IPSJ Transactions on Bioinformatics Vol.7 30–38 (Dec. 2014)

[DOI: 10.2197/ipsjtbio.7.30]

Original Paper

An Effective Method for the Inference of Reduced S-system Models of Genetic Networks

Shuhei Kimura[1,a)] Masanao Sato[2] Mariko Okada-Hatakeyama[3]

Received: July 17, 2014, Accepted: October 2, 2014, Released: December 19, 2014

Abstract: The inference of genetic networks is a problem to obtain mathematical models that can explain observed time-series of gene expression levels. A number of models have been proposed to describe genetic networks. The S-system model is one of the most studied models among them. Due to its advantageous features, numerous inference algorithms based on the S-system model have been proposed. The number of the parameters in the S-system model is however larger than those of the other well-studied models. Therefore, when trying to infer S-system models of genetic networks, we need to provide a larger amount of gene expression data to the inference method. In order to reduce the amount of gene expression data required for an inference of genetic networks, this study simplifies the S-system model by fixing some of its parameters to 0. In this study, we call this simplified S-system model a reduced S-system model. We then propose a new inference method that estimates the parameters of the reduced S-system model by minimizing two-dimensional functions. Finally, we check the effectiveness of the proposed method through numerical experiments on artificial and actual genetic network inference problems.

Keywords: reduced S-system model, genetic network, linear programming problem, decoupling

1. Introduction

High-throughput technologies, such as RNA-seq, make it possible to measure gene expression patterns on a genomic scale. Several researchers have become interested in the inference of genetic networks as one means of extracting useful information from the measured gene expression data. The genetic network is a functional circuit in living cells at the gene level, and can be considered as an abstract mapping of an actual biochemical network consisting of genes, proteins, metabolites, and so on. The inference of genetic networks is therefore conceived of as one promising way to understand biological systems.

The purpose of inference of genetic networks is to obtain mathematical models that can explain observed time-series of gene expression levels. In order to describe genetic networks, a number of models have been proposed [2], [6], [17], [28], [31]. An S-system model [21], [29] is one of the most studied models among them. This model possesses a rich structure capable of capturing various dynamics and can be analyzed by several available methods. Because of its advantageous features, numerous inference algorithms based on the S-system model have thus been proposed [3], [4], [9], [11], [12], [13], [16], [18], [26], [30]. In genetic network inferences based on the S-system model, we must estimate $2N(N+1)$ model parameters, where N is the number of genes contained in the target network. The number of the parameters of the S-system model is larger than those of the other well-studied models, such as the linear model [31], the Vohradský's model [28], and so on. When trying to infer S-system models of genetic networks, therefore, we need to provide more gene expression data to the inference method. As it is generally difficult to measure a sufficient amount of the gene expression data, however, the requirement for this larger amount of data is a drawback for inference approaches based on the S-system model.

In order to overcome the drawback of the S-system approaches, this study decreases the number of model parameters that need to be estimated by fixing some of them to 0. We refer to this simplified S-system model as a reduced S-system model in this study. In order to infer reduced S-system models of genetic networks, we could use existing algorithms that were developed for inferring S-system models. These methods generally estimate model parameters by solving non-linear function optimization problems whose dimensions depend on the number of genes contained in the target genetic network. When trying to infer a genetic network consisting of many genes, therefore, they must solve high-dimensional function optimization problems. In order to resolve the high-dimensionality in the parameter estimation, this study proposes an effective method that uses features of the reduced S-system model. The proposed method overcomes the high-dimensionality by defining the inference of the reduced S-system model of a genetic network consisting of N genes as N individual two-dimensional function optimization problems. As the defined two-dimensional functions seem to be multimodal, this study uses REX^{star}/JGG [15], an evolutionary algorithm, to optimize them. Finally, we confirm the effectiveness of the proposed approach through numerical experiments on artificial and

[1] Graduate School of Engineering, Tottori University, Tottori 680–8552, Japan
[2] National Institute for Basic Biology, Okazaki Institute for Integrative Bioscience, National Institute for Natural Sciences, Okazaki, Aichi 444–8787, Japan
[3] RIKEN Center for Integrative Medical Sciences, Yokohama, Kanagawa 230–0045, Japan
[a)] kimura@ike.tottori-u.ac.jp

actual genetic network inference problems.

2. Reduced S-system Model

The S-system model [21], [29] is a set of non-linear differential equations of the form

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

where X_n is the n-th state variable and N is the number of components in the network, α_n (> 0) and β_n (> 0) are multiplicative parameters called rate constants, and $g_{n,m}$ and $h_{n,m}$ are exponential parameters called kinetic orders. In the genetic network inference, X_n is the expression level of the n-th gene and N is the number of genes contained in the target network. The inference of an S-system model of a genetic network consisting of N genes is defined as the estimation problem of $2N(N+1)$ model parameters, i.e., α_n, β_n, $g_{n,m}$ and $h_{n,m}$ ($n, m = 1, 2, \cdots, N$), that produce time-series consistent with the observed gene expression data.

In the genetic network inference, it is often important to know whether the m-th gene regulates the n-th gene or not, and whether the inferred regulation of the n-th gene from the m-th gene is positive or negative. In the S-system model, the kinetic orders $g_{n,m}$ and $h_{n,m}$ represent the regulations of the n-th gene from the m-th gene. When X_m promotes or suppresses the synthesis of X_n, values for $g_{n,m}$ are positive or negative, respectively. Similarly, values for $h_{n,m}$ are positive or negative, when X_m promotes or suppresses the degradation of X_n, respectively. In the S-system approaches, we generally assume that the m-th gene positively regulates the n-th gene when $g_{n,m}$ is positive and/or $h_{n,m}$ is negative. When $g_{n,m}$ is negative and/or $h_{n,m}$ is positive, on the other hand, the n-th gene is assumed to be negatively regulated by the m-th gene. When the m-th gene has no influence on the n-th gene, $g_{n,m}$ and $h_{n,m}$ are both zero. When we try to extract the information about the regulations from the observed gene expression data, therefore, the S-system model seems to be redundant.

In order to remove the redundancy from the S-system model, this study fixes $h_{n,m}$ ($n \neq m$) to 0. We call this simplified model a reduced S-system model. The reduced S-system model is thus defined as

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

Note that, while the number of parameters we must estimate in the S-system model is $2N(N+1)$, that in the reduced S-system model is $N(N+3)$.

When trying to infer genetic networks, several researchers have already used models that are obtained by restricting the S-system model [3], [24]. Even when the numbers of parameters of these models are less than that of the original S-system model, they reportedly still have an ability to represent genetic networks. In order to estimate their parameters, however, the references [3], [24] used the inference methods developed for the S-system model. In this study, on the other hand, we propose a parameter estimation method that utilizes a unique feature of the reduced S-system model, as described below.

