus define the estimation problem of the 2(N+1) S-system parameters as an (N+2)-dimensional function optimization problem [17].

3.2.2 Adjustment of the parameters w and b

When the parameters **w** and b estimated by solving the n-th discrimination task are optimal, the S-system parameters obtained from the optimum solution of the (N+2)-dimensional problem defined just above will be identical to those of the original problem mentioned in the section 3.1. We find, however, because of the generally insufficient amounts of gene expression data observed, together with the pollution of the data by measurement error, that the LPM-based inference method often fails to find optimal values for the parameters **w** and b. As a workaround in this study, we perform the adjustment of these parameters together with the estimation of the parameters α_n, $\mathbf{g}_n$ and s.

We define the adjustment of the parameters **w** and b as an (N+1)-dimensional function minimization problem. The objective function is

$$H_n(\mathbf{w}, b) = \sum_{k=1}^{K} \left(\frac{X_n^{\mathrm{cal}}\big|_{t_k} - X_n^{\mathrm{exp}}\big|_{t_k}}{X_n^{\mathrm{cal}}\big|_{t_k}} \right)^2 + c \sum_{m=1}^{N-I} \left(\left| G_{n,m} \right| + \left| H_{n,m} \right| \right). \tag{11}$$

The function (11) is also identical, in structure, to the functions (6) and (8). To obtain an objective value for this function, therefore, we must provide the S-system parameters α_n, β_n, $\mathbf{g}_n$ and $\mathbf{h}_n$. When trying to compute the objective function (11), we fix the parameters α_n, $\mathbf{g}_n$ and s to the

values obtained through the optimization of the function G_n. Meanwhile, we estimate the parameters β_n and $\mathbf{h}_n$ directly from the input parameters **w** and b and the fixed parameters α_n, $\mathbf{g}_n$ and s according to the equations (9) and (10).

4. Efficient S-system parameter estimation

In this section, we propose a new efficient method to estimate S-system parameters by alternately optimizing the objective functions (8) and (11). In practice, any function optimization method can be used to optimize these functions. To optimize the function (8), here, we use AGLSDC [18], an evolutionary algorithm, that has shown good search performance in several optimization problems. To optimize the function (11), meanwhile, we used a local search method, the modified Powell's method [20][21], as the LPM-based inference method can yield a good initial candidate solution.

The following is an algorithm of the proposed method for the estimation of the S-system parameters corresponding to the n-th gene, i.e., α_n, β_n, $\mathbf{g}_n$ and $\mathbf{h}_n$.

<1> ***Initialization***

First, estimate the parameters **w** and b by applying the LPM-based inference method. Then, normalize the parameters **w** and b by their maximum absolute value. In this study, we represent them as $\mathbf{w}_0$ and b_0. Next, create n_p individuals randomly as an initial population of AGLSDC. Because AGLSDC is designed on the basis of real-coded genetic algorithms [7], it represents individuals as (N+2)-dimensional real number vectors, where N is the number of genes contained in the target network. We should note here that the proposed method must solve the differential equation (7) to compute the objective values of individuals. When generating individuals, however, our method sometimes fails in solving the differential equation (7). If the objective value of an individual cannot be computed, our method treats the individual as infeasible and assigns it a single objective value which is worse than the values of the possible candidate solutions. In this study, individuals are repeatedly constructed through this step, until all of the individuals contained in the initial population are feasible. Set counters T and *Generation* to 0, and set the iteration number of converging operations N_{iter} to 1.

<2> ***Optimization of the function*** $G_{\mathbf{n}}$

Repeat the steps <2.1> to <2.6> described below, n_p times.

<2.1> *Selection for reproduction*

As parents for the recombination operator, ENDX [13] (see also Appendix A), select m individuals, $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_m$, randomly without replacement from the population.

<2.2> *Generation of offspring*

Generate n_c children by applying ENDX to the selected parents. To reduce the computational cost, AGLSDC forgoes any computation of the objective values of the children generated here. Instead, this algorithm assigns each of the newly generated children a single objective value, one inferior to the objective values of all of the possible candidate solutions.

<2.3> *Application of the local search*

Optimize the objective function (8) by applying the local search method, the modified Powell's method, to the best individual in a family consisting of the two parents, $\mathbf{p}_1$ and $\mathbf{p}_2$, and their children. Note here that the children are assumed to have the worst objective value. Thus, whenever the objective values of the two parents have been actually computed in

previous generations, the algorithm applies the local search to one of the two parents. If, on the other hand, all of the individuals in the family have the same objective value, the local search is applied to an individual randomly selected from the family. When trying to compute the objective value of the function (8), we must give the parameters **w** and *b*. In this study, we use $\mathbf{w}_T$ and b_T for these parameters.

