BIOINFORMATICS ORIGINAL PAPER Vol. 22 no. 13 2006, pages 1631–1640
doi:10.1093/bioinformatics/btl122

*Systems biology*

# Identification of biochemical networks by S-tree based genetic programming

Dong-Yeon Cho[1], Kwang-Hyun Cho[2,3,*] and Byoung-Tak Zhang[1,3,*]

[1]School of Computer Science and Engineering, Seoul National University, Seoul 151-742, Korea, [2]College of Medicine, Seoul National University, Seoul 110-799, Korea and [3]Bio-MAX Institute, Seoul National University, Seoul 151-818, Korea

Received on April 26, 2005; revised on March 3, 2006; accepted on March 25, 2006
Advance Access publication April 3, 2006
Associate Editor: Nikolaus Rajewsky

## ABSTRACT

**Motivation:** Most previous approaches to model biochemical networks have focused either on the characterization of a network structure with a number of components or on the estimation of kinetic parameters of a network with a relatively small number of components. For system-level understanding, however, we should examine both the interactions among the components and the dynamic behaviors of the components. A key obstacle to this simultaneous identification of the structure and parameters is the lack of data compared with the relatively large number of parameters to be estimated. Hence, there are many plausible networks for the given data, but most of them are not likely to exist in the real system.

**Results:** We propose a new representation named S-trees for both the structural and dynamical modeling of a biochemical network within a unified scheme. We further present S-tree based genetic programming to identify the structure of a biochemical network and to estimate the corresponding parameter values at the same time. While other evolutionary algorithms require additional techniques for sparse structure identification, our approach can automatically assemble the sparse primitives of a biochemical network in an efficient way. We evaluate our algorithm on the dynamic profiles of an artificial genetic network. In 20 trials for four settings, we obtain the true structure and their relative squared errors are <5% regardless of releasing constraints about structural sparseness. In addition, we confirm that the proposed algorithm is robust within ±10% noise ratio. Furthermore, the proposed approach ensures a reasonable estimate of a real yeast fermentation pathway. The comparatively less important connections with non-zero parameters can be detected even though their orders are below $10^{-2}$. To demonstrate the usefulness of the proposed algorithm for real experimental biological data, we provide an additional example on the transcriptional network of SOS response to DNA damage in *Escherichia coli*. We confirm that the proposed algorithm can successfully identify the true structure except only one relation.

**Availability:** The executable program and data are available from the authors upon request.

**Contact:** ckh-sb@snu.ac.kr or btzhang@snu.ac.kr

## 1 INTRODUCTION

Recent progress in the development of new technologies has made it possible to take quantitative time-series measurements in an efficient way and thereby to reconstruct a large-scale biochemical network based on high-throughput measurements. We can further convert identified networks into mathematical models by which we can study the dynamic behavior and retrieve information about the interactions of the biochemical network components (Herrgard *et al*., 2004; Papin *et al*., 2003; Schmidt *et al*., 2005; Cho *et al*., 2005a, b). Although the mathematical modeling of biochemical networks can be achieved at different levels of detail [see Covert *et al*. (2001) and De Jong (2002) for a review of metabolic and genetic regulatory network modeling], we can largely classify those into several classes (Stelling, 2004).

One extreme class intends to mainly describe the pattern of interactions among the components. This kind of graph-based modeling provides us with an insight into large architectural features within a cell and allows us to discover the cellular organizing principles (Barabasi and Oltvai, 2004). It is however difficult to deal with the dynamics of a whole cellular system since those models are too abstract for this purpose while there are some recent developments based on probabilistic models to overcome this problem (Segal *et al*., 2003). The other extreme class primarily focuses on describing the dynamics of a system through rigorous mathematical formulations that can explain underlying biochemical interactions (De Jong *et al*., 2004; Rao *et al*., 2002). While this approach can lead to more accurate and quantitative modeling on cellular dynamics, the application is limited to small-scale systems owing to their overwhelming computational complexities.

To understand an organism at the system level we should examine both the interactions among the components and the dynamic behaviors of the components (Wolkenhauer *et al*., 2003). For both structural and dynamical modeling in one unified framework, we propose in this paper a new representation called S-trees. S-trees incorporate not only direct mapping onto a network structure, but also the transformation of data into a set of nonlinear differential equations describing the underlying dynamic behavior of the given time-course data. Here, the favorable data types are time-dense profiles of a limited number of genes, proteins, or metabolites rather than a few snapshots of thousands of components. The proposed S-tree can be regarded as an efficient representation of the S-system (Voit, 2000) which has been known as a good mathematical formalism to represent/analyze biochemical reactions such as metabolic pathways (Vera *et al*., 2003) and genetic networks (Kimura *et al*., 2005). Hence, S-tree representations is general enough to cover various types of biochemical networks.

*To whom correspondence should be addressed.