3. Parameter Estimation

This study proposes an effective method for estimating parameters of the reduced S-system model. The proposed method divides the inference problem of the reduced S-system model of a genetic network consisting of N genes into N subproblems, each of which is defined as a two-dimensional function optimization problem. By solving the n-th subproblem, our method estimates the parameters corresponding to the n-th gene, i.e., α_n, β_n, $\mathbf{g}_n = (g_{n,1}, g_{n,2}, \cdots, g_{n,N})$ and $h_{n,n}$. This section describes a method for solving the n-th subproblem.

3.1 Problem Definition

In the n-th subproblem corresponding to the n-th gene, the proposed method estimates the model parameters α_n, β_n, $\mathbf{g}_n$ and $h_{n,n}$ by solving a set of the following algebraic equations.

$$\begin{aligned}
\left.\frac{dX_n}{dt}\right|_{t_1} &= \alpha_n \textstyle\prod_{m=1}^{N} \left(X_m|_{t_1}\right)^{g_{n,m}} - \beta_n \left(X_n|_{t_1}\right)^{h_{n,n}}, \\
\left.\frac{dX_n}{dt}\right|_{t_2} &= \alpha_n \textstyle\prod_{m=1}^{N} \left(X_m|_{t_2}\right)^{g_{n,m}} - \beta_n \left(X_n|_{t_2}\right)^{h_{n,n}}, \\
&\vdots \\
\left.\frac{dX_n}{dt}\right|_{t_K} &= \alpha_n \textstyle\prod_{m=1}^{N} \left(X_m|_{t_K}\right)^{g_{n,m}} - \beta_n \left(X_n|_{t_K}\right)^{h_{n,n}},
\end{aligned} \qquad (3)$$

where $X_m|_{t_k}$ is the expression level of the m-th gene at time t_k, and $\left.\frac{dX_n}{dt}\right|_{t_k}$ is the time derivative of the expression level of the n-th gene at time t_k, and K is the number of measurements. In the proposed method, $X_m|_{t_k}$'s are measured using gene expression profiling technologies such as RNA-seq. $\left.\frac{dX_n}{dt}\right|_{t_k}$'s are, on the other hand, estimated directly from the observed time-series of the gene expression levels using a smoothing technique such as a spline interpolation [19], a local linear regression [7], a neural network [30], or a modified Whittaker's smoother [27]. Based on an idea similar to the method described here, several genetic network inference methods have already been proposed [5], [13], [14], [30], [31].

3.2 Effective Technique for Solving Simultaneous Equations

This study estimates the model parameters α_n, β_n, $\mathbf{g}_n = (g_{n,1}, g_{n,2}, \cdots, g_{n,N})$ and $h_{n,n}$ by solving the simultaneous Eqs. (3). Note however that these equations are non-linear with respect to the parameters. Moreover, the number of the parameters we must estimate is proportional to the number of genes contained in a network, i.e., N. Therefore, it is not always easy to solve Eqs. (3). In order to overcome the difficulty in solving them, this study proposes the effective method described below.

3.2.1 Concept

The proposed method resolves the difficulty in solving the simultaneous Eqs. (3) by using a feature that arises from their transformation, described below.

By rearranging the k-th member of Eqs. (3), we obtain

$$\left.\frac{dX_n}{dt}\right|_{t_k} + \beta_n \left(X_n|_{t_k}\right)^{h_{n,n}} = \alpha_n \prod_{m=1}^{N} \left(X_m|_{t_k}\right)^{g_{n,m}}. \qquad (4)$$

By taking the logarithms of both sides of the equation above, we then have

$$\log\left[\left.\frac{dX_n}{dt}\right|_{t_k} + \beta_n \left(X_n|_{t_k}\right)^{h_{n,n}}\right] = \log\alpha_n + \sum_{m=1}^{N} g_{n,m} \log\left(X_m|_{t_k}\right). \qquad (5)$$

Note that, although the transformed Eq. (5) is non-linear with respect to the parameters β_n and $h_{n,n}$, it is linear with respect to the parameters $\log \alpha_n$ and $\mathbf{g}_n = (g_{n,1}, g_{n,2}, \cdots, g_{n,N})$. This fact suggests that, when the parameters β_n and $h_{n,n}$ are given, the other parameters α_n and $\mathbf{g}_n$ are easily estimated. The proposed method utilizes this feature to solve the simultaneous Eqs. (3), as mentioned below.

3.2.2 Objective Function

As mentioned just above, we can easily estimate the parameters α_n and $\mathbf{g}_n$, when the parameters β_n and $h_{n,n}$ are given. The proposed method therefore solves the simultaneous Eqs. (3) simply by estimating the parameters β_n and $h_{n,n}$. This study thus defines the problem of solving the simultaneous equations as a minimization problem of the following two-dimensional function.

$$S_n(\beta_n, h_{n,n}) = \sum_{k=1}^{K} \left[\left. \frac{dX_n}{dt} \right|_{t_k} - \alpha_n^* \prod_{m=1}^{N} \left(X_m|_{t_k} \right)^{g^*_{n,m}} + \beta_n \left(X_n|_{t_k} \right)^{h_{n,n}} \right]^2 + \max\{0, d_n(\beta_n, h_{n,n})\}, \quad (6)$$

where

$$d_n(\beta_n, h_{n,n}) = \max\left\{ \beta_n \left(X_n|_{t_1} \right)^{h_{n,n}}, \beta_n \left(X_n|_{t_2} \right)^{h_{n,n}}, \cdots, \beta_n \left(X_n|_{t_K} \right)^{h_{n,n}} \right\} - c_d \times \max\left\{ \left| \left. \frac{dX_n}{dt} \right|_{t_1} \right|, \left| \left. \frac{dX_n}{dt} \right|_{t_2} \right|, \cdots, \left| \left. \frac{dX_n}{dt} \right|_{t_K} \right| \right\},$$

$\max\{\cdot\}$ is an operator that returns the maximum value of a set of elements, c_d is a constant parameter, and α_n^* and $\mathbf{g}_n^* = (g^*_{n,1}, g^*_{n,2}, \cdots, g^*_{n,N})$ are the optimal values for α_n and $\mathbf{g}_n$, respectively, under given β_n and $h_{n,n}$. The next section describes a way to obtain α_n^* and $\mathbf{g}_n^*$.

The proposed approach is based on the least-squares method. The first term of the function (6) is therefore a sum of the squared errors between the left-hand sides and the right-hand sides of Eqs. (3). The second term is, on the other hand, a penalty term to avoid β_n being excessively large. This term tries to keep the maximum absolute values of $\beta_n \left(X_n|_{t_k} \right)^{h_{n,n}}$ and $\left. \frac{dX_n}{dt} \right|_{t_k}$ contained in the simultaneous Eqs. (3) to a similar size. When this term was not applied, our method often got trapped in local optima where β_n is large, $\alpha_n = \beta_n$, $g_{n,n} = h_{n,n}$ and $g_{n,m} = 0$ $(m \neq n)$. According to our preliminary experiments, this study set the parameter c_d to 10.