<2.4> *Selection for survival*

Select two individuals from the family. The first selected individual should be the individual with the best objective value, and the second should be selected randomly. Then, replace the two parents, i.e., $\mathbf{p}_1$ and $\mathbf{p}_2$, in the population with the two selected individuals. Note that the individual to which the local search has been applied in the step <2.3> is always selected as the best.

<2.5> *Application of the converging procedure*

Replace the individuals according to the procedure described in Appendix B, $N_{iter} - 1$ times. In AGLSDC, the individuals newly generated in this step are also assumed to have the worst objective value.

<2.6> *Adaptation of* N_{iter}

If the best individual has not improved during the last n_p generations, $N_{iter} \leftarrow 2 \times N_{iter}$. Otherwise, set N_{iter} to 1. Then, $Generation \leftarrow Generation + 1$.

<3> *Optimization of the function* H_{n}

Optimize the objective function (11) by applying the modified Powell's method to the parameters $\mathbf{w}_T$ and b_T. To compute the objective values of the function (11), we must give the parameters α_n, $\mathbf{g}_n$ and *s*, as mentioned in the section 3.2.2. For these parameters, use the current best individual of AGLSDC. Then, represent the solution optimized here as $\mathbf{w}_{T+1}$ and b_{T+1}.

<4> *Termination*

Stop if the halting criteria are satisfied. Otherwise, $T \leftarrow T + 1$ and return to the step <2>.

The steps <2.1> to <2.6> described above correspond to a single generation of AGLSDC. In the step <2>, therefore, the proposed algorithm repeats n_p generations of AGLSDC in order to optimize the function (8). In the step <3>, our algorithm uses the modified Powell's method to adjust the parameters **w** and *b*. The modified Powell's method is also used as the search operator of AGLSDC (see the step <2.3>). This local search method is an iterative optimization algorithm: in each iteration, the method applies *n* line optimizations along *n* different directions previously determined, then updates the directions, where *n* is the number of dimensions of the search space. To reduce the computational cost in this study, we discontinue the iteration for each local search when the number of iterations reaches 10. We determined this setting based on the setting of AGLSDC. Readers can find more detailed information about AGLSDC in the reference [18].

5. Numerical experiments

This section confirms the effectiveness of the proposed approach by applying it to two genetic network inference problems.

Table 1. The S-system parameters of the target model

n	α_n	$g_{n,1}$	$g_{n,2}$	$g_{n,3}$	$g_{n,4}$	$g_{n,5}$	β_n	$h_{n,1}$	$h_{n,2}$	$h_{n,3}$	$h_{n,4}$	$h_{n,5}$
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

Table 2. The S-system parameters estimated by the proposed approach

n	α_n	$g_{n,1}$	$g_{n,2}$	$g_{n,3}$	$g_{n,4}$	$g_{n,5}$	β_n	$h_{n,1}$	$h_{n,2}$	$h_{n,3}$	$h_{n,4}$	$h_{n,5}$
1	5.213	0.021	0.006	0.982	-0.028	-0.994	10.186	1.955	-0.010	0.012	0.005	0.027
2	9.509	2.053	-0.058	-0.047	-0.002	0.012	9.512	-0.036	2.027	-0.039	-0.001	0.030
3	10.366	-0.012	-0.981	0.016	0.007	0.009	10.364	-0.015	-0.982	1.932	0.021	0.011
4	10.017	-0.034	0.031	1.666	0.214	-0.973	12.020	-0.010	0.008	0.112	1.838	-0.120
5	8.582	0.020	0.014	0.115	2.057	-0.189	8.564	0.009	0.006	0.104	-0.187	2.126

Table 3. The S-system parameters estimated by the problem decomposition approach

n	α_n	$g_{n,1}$	$g_{n,2}$	$g_{n,3}$	$g_{n,4}$	$g_{n,5}$	β_n	$h_{n,1}$	$h_{n,2}$	$h_{n,3}$	$h_{n,4}$	$h_{n,5}$
1	5.214	0.026	0.003	0.990	-0.039	-0.992	10.189	1.975	-0.022	0.029	-0.017	0.020
2	9.453	2.065	-0.065	-0.030	-0.008	0.022	9.453	-0.043	2.041	-0.039	-0.001	0.039
3	10.275	-0.002	-1.001	0.021	0.001	-0.006	10.277	-0.002	-1.001	1.960	0.005	-0.006
4	8.982	-0.005	0.043	1.877	0.076	-1.005	10.957	-0.004	0.028	0.076	1.923	-0.061
5	8.429	0.005	0.015	0.128	2.090	-0.198	8.423	0.001	0.007	0.112	-0.189	2.144