 

We further present S-tree based genetic programming (GP) to identify the structure of a biochemical network and to estimate the corresponding parameter values at the same time. As this algorithm has the advantage of automatically assembling the sparse primitives of a biochemical network, the proposed approach has the potential to identify the underlying structure in an efficient way. While the GP has already been used to estimate the parameters of the S-system (Ando *et al*., 2002), it transpires to require very complex trees to obtain reasonable parameter estimates of real numbers owing to the limited expression power of terminal nodes. The GP has been also employed to evolve the mathematical expressions (Sugimoto *et al*., 2005) such as differential equations (Ando *et al*., 2002) of biological systems from their simulated dynamic profiles. Although these conventional GPs can successfully reconstruct the given time-series data, it is still difficult to infer the relationships among the involved components from their irregular expressions. Moreover, there can be multiple plausible trees resulting in almost the same profiles with the given profiles even for a slightly large system. On the other hand, S-tree has a regular functional form which can shrink the structural search space. It is general enough to describe various types of non-linear dynamics in an efficient way (for a sparse network structure in particular). Hence, the S-tree representation is more advantageous in its expression powers although the traditional kinetic models (e.g. the Michaelis–Menten equation) and the neural network models have been widely used for modeling of the transcriptional regulations and parameter learning. Separation of the topology search and parameter optimization of S-systems was proposed (Spieth *et al*., 2004), but in this case the good estimates in one generation are no longer passed down to the next generation since this scheme restarts the local optimum selection for a given topology in every generation. We therefore propose a new algorithm to identify the structure as well as to estimate the parameter values in a more efficient way based on the S-tree representation.

Although some other evolutionary search techniques have been proposed (Kikuchi *et al*., 2003; Tominaga *et al*., 2000) for identification of a biochemical network represented by the S-system, they all require additional methods to resolve the problem of inferring an ill-posed structure because of the relative lack of data compared to a large number of parameters to be estimated. For instance, a structural skeletalizing procedure is commonly employed to resolve such a problem by setting some parameters of less than a given threshold to zero, which is because biochemical networks are in most cases sparsely connected (Jeong *et al*., 2000; Thieffry *et al*., 1998). A penalty term for a complex structure has also been added to the fitness function to evolve simplified structures. However, some true connections exhibiting rather small effects can be deleted during the skeletalizing process. Moreover, it is difficult to set a suitable value for the coefficient of the penalty term. Stochastic ranking (Runarsson and Yao, 2000) can be used to alleviate such difficulty since it aims to balance the objective (error) and penalty term in the fitness function. One obvious drawback of this method is however that it requires an additional parameter which defines the probability of an objective term for comparisons in ranking. On the other hand, note that we do not need to append the complexity term to the fitness function in the proposed framework since we represent a sparse biochemical network as an S-tree structure with the network parameter values.

The robustness and the effectiveness of the proposed algorithm are illustrated by three examples: an artificial genetic network, the yeast fermentation pathway and SOS DNA repair system in *E.coli*. Experimental results reveal that the proposed S-tree based GP can successfully identify the true sparse structure and can reasonably estimate the parameter values for the artificial genetic network in a robust way (i.e. regardless of releasing constraints on the structural sparseness). Furthermore, the proposed approach ensures a reasonable estimate of the real biochemical network by efficiently assembling its sparse substructures.

## 2 SYSTEMS AND METHODS

### 2.1 S-tree representation

If the given system consists of $n$ dependent variables whose concentrations $X_i$ change dynamically and $m$ independent variables whose concentrations remain constant over the process evolution, the dynamics of the system can be expressed by the following set of differential equations (for $i = 1, 2, \ldots, n$):

$$\frac{\mathrm{d}X_i}{\mathrm{d}t} = \alpha_i G_i(X_1, \ldots, X_{n+m}; g_{i1}, \ldots, g_{in+m}) - \beta_i H_i(X_1, \ldots, X_{n+m}; h_{i1}, \ldots, h_{in+m}),$$

where the positive parameters $\alpha_i$ and $\beta_i$ are rate constants. These differential equations are composed of two components. The first term denotes an increasing effect on $X_i$ while the second term indicates a decreasing effect. Here, the real-valued parameters $g_{ij}$ and $h_{ij}$ (for $j = 1, 2, \ldots, n + m$) of functions $G_i$ and $H_i$ represent the interactive influence of $X_j$ on $X_i$. Although these parameterizations of the S-tree functional form seem to be limited in representing the interactions of biochemical network elements, they can actually represent various kinds of interactions including reversible reactions, pathways with branch points, single synthesis reactions involving more than two source components, etc. like the S-systems (Voit, 2000).