3.2.3 Estimation of α_n^* and $\mathbf{g}_n^*$

As mentioned in Section 3.2.2, when trying to compute a value for the objective function (6), we must always obtain values for α_n^* and $\mathbf{g}_n^* = (g^*_{n,1}, g^*_{n,2}, \cdots, g^*_{n,N})$. In the proposed approach, they serve as the solution of the transformed simultaneous Eqs. (5) under given β_n and $h_{n,n}$. Note here that, when values for β_n and $h_{n,n}$ are given, Eqs. (5) are linear with respect to $\log \alpha_n$ and $\mathbf{g}_n$. Therefore, it is easy to solve these equations. This study defines the problem of solving them as the following constrained function minimization problem.

$$\underset{\log \alpha_n, \mathbf{g}_n, \xi_k^+, \xi_k^-}{\text{minimize}} \; C \sum_{k=1}^{K} \gamma_k \left(\xi_k^+ + \xi_k^- \right) + \sum_{m=1}^{N} |g_{n,m}|, \quad (7)$$

subject to

$$\begin{aligned} & L_k - \log \alpha_n - \textstyle\sum_{m=1}^{N} g_{n,m} \log\left(X_m|_{t_k} \right) \leq \xi_k^+, && (k = 1, 2, \cdots, K), \\ & \xi_k^+ \geq 0, && (k = 1, 2, \cdots, K), \\ & L_k - \log \alpha_n - \textstyle\sum_{m=1}^{N} g_{n,m} \log\left(X_m|_{t_k} \right) \geq -\xi_k^-, && (k = 1, 2, \cdots, K), \\ & \xi_k^- \geq 0, && (k = 1, 2, \cdots, K), \end{aligned}$$

where

$$L_k = \log(Z_k),$$

$$Z_k = \begin{cases} \left. \frac{dX_n}{dt} \right|_{t_k} + \beta_n \left(X_n|_{t_k} \right)^{h_{n,n}}, & \left(\text{if } \left. \frac{dX_n}{dt} \right|_{t_k} + \beta_n \left(X_n|_{t_k} \right)^{h_{n,n}} \geq \delta \right), \\ \delta, & (\text{otherwise}), \end{cases}$$

ξ_k^+ and ξ_k^- are slack variables, and γ_k, δ and C are constant parameters. Note that, when solving this problem, we treat the parameters β_n and $h_{n,n}$ as constants.

ξ_k^+ and ξ_k^- represent the differences between the left-hand side and the right-hand side of the k-th member of the transformed Eqs. (5). The first term of the objective function of the problem (7), i.e., $C \sum_{k=1}^{K} \gamma_k \left(\xi_k^+ + \xi_k^- \right)$, is thus the weighted sum of the absolute errors between the left-hand sides and the right-hand sides of the transformed equations. Note that, while the problem described in Section 3.2.2 tries to solve the simultaneous Eqs. (3), the problem described here tries to solve the transformed Eqs. (5). On the other hand, the second term, i.e., $\sum_{m=1}^{N} |g_{n,m}|$, is a penalty term that forces most of $g_{n,m}$'s down to 0. As mentioned in Section 2, $g_{n,m}$ is set to 0 when the m-th gene does not regulate the n-th gene. When this penalty term is applied, therefore, most of the genes are disconnected from each other. We introduce this term, since genetic networks are known to be sparsely connected [23]. The constant parameter C therefore determines the tradeoff between the goodness of fit and the sparseness of the inferred network.

As mentioned in Section 3.2.1, in order to estimate the model parameters, the proposed method uses the feature arising from the transformation of Eqs. (3). Note here that, only when the condition $\left. \frac{dX_n}{dt} \right|_{t_k} + \beta_n \left(X_n|_{t_k} \right)^{h_{n,n}} > 0$ is satisfied, we can transform the k-th member of Eqs. (3). Even when the optimum values are set for β_n and $h_{n,n}$, however, the noise contained in the measurement data might make this condition unsatisfied. This study thus introduces a threshold parameter δ, and sets its value to 1.0×10^{-6}. On the other hand, we should note that, when $\left. \frac{dX_n}{dt} \right|_{t_k} + \beta_n \left(X_n|_{t_k} \right)^{h_{n,n}}$ approaches 0, the term $\log\left[\left. \frac{dX_n}{dt} \right|_{t_k} + \beta_n \left(X_n|_{t_k} \right)^{h_{n,n}} \right]$ contained in Eq. (5) approaches $-\infty$. When $\left. \frac{dX_n}{dt} \right|_{t_k} + \beta_n \left(X_n|_{t_k} \right)^{h_{n,n}}$ is small, therefore, the transformation of the equation would amplify the noise contained in the measurement data. We should not rely too much on the equations transformed under this condition. In order to introduce this notion into our parameter estimation, this study sets the constant parameter γ_k to

$$\gamma_k = \frac{Z_k}{\sum_{j=1}^{K} Z_j}.$$

We can transform the optimization problem (7) to a linear programming problem. Thus, the proposed method solves this problem by using the simplex method [25].

3.3 Algorithm

As mentioned previously, our approach divides the inference of a genetic network consisting of N genes into N subproblems. In

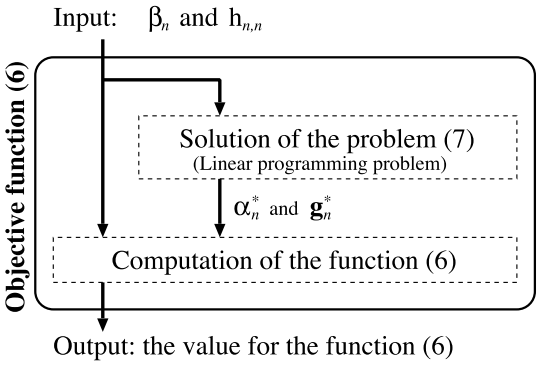

Fig. 1 The computation of the objective function (6).