5.1 Experiment 1: Inference of an artificial genetic network from noise-free data

In this experiment, we confirm that our approach is capable of correctly inferring an S-system model of a genetic network when a sufficient amount of noise-free data are given.

5.1.1 Experimental setup

The target network for this experiment is an S-system model of a genetic network consisting of 5 genes (N=5). Table 1 gives the model parameters of this system. The inference of this system has often been used as a benchmark problem for inference methods (e.g., [2][12][14][15][16][17][25]). The problem decomposition strategy divided the genetic network inference problem of this system into 5 subproblems, each a 12-dimensional function optimization problem. The proposed approach further decomposed each 12-dimensional subproblem into 7-dimensional and 6-dimensional problems, and then optimized them alternately.

As the observed gene expression patterns, 15 sets of time-series data, each covering 5 genes, were computed from the differential equations (1) in the target model. The sets began from randomly generated initial values in [0.0, 2.0], and 11 sampling points for the time-series data were assigned to each gene in each set. No measurement noise was simulated in the computed data. As the gene expression data constructed here contained no measurement noise, we used the spline interpolation [21] to obtain $\hat{X}_m$'s used in the equation (7). The quality of the model inferred by the proposed approach depends closely on the precision of the estimated gene expression levels, $\hat{X}_m$'s. This requires that we carefully choose a smoothing technique to obtain $\hat{X}_m$'s when attempting to

solve actual genetic network inference problems. Several techniques have been used to obtain reasonable $\hat{X}_m$'s [5][21][27][29]. Our approach, on the other hand, requires the application of the LPM-based inference method, as described in the section 4. To use the LPM-based inference method, we also must provide the time derivatives of the gene expression levels. In actual genetic network inference problems, the derivatives must be estimated from the smoothed gene expression levels, $\hat{X}_m$'s. In this study, however, we computed the derivatives directly from the target model. The S-system parameters of the target model in this study were estimated solely from the gene expression levels and their derivatives.

We set the hyper-parameters of AGLSDC as follows: the population size n_p was 3*n*, and the number of the children generated by a recombination operator per selection n_c was 10, where *n* is the dimension of the search space. The following recommended values were used as the parameters for the LPM-based inference method in this study: $C_1 = 200/(N\sqrt{K})$, $C_2 = 0.4C_1$ and $\sigma = 0.15$, where *K* is the number of the training examples. The parameters for the objective functions, i.e., *I* and *c*, were set to 5 and 1.0, respectively. The search regions of the parameters were [0.0, 20.0] for α_n, [-3.0, 3.0] for $g_{n,m}$ and [-4.0,1.0] for *s*. We performed 10 trials, each with different sets of gene expression data. Each trial was continued until the number of generations reached $50n_p$, or the population converged within the range of 10^{-6} in each coordinate. For comparison, we also performed experiments where AGLSDC is used to optimize the objective function (6).

In order to reduce the computational cost, this study applied a structure skeletalizing technique [24]. This technique assigns a value of zero to the S-system parameters $g_{n,m}$ and $h_{n,m}$, whose absolute values are less than the given threshold δ_s. According to the references [14][15], this study set δ_s to 10^{-3}.

5.1.2 Results

Samples of the S-system parameters estimated by the proposed approach and the problem decomposition approach are listed in Tables 2 and 3, respectively. As the tables show, both approaches fell short of estimating the parameter values with perfect precision. We should note, however, that the large absolute values of the parameters $g_{n,m}$ and $h_{n,m}$ indicate the strong regulations of the *n*-th gene from the *m*-th gene in the S-system model. When the *m*-th gene does not regulate the *n*-th gene, on the other hand, $g_{n,m}$ and $h_{n,m}$ are 0. The tables therefore show that both approaches correctly inferred the structure of the target network. We should note that both approaches succeeded in estimating reasonable parameter values for $g_{3,2}$ and $h_{3,2}$. A number of inference methods, meanwhile, have failed in inferring the regulations corresponding to these parameters (see, e.g., [3][16][26]).