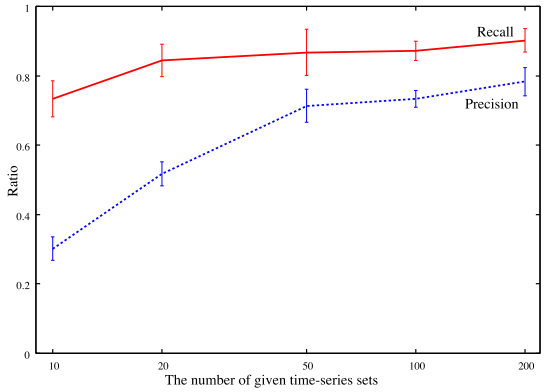

Fig. 2 Performances of the proposed method on experiments with different numbers of time-series datasets. Solid and dotted lines represent the recall and the precision of the proposed method, respectively.

the n-th subproblem corresponding to the n-th gene, the proposed method estimates the parameters $\alpha_n, \beta_n, \mathbf{g}_n = (g_{n,1}, g_{n,2}, \cdots, g_{n,N})$ and $h_{n,n}$ by minimizing the objective function (6). Note that, when computing a value for this function, we must always solve the constrained function minimization problem (7) (see **Fig. 1**). As the problem (7) is converted to a linear programming problem, however, we can easily solve it using the simplex method. On the other hand, we can use any function optimization algorithm to optimize the objective function (6). While this function is only two-dimensional, however, it seemed to be multimodal. In order to minimize it, thus, this study uses REX^{star}/JGG (see Appendix A.1) [15], an evolutionary algorithm.

4. Numerical Experiments

4.1 Inference of Artificial Networks

In this experiment, we confirm that the proposed method has an ability to infer structures of genetic networks.

4.1.1 Experimental Setup

This experiment used reduced S-system models consisting of 30 genes ($N = 30$) as target networks. As the inference ability of the proposed method may depend on the structure of the target network, we generated the target networks of different structures by changing the model parameters. When trying to determine the model parameters corresponding to the n-th gene, we randomly chose an integer k from a power-law distribution with a cutoff of 5. Then, k genes were randomly selected from all of the genes contained in the network. The kinetic orders $g_{n,m}$'s corresponding to the regulations of the n-th gene from the selected genes were randomly chosen from $[-1.0, 1.0]$, and the other $g_{n,m}$'s were set to 0.0. The kinetic order $h_{n,n}$ and the rate constants α_n and β_n were all set to 1.0. This study changed the network structure on every trial.

As the performance of the inference method also depends on the amount of given time-series data, we performed the experiments with different numbers of time-series datasets. The time-series datasets were obtained by solving the differential Eqs. (2) on the target networks. The initial values of these sets were selected randomly from $[0.0, 2.0]$. Each dataset consisted of the expression levels at 11 time points. The measurement noise was simulated by adding 10% Gaussian noise to the computed time-series data. In order to estimate the time derivatives of the gene expression levels from the given time-series datasets, we used the local linear regression [7], a smoothing technique.

In order to check the performance of the proposed method, this study constructed and then solved 10 genetic network inference problems with each available number of time-series datasets. The search area of the parameter $h_{n,n}$ was $[-5, 5]$. As the other parameter β_n is positive, on the other hand, this study searched for it in a logarithmic space. The search area of $\log \beta_n$ was $[-20, 10]$. According to its recommended settings, this study set the following values for the parameters of the optimization algorithm, REX^{star}/JGG [15]: the population size n_p is 40, the number of children generated per selection n_c is 6, and the step-size parameter t is 2.5. Each run of REX^{star}/JGG was continued until the number of generation alternations reached 500. Based on the preliminary experiments, we set the constant parameter C contained in the defined problem (7) to 30.

4.1.2 Results

As the given data were noisy, it was difficult to use the proposed method to estimate model parameters precisely. In this experiment, therefore, we only compared the structures of the inferred networks with those of the target networks. This study extracted the structures of the networks from the estimated model parameters according to the rules used for the S-system model [13]: when $g_{n,m} \geq Th_n$ and/or $h_{n,m} \leq -Th_n$, we conclude that the m-th gene positively regulates the n-th gene, where Th_n is a threshold; similarly, this study concludes that the n-th gene is negatively regulated by the m-th gene, when $g_{n,m} \leq -Th_n$ and/or $h_{n,m} \geq Th_n$; otherwise, we infer no regulation of the n-th gene from the m-th gene. As the threshold, this study used

$$Th_n = \gamma \max\{|g_{n,1}|, |g_{n,2}|, \cdots, |g_{n,N}|, |h_{n,1}|, |h_{n,2}|, \cdots, |h_{n,N}|\},$$

where γ is a parameter, and this study set its value to 0.05 [14]. Note that, as we used the reduced S-system model in this study, the values for the parameters $h_{n,m}$ $(n \neq m)$ are all 0.

The recalls and the precisions of the proposed method on the experiments with 10, 20, 50, 100 and 200 sets of time-series data are shown in **Fig. 2**. The recall and the precision are defined as

$$\text{(recall)} = \frac{TP}{TP + FN}, \quad \text{(precision)} = \frac{TP}{TP + FP},$$

where TP, FN and FP are the numbers of true-positive, false-negative and false-positive regulations, respectively. The recall increases from 0 to 1 with decrease in the number of false-negative regulations, and the precision increases from 0 to 1 with

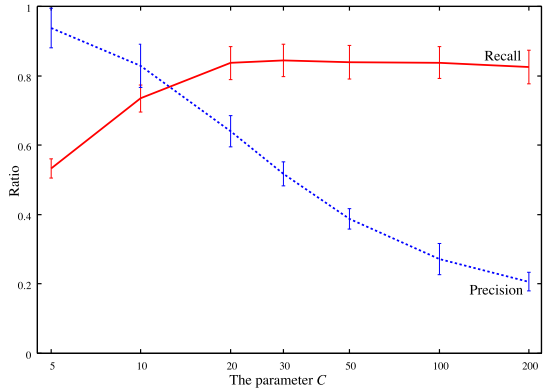

Fig. 3 Performances of the proposed method applying different values of parameter C on the experiments with 20 sets of noisy time time-series data.

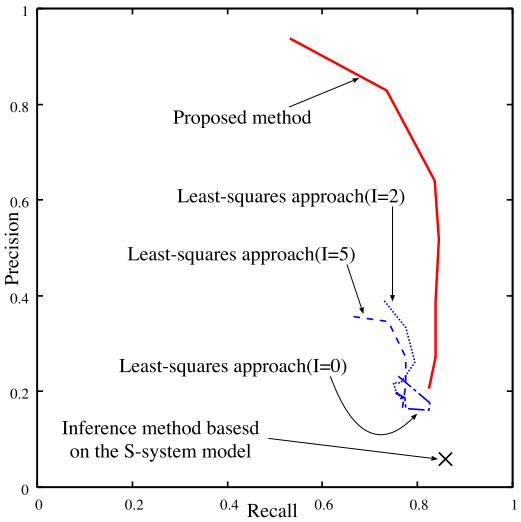

Fig. 4 Precision versus recall for the genetic network inference problems of 30 genes with 20 sets of noisy time-series data. A solid line represents the performances of the proposed method. Dash-dotted, dotted and dashed lines represent the performances of the least-squares approach with $I = 0$, $I = 2$ and $I = 5$, respectively. These curves were obtained by changing the parameters of the inference methods. A symbol '×' represents the performances of the inference method based on the S-system model [13].

a decrease in the number of false-positive regulations. Figure 2 shows that the recall and the precision of our method increase with an increase in the amount of given time-series data. On the other hand, as described above, this study set the parameter C to 30. However, as shown in **Fig. 3**, the performance of the proposed method depends on the parameter C. When we try to analyze actual genetic networks, thus, we should carefully determine the value of C.