The proposed approach succeeded in estimating the S-system parameters of the target network within a shorter computation time. The proposed approach, including the LPM-based inference method, required about 83.2 minutes on a single-CPU personal computer (Pentium IV 2.8GHz) to estimate all of the S-system parameters. The problem decomposition approach took much longer, about 265.2 minutes, on the same computer. The objective values of the solutions obtained by the proposed approach were slightly worse than those obtained by the problem decomposition approach. The averaged objective values of the proposed approach and the problem decomposition approach were $4.201\times10^{-3} \pm 6.660\times10^{-3}$ and $2.614\times10^{-3} \pm 4.200\times10^{-3}$, respectively. While the relative

error between the time-courses obtained from the problem decomposition approach and the given time-series data at each time point, i.e., $\left|X_n^{\mathrm{cal}}|_t - X_n^{\mathrm{exp}}|_t\right| / X_n^{\mathrm{cal}}|_t$, averaged about $2.074\times10^{-3} \pm 3.623\times10^{-3}$, that of the proposed approach averaged about $2.847\times10^{-3} \pm 4.539\times10^{-3}$. Yet, as mentioned earlier, the S-system parameters estimated by the proposed approach were precise enough to biologically interpret the network.

5.2 Experiment 2: Analysis of the SOS DNA repair system in *E.coli*

Next, we check the performance of the proposed approach in an experiment using actual biological data.

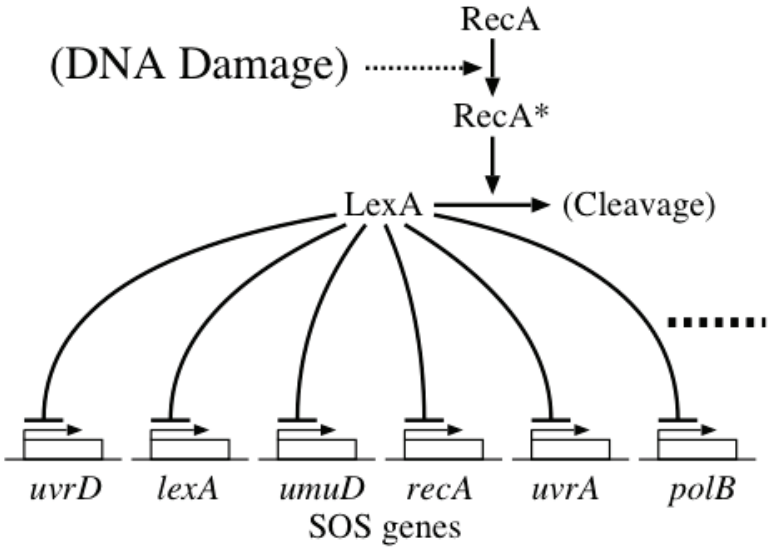

Figure 2. The SOS DNA repair system in *E.coli*

5.2.1 Experimental setup

This experiment used the proposed approach to analyze the actual gene expression data of the SOS DNA repair system in *E.coli* [23]. More than 30 genes are known to be involved in this system. In a basal state, a master repressor, LexA, is bound to the promoter region of these genes to suppress their expression. DNA damage activates one of the SOS proteins, RecA, and RecA then mediates LexA autocleavage. The drop in the LexA level, in turn, halts the repression of the SOS genes. Once the damage has been repaired, RecA stops mediating LexA autocleavage, LexA accumulates and represses the SOS genes, and the cells return to their basal state (Figure 2).

In this experiment, we applied the proposed approach to the expression data measured by Ronen and his colleagues [22]. We selected six genes, i.e., *uvrD*, *lexA*, *umuD*, *recA*, *uvrA* and *polB*, according to the work done by Cho and his colleagues [2], and then, inferred the genetic network of these genes. Though the original data contain 4 sets of time-series data, we used only 2 sets (the third and fourth sets) measured under the same experimental condition. Each set of the time-series data consisted of 50 measurements including the initial concentrations, which were zero. In this experiment, we removed the initial concentrations from both sets, since models based on a set of differential equations cannot produce different time-courses from the same initial conditions. Thus, each set of the time-series data consisted of 49 sampling points. Based on our previous work [16], we normalized the data corresponding to each gene against its maximum expression level. Next, we

smoothed the normalized data by the local linear regression [5]. We assigned a value of 10^{-6} to expression levels with values less than 10^{-6}, since the LPM-based inference method is incapable of treating expression levels equal to zero.

We carried out 10 trials by changing the seed for the pseudo random number used in AGLSDC. The following parameter settings were used: $\sigma = 0.001$, $I = 2$ and $c = 5.0$. The search regions of the parameters were [0.0, 3.0] for α_n, [-3.0, 3.0] for $g_{n,m}$ and [-4.0, 1.0] for s. All of the other experimental conditions were the same as those used in the previous experiment.

Note that the time delay is generally observed during the actual gene regulation phenomenon. As the model used in this study cannot reflect it, however, we disregarded it in this experiment.

5.2.2 Results

The proposed method inferred the S-system model with a shorter computation time, but still with inferior precision. The problem decomposition approach analyzed this system in 8.44 hours on a single-CPU personal computer (Pentium IV 2.8GHz). Our approach analyzed the system in an average of 2.54 hours on the same computer. The averaged objective values of the solutions obtained by the problem decomposition approach and our approach were, on the other hand, $9.322 \times 10^0 \pm 9.328 \times 10^0$ and $1.392 \times 10^1 \pm 8.257 \times 10^0$, respectively.

To confirm the effectiveness of the proposed approach, we next extracted the structure of the genetic network from the estimated S-system parameters according to the following rules: (i) When $g_{n,m} > 0.05$ and/or $h_{n,m} < -0.05$, we conclude that the m-th gene positively regulates the n-th gene; (ii) Similarly, when $g_{n,m} < -0.05$ and/or $h_{n,m} > 0.05$, we conclude that the m-th gene negatively regulates the n-th gene; (iii) Otherwise, we infer no regulation of the n-th gene from the m-th gene. The proposed approach is based on the stochastic search algorithm, hence the network structures inferred by it were slightly different from each other in the 10 trials. However, most of the inferred regulations were the same. Figure 3 shows the core network structure where the proposed approach inferred the regulations more than 9 times in the 10 trials. In Figure 4, we also show the core network structure inferred by the problem decomposition approach.

The objective values of the proposed approach were worse than those of the problem decomposition approach, but the networks inferred by the proposed approach were more reasonable. The figures show that the proposed approach succeeded in finding the regulations of all of the genes from *lexA* and the regulation of *lexA* from *recA*, although they were positive. The regulation of *umuD* from *recA*, that is contained in a network now known [8], also appeared in the network inferred by our approach. The problem decomposition approach, on the other hand, failed in inferring some of them. Our approach, however, inferred a larger number of regulations, most of which likely to be false-positive. A focus for future work will be to reduce the number of regulations inferred by the proposed approach.

The experimental results indicate that, even when the problem decomposition approach yields better objective values, the genetic networks inferred by it are not always reasonable. The unreasonable solutions obtained by the problem decomposition approach are probably attributable to the multimodality of this genetic network inference problem. The use of the LPM-based inference method limits the search space of the S-system parameters, which gives our approach the ability to infer reasonable networks. When trying to infer genetic networks, therefore, we should not only minimize the difference between the observed gene expression levels and the numerically computed expression levels but also utilize the time derivatives estimated from the observed time-series of the gene expression levels.