As mentioned in Section 3, this study defines the estimation of the model parameters corresponding to the n-th gene as a problem of solving the simultaneous Eqs. (3). The method proposed in this study effectively solves them by minimizing the two-dimensional function (6). We can however solve the simultaneous Eqs. (3) simply by using the least-squares method. This study thus constructed an inference method based on the simple least-squares method (see Appendix A.2), and then compared the proposed method with it. In this study, we call this inference method a least-squares approach. **Figure 4** shows the precision-recall curves of the proposed method and the least-squares approach on the experiments with 20 sets of noisy time-series data. The least-squares approach was performed under different parameter settings, i.e., $I = 0$, $I = 2$ and $I = 5$. These curves were obtained by changing the parameter of our method, C, from 5 to 200, and that of the least-squares approach, D_{lsq}, from 0.05 to 5. The figure indicates that our method outperforms the least-squares approach. Because of the low-dimensionality of the proposed objective function (6), our method would succeed in finding reasonable results with a higher probability. The computation time of the proposed method was also shorter. While the least-squares approach took 158.2 ± 21.5 minutes on a personal computer (Core i5-4670 3.4 GHz) to infer each network, the proposed method took 35.0 ± 1.6 minutes on the same computer. However, the computation time of our method is not always shortest. In order to analyze each network, for example, the inference method based on the S-system model [13] required only 12.6 ± 0.7 seconds on a personal computer (Pentium IV 2.8 GHz). However, this method inferred a lot of false-positive regulations. Although its recall was comparable to that of the proposed method, its precision was much worse (see Fig. 4). The feature that the proposed method infers a fewer number of false-positive regulations may be due to the lower degree-of-freedom of the reduced S-system model.

The proposed method could not eliminate erroneous regulations from the networks inferred in the experiments described above. When noise-free data are provided, however, our method has an ability to estimate model parameters with high precision. The averaged objective value (6) of the proposed method with $C = 3000$ was $1.237 \times 10^{-11} \pm 5.529 \times 10^{-12}$ on the experiments with 20 sets of noise-free time-series data. The averaged difference between the true model parameter values and the estimated ones was $2.081 \times 10^{-7} \pm 4.684 \times 10^{-7}$.

4.2 Inference of an Actual Network

Next, we apply the proposed method to an actual genetic network inference problem.

4.2.1 Experimental Setup

We applied the proposed inference method to an actual inference problem from the SOS DNA repair regulatory network in *E.coli* [22]. Many genes, including *lexA* and *recA*, are known to be involved in this system. These genes are regulated by *lexA* and *recA*. In a basal state, LexA, a master repressor, is bound to the interaction site in the promoter regions of these genes. When DNA is damaged, RecA, another SOS protein, senses the damage and mediates LexA autocleavage. The decrease in LexA protein level halts the repression of the SOS genes, and then they start the DNA repair. 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.

This experiment analyzed the expression data of six genes, i.e., *uvrD*, *lexA*, *umuD*, *recA*, *uvrA* and *polB*, that had been measured by Ronen and colleagues [20] ($N = 6$). Consequently, in order to infer the genetic network, this study solved 6 individual two-dimensional function optimization problems. These expression data have often been used to confirm the performances of inference methods [3], [4], [10], [13], [14]. The original expres-

Table 1 Estimated model parameters in the experiment on the bacterial SOS DNA repair system.

n	α_n β_n	$g_{n,1}$ $h_{n,n}$	$g_{n,2}$	$g_{n,3}$	$g_{n,4}$	$g_{n,5}$	$g_{n,6}$	C AIC
1 (*uvrD*)	9.096×10^{-2} 5.747×10^{-3}	8.512×10^{0} 1.313×10^{0}	-2.179×10^{1}	-2.650×10^{0}	-2.200×10^{0}	2.235×10^{1}	-1.506×10^{0}	13,000 -7.988×10^{2}
2 (*lexA*)	6.906×10^{-1} 4.484×10^{-1}	2.785×10^{0} 3.455×10^{0}	-3.438×10^{0}	-4.689×10^{0}	1.492×10^{0}	6.937×10^{0}	-4.106×10^{-1}	20,000 -8.708×10^{2}
3 (*umuD*)	2.173×10^{-1} 1.879×10^{-2}	1.389×10^{1} 1.678×10^{0}	-2.405×10^{1}	0.000×10^{0}	-1.293×10^{1}	3.669×10^{1}	-4.191×10^{0}	12,000 -9.902×10^{2}
4 (*recA*)	3.397×10^{-2} 2.189×10^{-2}	2.155×10^{0} 1.772×10^{0}	0.000×10^{0}	-9.891×10^{0}	-1.346×10^{1}	2.744×10^{1}	-2.052×10^{0}	7,000 -1.077×10^{3}
5 (*uvrA*)	2.307×10^{-1} 5.341×10^{-2}	8.679×10^{0} 1.720×10^{0}	-1.747×10^{1}	0.000×10^{0}	-4.495×10^{0}	2.059×10^{1}	-2.128×10^{0}	6,000 -9.889×10^{2}
6 (*polB*)	3.182×10^{-2} 6.396×10^{-3}	-3.540×10^{-2} 9.144×10^{-1}	-6.269×10^{0}	-2.117×10^{0}	-1.107×10^{0}	9.921×10^{0}	0.000×10^{0}	10,000 -9.622×10^{2}

sion data contained four sets of time-series data. This experiment however used only two sets (the third and fourth sets), since those two had been measured under the same experimental conditions. Each set of time-series data consisted of 50 measurement values including the initial concentrations of 0. This experiment however removed the initial concentrations from both sets as models based on a set of differential equations cannot produce different time-courses from the same initial conditions. The number of measurements K is thus $2 \times 49 = 98$. We normalized the data corresponding to each gene against its maximum expression level. This experiment then smoothed the normalized gene expression data using the local linear regression [7]. We assigned a value of 10^{-6} to expression levels with values of less than 10^{-6}, as the gene expression levels must not be negative. The time derivatives of the gene expression levels were estimated from the smoothed data.

Before executing the experiments, it is difficult for us to determine a value for the parameter C contained in the problem (7). This study thus performed the experiments by changing the parameter C from 1,000 to 20,000. For each experiment with a different parameter setting, we performed 10 trials by changing a seed for pseudo random numbers. All of the other experimental conditions were kept the same as those of the previous experiment.

4.2.2 Results

In the proposed approach, the number of inferred regulations roughly decreases with a decreasing the parameter C. The complexity of the obtained model thus decreases with a decrease in C. On the other hand, the goodness of fit of the obtained model for the observed gene expression data improves with an increase in its complexity. When trying to infer genetic networks, we should obtain simpler mathematical models that fit the observed gene expression levels better. In general, we can use the Akaike information criterion (AIC) [1] to determine the tradeoff between the goodness of fit and the model complexity. In this study, we thus chose the most reasonable results with respect to AIC (**Table 1**). These results indicate that the reasonable value for the parameter C differs for every subproblem. This study computes the AIC value of the n-th sub-model corresponding to the n-th gene according to

$$\mathrm{AIC} = -2 \log Li_n + 2N_f, \tag{8}$$

where

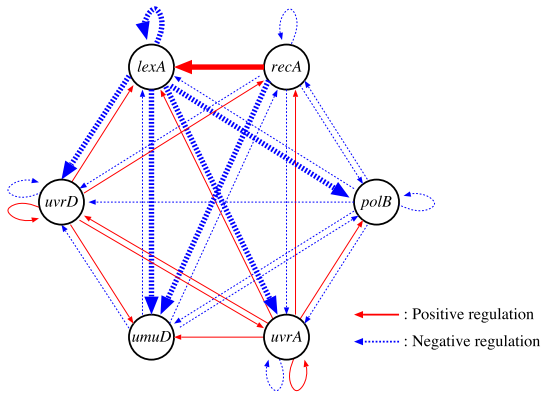

Fig. 5 The network structure obtained for the SOS DNA repair regulatory network in *E.coli*. Bold lines represent biologically plausible regulations.

$$Li_n = \prod_{k=1}^{K} \left[\frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{\delta_{n,k}^2}{2\sigma_n^2} \right) \right],$$

$$\delta_{n,k} = \left. \frac{dX_n}{dt} \right|_{t_k} - \alpha_n \prod_{m=1}^{N} \left(X_m|_{t_k} \right)^{g_{n,m}} + \beta_n \left(X_n|_{t_k} \right)^{h_{n,n}},$$

$$\sigma_n^2 = \frac{1}{K} \sum_{k=1}^{K} \delta_{n,k}^2,$$

and N_f is the number of free parameters contained in the n-th sub-model. Note here that, when a smaller number of regulations are inferred, the model becomes simpler. This study thus defines N_f as

$$N_f = N_{total} - N_0,$$

where N_{total} is the total number of the parameters of the n-th sub-model, i.e., $N_{total} = N + 3$, and N_0 is the number of the kinetic orders, $g_{n,m}$'s and $h_{n,n}$, whose absolute values are less than 10^{-6}.

According to the rules described in Section 4.1.2, we extracted the structure of the network from the estimated parameters given in Table 1 (**Fig. 5**). The inferred network contained some reasonable regulations. As mentioned previously, LexA is known to repress SOS genes. Therefore, although the regulation of *recA* from *lexA* was not inferred, the regulations of the other genes from *lexA* would be reasonable. Likewise, the regulation of *lexA* from *recA* also appears to be reasonable, as RecA senses the damage of DNA and mediates LexA autocleavage. In addition, the regulation of *umuD* from *recA*, inferred by the proposed method, has been contained in a network currently known [8]. Although some

of the other inferred regulations might be new findings, most of them should be false-positive.

In the proposed approach, we can control the complexity of the inferred model by using the parameter C. In order to choose a reasonable value for this parameter, we proposed to use AIC. As mentioned above, however, the inferred network still seems to have a lot of erroneous regulations. In a future work, therefore, we need to find a way to reduce these erroneous regulations. For this purpose, we are now planning to use other a priori knowledge about genetic networks.

5. Conclusion

The S-system model has been considered appropriate for representing biochemical networks. However, this model has a larger number of parameters. In order to infer reasonable genetic networks, therefore, the inference method based on the S-system model requires a larger amount of gene expression data. In order to resolve the drawbacks of the S-system approach, this study first proposed a reduced S-system model that is obtained by simplifying the S-system model. The number of the parameters of the original S-system model is $2N(N + 1)$, where N is the number of genes contained in the network. On the other hand, that of the reduced S-system model is $N(N + 3)$. This study then proposed the genetic network inference method based on the reduced S-system model that utilizes unique features of the model. The proposed method effectively estimates the model parameters by solving the two-dimensional function optimization problems. The experimental results indicate that the proposed method has an ability to infer genetic networks reasonably well even with a smaller amount of gene expression data. This is an advantageous feature, since it is generally difficult to measure a sufficient amount of gene expression data.

We can simulate the gene expression of the target system by solving a set of differential Eqs. (2) with the estimated model parameters. As the proposed method estimates the parameters without solving any differential equations, however, the computed time-courses of the gene expression levels would not resemble the observed data. Therefore, our method should be used not for the computational simulation, but mainly for the inference of a structure of the target network. For the computational simulation, we should use other inference methods that estimate the parameters with solving a set of differential Eqs. (2). As these methods must solve differential equations many times, however, their computational costs are generally high. By using the model parameters estimated by the proposed method as an initial guess for these inference methods, we could decrease their computational costs.

The number of the regulations inferred by the proposed method depends on the parameter C. Thus, this study also proposed a technique to choose a reasonable value for the parameter C. Through the experiments with the actual gene expression data, however, we found that the models obtained by using reasonable values for the parameter C still seem to produce a number of false-positive regulations. In future work, therefore, we aim to develop a technique to reduce them.

Acknowledgments In order to solve the linear programming problems defined in this study, we used the Coin-or linear programming, an open-source linear programming solver. This work was supported by JSPS KAKENHI Grant Number 26330275.