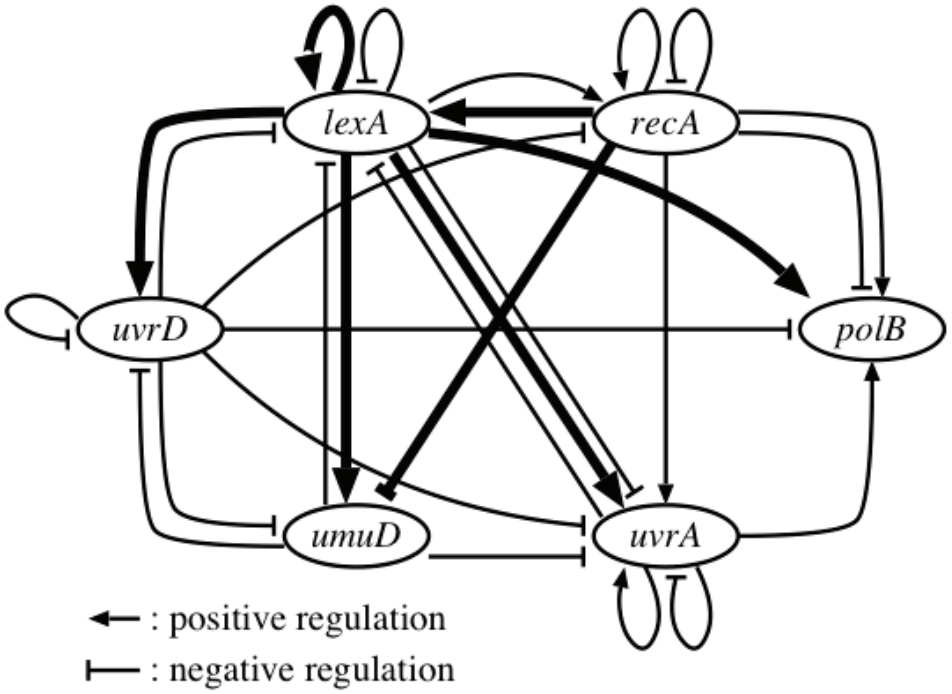

Figure 3. The core network structure inferred by the proposed approach
A bold lines represent biologically plausible regulations mentioned in the section 5.2.2.

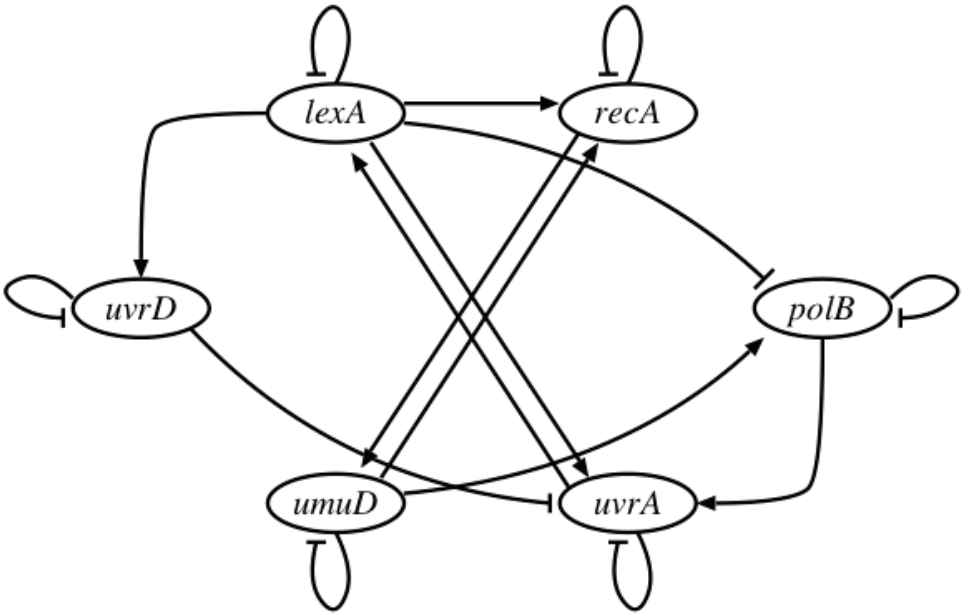

Figure 4. The core network structure inferred by the problem decomposition approach

6. Conclusion

In this study, we proposed a new efficient approach for the estimation of S-system parameters. The proposed approach divides each 2(N+1)-dimensional subproblem defined by the problem decomposition strategy into one (N+2)-dimensional problem and one (N+1)-dimensional problem,

where N is the number of genes contained in the target network. To divide the subproblems, the proposed approach uses the LPM-based inference method. Next, our approach estimates the S-system parameters by alternately optimizing the two divided problems. Then, through the experiments on the artificial genetic network inference problem, we showed that the proposed approach is more than 3 times faster than the problem decomposition approach. While the objective values of the solutions obtained by our approach were slightly worse, the estimated parameters were precise enough to biologically interpret the network. Finally, through an analysis of the gene expression data of the SOS DNA repair system in *E.coli*, we showed that our approach can be expected to infer reasonable genetic networks.

Several techniques have been already proposed to reduce the computational cost for the estimation of S-system parameters (see, e.g., [3][4][9][26][29]). Therefore, the computation time of the proposed approach is not always the shortest. However, we can apply the idea of the proposed approach to many other techniques developed for the inference of S-system models. The application of our idea will help us decrease the computational costs of these inference techniques.