References

[1] Akaike, H.: Information Theory and an Extention of the Maximum Likelihood Principle, *Proc. 2nd International Symposium on Information Theory*, pp.267–281 (1973).
[2] Akutsu, T., Miyano, S. and Kuhara, S.: Inferring Qualitative Relations in Genetic Networks and Metabolic Pathways, *Bioinformatics*, Vol.16, No.8, pp.727–734 (2000).
[3] Chemmangattuvalappil, N., Task, K. and Banerjee, I.: An Integer Optimization Algorithm for Robust Identification of Non-linear Gene Regulatory Networks, *BMC Systems Biology*, Vol.6: 119 (2012).
[4] Cho, D.-Y., Cho, K.-H. and Zhang, B.-T.: Identification of Biochemical Networks by S-tree Based Genetic Programming, *Bioinformatics*, Vol.22, No.13, pp.1631–1640 (2006).
[5] Chou, I.C., Martens, H. and Voit, E.O.: Parameter Estimation in Biochemical Systems Models with Alternating Regression, *Theoretical Biology and Medical Modelling*, Vol.3: 25 (2006).
[6] Chou, I.-C. and Voit, E.O.: Recent Developments in Parameter Estimation and Structure Identification of Biochemical and Genomic Systems, *Mathematical Biosciences*, Vol.219, No.2, pp.57–83 (2009).
[7] Cleveland, W.S.: Robust Locally Weight Regression and Smoothing Scatterplots, *J. American Statistical Association*, Vol.79, No.368, pp.829–836 (1979).
[8] Gardner, T.S., di Bernardo, D., Lorenz, D. and Collins, J.J.: Inferring Genetic Networks and Identifying Compound Mode of Action via Expression Profiling, *Science*, Vol.301, pp.102–105 (2003).
[9] Gonzalez, O.R., Küpper, C., Jung, K., Naval Jr., P.C. and Mendoza, E.: Parameter Estimation Using Simulated Annealing for S-system Models of Biochemical Networks, *Bioinformatics*, Vol.23, No.4, pp.480–486 (2007).
[10] Kabir, S., Noman, N. and Iba, H.: Reverse Engineering Gene Regulatory Network from Microarray Data Using Linear Time-variant Model, *BMC Bioinformatics*, Vol.11: S56 (2010).
[11] Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K. and Tomita, M.: Dynamic Modeling of Genetic Networks Using Genetic Algorithm and S-system, *Bioinformatics*, Vol.19, No.5, pp.643–650 (2003).
[12] Kimura, S., Ide, K., Kashihara, A., Kano, M., Hatakeyama, M., Masui, R., Nakagawa, N., Yokoyama, S., Kuramitsu, S. and Konagaya, A.: Inference of S-system Models of Genetic Networks Using a Cooperative Coevolutionary Algorithm, *Bioinformatics*, Vol.21, No.7, pp.1154–1163 (2005).
[13] Kimura, S., Araki, D., Matsumura, K. and Okada-Hatakeyama, M.: Inference of S-system Models of Genetic Networks by Solving One-dimensional Function Optimization Problems, *Mathematical Biosciences*, Vol.235, No.2, pp.161–170 (2012).
[14] Kimura, S., Sato, M. and Okada-Hatakeyama, M.: Inference of Vohradský's Models of Genetic Networks by Solving Two-dimensional Function Optimization Problems, *PLoS One*, Vol.8: e83308 (2013).
[15] Kobayashi, S.: The Frontiers of Real-coded Genetic Algorithms, *Trans. of the Japanese Society for Artificial Intelligence*, Vol.24, No.1, pp.147–162 (2009) (in Japanese).
[16] Liu, P.-K. and Wang, F.-S.: Inference of Biochemical Network Models in S-system Using Multiobjective Optimization Approach, *Bioinformatics*, Vol.24, No.8, pp.1085–1092 (2008).
[17] Margolin, A.A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R.D. and Califano, A.: ARACNE: an Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context, *BMC Bioinformatics*, Vol.7: S7 (2006).
[18] Nakatsui, M., Ueda, T., Maki, Y., Ono I. and Okamoto, M.: Method for Inferring and Extracting Reliable Genetic Interactions from Time-series Profile of Gene Expression, *Mathematical Biosciences*, Vol.215, No.1, pp.105–114 (2008).
[19] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P.: *Numerical Recipes in C 2nd Edition*, Cambridge University Press, Cambridge, UK (1995).
[20] Ronen, M., Rosenberg, R., Shraiman, B.I. and Alon, U.: Assigning Numbers to the Arrows: Parameterizing a Gene Regulation Network by Using Accurate Expression Kinetics, *Proc. National Academy of Sciences of USA*, Vol.99, No.16, pp.10555–10560 (2002).
[21] Savageau, M.A.: Biochemical Systems Analysis I. Some Mathematical Properties of the Rate Law for the Component Enzymatic Reactions, *J. of Theoretical Biology*, Vol.25, No.3, pp.365–369 (1969).
[22] Sutton, M.D., Smith, B.T., Godoy, V.G. and Walker, G.C.: The SOS Response: Recent Insights into umuDC-dependent Mutagenesis and DNA Damage Tolerance, *Annual Review of Genetics*, Vol.34, pp.479–497 (2000).

[23] Thieffry, D., Huerta, A.M., Pérez-Rueda, E. and Collado-Vides, J.: From Specific Gene Regulation to Genomic Networks: a Global Analysis of Transcriptional Regulation in *Escherichia Coli*, *BioEssays*, Vol.20, No.5, pp.433–440 (1998).
[24] Thomas, R., Mehrotra, S., Papoutsakis, E.T. and Hatzimanikatis, V.: A Model-based Optimization Framework for the Inference on Gene Regulatory Networks from DNA Array Data, *Bioinformatics*, Vol.20, No.17, pp.3221–3235 (2004).
[25] Todd, M.J.: The Many Facets of Linear Programming, *Mathematical Programming*, Vol.91, No.3, pp.417–436 (2002).
[26] Tsai, K.-Y. and Wang, F.-S.: Evolutionary Optimization with Data Collocation for Reverse Engineering of Biological Networks, *Bioinformatics*, Vol.21, No.7, pp.1180–1188 (2005).
[27] Vilela, M., Borges, C.C.H., Vinga, S., Vasconcelos, A.T.R., Santos, H., Voit, E.O. and Almeida, J.S.: Automated Smoother for the Numerical Decoupling of Dynamics Models, *BMC Bioinformatics*, Vol.3: 305 (2007).
[28] Vohradský, J.: Neural Network Model of Gene Expression, *FASEB J.*, Vol.15, No.3, pp.846–854 (2001).
[29] Voit, E.O.: *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*, Cambridge University Press, Cambridge, UK (2000).
[30] Voit, E.O. and Almeida, J.: Decoupling Dynamical Systems for Pathway Identification from Metabolic Profiles, *Bioinformatics*, Vol.20, No.11, pp.1670–1681 (2004).
[31] Yeung, M.K.S., Tegnér, J. and Collins, J.J.: Reverse Engineering Gene Networks Using Singular Value Decomposition and Robust Regression, *Proc. National Academy of Sciences of USA*, Vol.99, No.9, pp.6163–6168 (2002).

Appendix

A.1 REX^{star}/JGG

REX^{star}/JGG [15] is a real-coded genetic algorithm, a sort of evolutionary algorithm, that uses JGG as a generation alternation model and REX^{star} as a recombination operator. This section describes each of the operators in detail.

A.1.1 JGG

JGG is a generation alternation model. The generation alternation model is a procedure for selecting individuals to breed and for selecting individuals to form a new population in the next generation. The following is an algorithm of JGG.