Appendix A. ENDX

An extended normal distribution crossover (ENDX) [13] is a recombination operator for real-coded genetic algorithms. ENDX requires m parents, $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_m$, and then generates a child $\mathbf{c}$ according to the following equation.

$$\mathbf{c} = \mathbf{p} + \xi \mathbf{d} + \sum_{i=3}^{m} \eta_i \mathbf{p}'_i ,$$

where

$$\mathbf{p} = (\mathbf{p}_1 + \mathbf{p}_2)/2 ,$$

$$\mathbf{d} = \mathbf{p}_2 - \mathbf{p}_1 ,$$

$$\mathbf{p}'_i = \mathbf{p}_i - \frac{1}{m-2} \sum_{j=3}^{m} \mathbf{p}_j ,$$

ξ and η_i are random numbers drawn from normal distributions $N(0, \alpha^2)$ and $N(0, \beta^2)$, respectively. This study used the following values for the hyper-parameters of ENDX; α=0.434, $\beta = 0.35/\sqrt{m-3}$ and m=min(12, n+2), where n is the dimension of the search space.

Appendix B. Converging procedure of AGLSDC

To converge the population, AGLSDC repeats the following procedure.

1. **[Selection for reproduction]**
   Select m individuals without replacement from the population. The selected individuals, expressed here as $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_m$, are used as the parents for ENDX applied in the next step.
2. **[Generation of offspring]**
   Generate n_c children by applying ENDX to the parents selected in the previous step. AGLSDC assigns the newly generated children a single objective value, one which is inferior to the objective values of any of the possible candidate solutions.
3. **[Selection for survival]**

Select two individuals from a family containing the two parents, i.e., $\mathbf{p}_1$ and $\mathbf{p}_2$, and their children. The first selected individual should be the one with the best objective value, and the second should be selected randomly. Then, replace the two parents in the population with the selected individuals.

References

[1] Akutsu, T.; Miyano, S.; Kuhara, S. Inferring Qualitative Relations in Genetic Networks and Metabolic Pathways. *Bioinformatics.* **2000**, *16*, 727-734.

[2] Cho, D.Y.; Cho, K.H.; Zhang, B.-T. Identification of Biochemical Networks by S-tree based Genetic Programming. *Bioinformatics.* **2006**, *22*, 1631-1640.

[3] Chou, I.-C.; Martens, H.; Voit, E.O. Parameter Estimation in Biochemical Systems Models with Alternating Regression. *Theoretical Biology and Medical Modelling.* **2006**, *3*, 25.

[4] Chou, I.-C.; Voit, E.O. Recent Developments in Parameter Estimation and Structure Identification of Biochemical and Genomic Systems. *Mathematical Biosciences.* **2009**, *219*, 57-83.

[5] Cleveland, W.S. Robust Locally Weight Regression and Smoothing Scatterplots. *J. of American Statistical Association.* **1979**, *74*, 829-836.

[6] D'haeseleer, P.; Liang, S.; Somogyi, R. Genetic Network Inference: From Co-expression Clustering to Reverse Engineering. *Bioinformatics.* **2000**, *16*, 707-726.

[7] Eshelman E.J.; Schaffer, J.D. Real-coded Genetic Algorithms and Interval Schemata. *Proc. of Foundations of Genetic Algorithms.* **1993**, *2*, 187-202.

[8] Gardner, T.S.; di Bernardo, D.; Lorenz, D.; Collins, J.J. Inferring Genetic Networks and Identifying Compound Mode of Action via Expression Profiling. *Science.* **2003**, *301*, 102-105.

[9] Gonzalez, O.R.; Küper, C.; Jung, K.; Naval Jr., P.C.; Mendoza, E. Parameter Estimation using Simulated Annealing for S-system Models of Biochemical Networks. *Bioinformatics.* **2007**, *23*, 480-486.

[10] Graepel, T.; Herbrich, R.; Schölkopf, B.; Smola, A.; Bartlett, P.; Müller, K.R.; Obermayer, K.; Williamson, R. Classification on Proximity Data with LP-machines. *Proc. of Int. Conf. on Artificial Neural Networks'99.* **1999**, 304-309.

[11] Kabir, M.; Noman, N.; Iba, H. Reverse Engineering Gene Regulatory Network from Microarray Data using Linear Time-variant Model. *BMC Bioinformatics.* **2010**, *11*(Suppl 1), S56.

[12] Kikuchi, S.; Tominaga, D.; Arita, M.; Takahashi, K.; Tomita, M. Dynamic Modeling of Genetic Networks using Genetic Algorithm and S-system. *Bioinformatics.* **2003**, *19*, 643-650.