[Algorithm: MGG]

(1) **Initialization**
As an initial population, create n_p individuals. As REX^{star}/JGG is a real-coded genetic algorithm, these individuals are represented as s-dimensional real number vectors, where s is the dimension of the search space. Set *Generation* = 0.

(2) **Selection for reproduction**
Select m individuals without replacement randomly from the population. The selected individuals, that are expressed here as $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_m$, are used as the parents for the recombination operator in the next step.

(3) **Generation of offspring**
Generate n_c children by applying the recombination operator to the parents selected in the previous step. This study uses REX^{star} as the recombination operator, and it requires $s+1$ individuals as parents, i.e., $m = s+1$.

(4) **Selection for survival**
Select the best m individuals from the family containing the m parents, i.e., $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_m$, and their children. Then, replace the m parents with the selected individuals. In the original JGG, the best m individuals are selected only from the children. As its optimization process seemed to be unstable, however, the algorithm is slightly modified in this study.

(5) **Termination**
Stop if the halting criteria are satisfied. Otherwise, *Generation* ← *Generation* + 1, and then return to step 2.

A.1.2 REX^{star}

REX^{star} is a real-coded crossover operator. REX^{star} uses $s+1$ parents, where s is the dimension of the search space, and generate n_c ($> s+1$) children according to the following algorithm.

[Algorithm: REX^{star}]

(1) Generate reflection points, $\underline{\mathbf{p}_1}, \underline{\mathbf{p}_2}, \cdots, \underline{\mathbf{p}_{s+1}}$, of the parents $\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_{s+1}$, i.e.,

$$\underline{\mathbf{p}_i} = 2\mathbf{G} - \mathbf{p}_i, \tag{A.1}$$

where

$$\mathbf{G} = \frac{1}{s+1}\sum_{i=1}^{s+1}\mathbf{p}_i.$$

(2) Compute the objective values of the $s+1$ reflection points generated in the previous step. In REX^{star}, these reflection points are treated as the children.

(3) From the parents and their reflection points, select the best $s+1$ individuals, and then compute the center of the gravity of the selected individuals. This study represents it as $\mathbf{G}^b$.

(4) Generate $n_c - s - 1$ children by applying the following equation $n_c - s - 1$ times. Note that the $s+1$ reflection points generated in step 1 are treated as the children. The total number of the children generated is therefore n_c.

$$\mathbf{c} = \mathbf{G} + \mathrm{diag}(\xi_1^t, \xi_2^t, \cdots, \xi_s^t)(\mathbf{G}^b - \mathbf{G}) + \sum_{i=1}^{s+1}\xi^i(\mathbf{p}_i - \mathbf{G}), \tag{A.2}$$

where $\mathbf{c}$ represents a child, and ξ_i^t's and ξ^i's are random numbers drawn from uniform distributions $[0, t]$ and $\left[-\sqrt{\frac{3}{s+1}}, \sqrt{\frac{3}{s+1}}\right]$, respectively, where t is a constant parameter named a step-size parameter.

In Ref. [15], the following settings are recommended for the parameters of REX^{star}/JGG: the population size n_p is set between $2s$ and $20s$, the number of children generated per selection n_c is set between $2s$ and $3s$, and the step-size parameter t is set between 2.5 and 15.

A.2 Least-squares Approach

The proposed method estimates the model parameters corresponding to the n-th gene by solving the simultaneous Eqs. (3) effectively. However, we can solve these equations simply by using the least-squares method. In this case, for example, we estimate the parameters $\alpha_n, \beta_n, \mathbf{g}_n = (g_{n,1}, g_{n,2}, \cdots, g_{n,N})$ and $h_{n,n}$ by minimizing

$$\begin{aligned} &T_n(\alpha_n, \beta_n, \mathbf{g}_n, h_{n,n}) \\ &= \sum_{k=1}^{K}\left[\left.\frac{dX_n}{dt}\right|_{t_k} - \alpha_n\prod_{m=1}^{N}\left(X_m|_{t_k}\right)^{g_{n,m}} + \beta_n\left(X_n|_{t_k}\right)^{h_{n,n}}\right]^2 \\ &+\max\{0, d_n(\beta_n, h_{n,n})\} + D_{lsq}\sum_{m=1}^{N-I}\left|G_{n,m}\right|, \end{aligned} \tag{A.3}$$

where $G_{n,m}$'s are given by rearranging $g_{n,m}$'s in descending order of their absolute values, i.e., $|G_{n,1}| \leq |G_{n,2}| \leq \cdots \leq |G_{n,N}|$. D_{lsq} is a constant parameter, and I is a maximum indegree. The maximum indegree determines the maximum number of genes that affect the n-th gene directly.

The first and the second terms of the objective function (A.3) are identical to those of our objective function (6). The third term is a penalty term that forces most of $g_{n,m}$'s down to zero. When this term is applied, therefore, most of the genes are disconnected from each other. The term does not penalize, however, when the number of genes that directly affect the n-th gene is lower than the maximum indegree I. Similar terms have been used in several genetic network inference methods [11], [12], [16].

In this study, we compared the proposed method with a method that minimizes the objective function (A.3). This study refers to the method of optimizing this function as the least-squares approach. As with the proposed method, the least-squares approach also uses REX^{star}/JGG [15] as a function optimizer. The following values were used for the parameters of REX^{star}/JGG applied in the least-squares approach; the population size n_p is $20s$, the number of children generated per selection n_c is $3s$, and the step-size parameter t is 2.5, where s is the dimension of the search space. Note that, when we try to infer a genetic network consisting of N genes, the dimension s equals $N + 3$. Each run was continued until the number of generations reached 1.0×10^5 or the objective value of the best individual contained in the population did not improve over 5,000 generations.

Shuhei Kimura received his M.E. degree from Kyoto University in 1998 and Ph.D. degree from Tokyo Institute of Technology in 2001. He had been in RIKEN Genomic Sciences Center as a research scientist since 2001. In 2004, he moved to Tottori University as an associate professor. Since 2014, he has been a professor at Tottori University. His current research interests are evolutionary algorithms and bioinformatics.

Masanao Sato was born in 1976. He received his M.S. and Ph. D. from Hokkaido University in 2001 and 2004, respectively. He was a postdoctoral associate in University of Minnesota and the University of Tokyo from 2004 to 2009 and was partially supported by JSPS Research Fellowship for Young Scientists and Postdoctoral Fellowship for Research Abroad during this period. His research interest is modeling and rationally redesigning biological networks.

Mariko Okada-Hatakeyama received her Ph.D. from Tokyo University of Agriculture and Technology. She is the team leader of the Laboratory for Integrated Cellular Systems in RIKEN Center for Integrative Medical Sciences (IMS), Japan. Her major research is aimed for the experimental and computational analysis of the signal transduction and transcriptional network for determination of cancer and immune cell development.

(Communicated by *Takeshi Obayashi*)