[13] Kimura, S.; Ono, I.; Kita, H.; Kobayashi, S. An Extension of UNDX based on Guidelines for Designing Crossover Operators: Proposition and Evaluation of ENDX. *Trans. of the Society of Instrument and Control Engineers.* **2000**, *36*, 1162-1171 (in Japanese).

[14] Kimura, S.; Hatakeyama, M.; Konagaya, A. Inference of S-system Models of Genetic Networks from Noisy Time-series Data. *CBIJ.* **2004**, *4*, 1-14.

[15] Kimura, S.; Ide, K.; Kashihara, A.; Kano, M.; Hatakeyama, M.; Masui, R.; Nakagawa, N.; Yokoyama, S.; Kuramitsu, S.; Konagaya, A. Inference of S-system Models of Genetic Networks using a Cooperative Coevolutionary Algorithm. *Bioinformatics.* **2005**, *21*, 1154-1163.

[16] Kimura, S.; Nakayama, S.; Hatakeyama, M. Genetic Network Inference as a Series of Discrimination Tasks. *Bioinformatics.* **2009**, *25*, 918-925.

[17] Kimura, S.; Amano, Y.; Matsumura, K.; Okada-Hatakeyama, M. Effective Parameter Estimation for S-system Models using LPMs and Evolutionary Algorithms. *Proc. of 2010 Congress on Evolutionary Computation.* **2010**, 2034-2041.
[18] Kimura, S.; Nakakuki, T.; Kirita, S.; Okada, M. AGLSDC: a Genetic Local Search suitable for Parallel Computation. *SICE J. of Control, Measurement, and System Integration*, in press.
[19] Maki, Y.; Ueda, T.; Okamoto, M.; Uematsu, N.; Inamura, Y.; Eguchi, Y. Inference of Genetic Network using the Expression Profile Time Course Data of Mouse P19 Cells. *Genome Informatics.* **2002**, *13*, 382-383.
[20] Powell, M.J.D. An Efficient Method for Finding the Minimum of a Function of Several Variables without Calculating Derivatives. *Computer J.* **1964**, *7*, 155-162.
[21] Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. *Numerical Recipes in C 2nd Edition.* **1995**, Cambridge University Press.
[22] Ronen, M.; Rosenberg, R.; Shraiman, B.I.; Alon, U. Assigning Numbers to the Arrows: Parameterizing a Gene Regulation Network by using Accurate Expression Kinetics. *Proc. of National Academy of Sciences of USA.* **2002**, *99*, 10555-10560.
[23] Sutton, M.D.; Smith, B.T.; Godoy, V.G.; Walker, G.C. The SOS Response: Recent Insights into umuDC-dependent Mutagenesis and DNA Damage Tolerance. *Annual Review of Genetics.* **2000**, *34*, 479-497.
[24] Tominaga, D.; Koga, N.; Okamoto, M. Efficient Numerical Optimization Algorithm based on Genetic Algorithm for Inverse Problem. *Proc. of Genetic and Evolutionary Computation Conference.* **2000**, 251-258.
[25] Tsai, K.Y.; Wang, F.S. Evolutionary Optimization with Data Collocation for Reverse Engineering of Biological Networks, *Bioinformatics.* **2005**, *21*, 1180-1188.
[26] Veflingstad, S.R.; Almeida, J.; Voit, E.O. Priming Nonlinear Searches for Pathway Identification. *Theoretical Biology and Medical Modelling.* **2004**, *1*, 8.
[27] Vilela, M.; Borges, C.C.H.; Vinga, S.; Vanconcelos, A.T.R.; Santos, H.; Voit, E.O.; Almeida, J.S. Automated Smoother for the Numerical Decoupling of Dynamic Models. *BMC Bioinformatics.* **2007**, *8*, 305.
[28] Voit, E.O. *Computational Analysis of Biochemical Systems.* **2000**, Cambridge University Press.
[29] Voit, E.O.; Almeida, J. Decoupling Dynamical Systems for Pathway Identification from Metabolic Profiles. *Bioinformatics.* **2004**, *20*, 1670-1681.
[30] Yeung, M.K.S.; Tegnér, J.; Collins, J.J. Reverse Engineering Gene Networks using Singular Value Decomposition and Robust Regression. *Proc. of National Academy of Sciences of USA.* **2002**, *99*, 6163-6168.
[31] Yu, J.; Smith, V.A.; Wang, P.P; Hartemink, A.J.; Jarvis, E.D. Advances to Bayesian Network Inference for Generating Causal Networks from Observational Biological Data. *Bioinformatics.* **2004**, *20*, 3594-3603.