For identification of the underlying biochemical network from given time-course data of a system, we need to estimate at least $2n(n + 1)$ parameters ($\alpha_i$, $\beta_i$, $g_{ij}$, $h_{ij}$ for $i, j = 1, 2, \ldots, n$) even if the values related to the independent variables are assumed to be known. In many cases, we can however assign zeros to a large part of the parameters $g_{ij}$ and $h_{ij}$ since most of the biological networks are sparse. To effectively reflect this practical situation, we propose an S-tree representation as illustrated in Figure 1a. The depth of this S-tree is always fixed to three and the root node at depth zero always has $n$ subtrees that correspond to the ordinary differential equations of $X_i$. Each subtree is then further divided into two parts where the left and right ones represent the first and second terms of the equation, respectively. Terminal nodes denote the effectual elements and the strength of these. In other words, only the non-zero values of $g_{ij}$ and $h_{ij}$ appear at the bottom of an S-tree. If we adopt the power-law formalism for functions $G_i$ and $H_i$, S-trees provide a compact representation (Fig. 1b) of the general S-system (Fig. 1c) and their full-matrix representation (Fig. 1d).

For identification of a biochemical network represented by an S-tree, we propose a new evolutionary search technique which is to be explained in the following section more in detail. Evolutionary computation has been widely used from its inception for automatic identification of a given system or process based on the measured data (Fogel, 1991). As one of its variants, the GP has an evolving tree structure for given data (Koza, 1992), which is appropriate for our purpose. While the individual tree in conventional GP has an irregular depth and structure, our algorithm automatically assembles the sparse primitives of a biochemical network by exchanging the fixed-depth subtrees in S-trees. It can also estimate the parameter values by using the Gaussian mutation-based hill-climbing optimization.

### 2.2 S-tree based genetic programming

The overall procedures of the proposed S-tree based genetic programming are summarized in Figure 2. As in conventional evolutionary algorithms, we maintain a population $\mathcal{P}(k)$ of individuals $S_i$ (S-trees in this case) at $k$-th

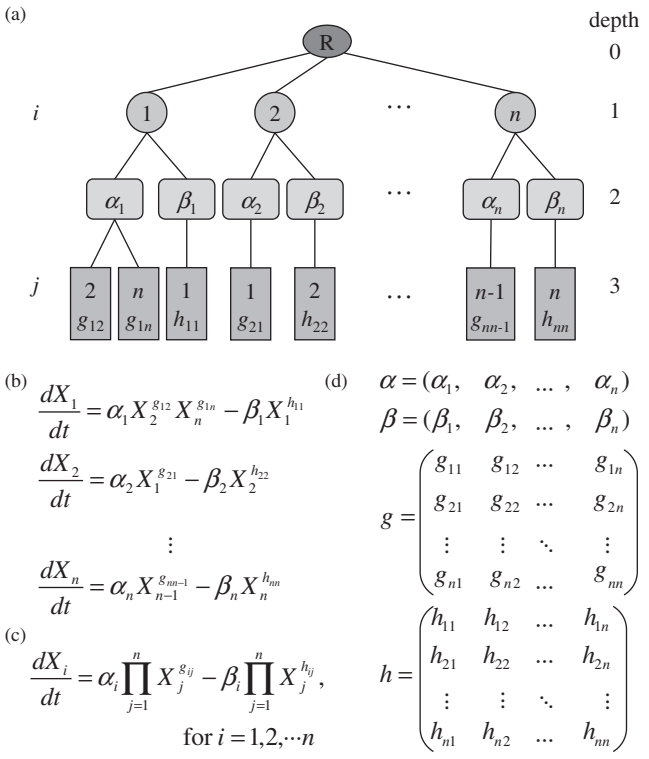


**Fig. 1.** The S-tree representation ($m = 0$). (**a**) S-tree, (**b**) the corresponding compact S-system, (**c**) the general S-system, and (**d**) the full-matrix representation. S-trees can provide compact representation of the general S-system as well as their full-matrix representation since the terminal nodes of S-trees denote only the effectual elements and their strengths in the system.

1. (**Initialization**) Generate an initial population $\mathcal{P}(0)$ randomly. Set iteration count $k \leftarrow 0$.
2. (**Evaluation**) For each S-tree in the population, calculate the fitness by using Equation (1).
3. (**Variation**) Tow randomly chosen parents $S_p$ and $S_q$, produce two offspring through the following steps:
   - (Crossover) Swap two corresponding subtrees of the parents by Equation (2) or (3) as shown in Figure 3.
   - (Mutation) Apply insertion, deletion, or replacement operators to the offspring $S'_p$ and $S'_q$ according to Equation (4) as shown in Figure 4.
   - (Evaluation) Calculate the fitness values of two offspring $S''_p$ and $S''_q$ by using Equation (1).
   - (Hill-climbing) Change each parameter value of the offspring just once by Gaussian mutation where each adjustment is accepted only if the fitness value is better than before.
4. (**RTS**) Replace an S-tree in the population with a newly created and most similar offspring only if the new one is better.
5. (**Finish**) Stop if the termination criteria are met.
6. (**Loop**) Set $k \leftarrow k + 1$ and go to Step 3.

**Fig. 2.** The schematic procedures of the S-tree based genetic programming.

iteration as:

$$\mathcal{P}(k) = \{S_1, S_2, \ldots, S_M\},$$

where $M$ is the population size and $S_i$ is defined as follows:

$$S_i = \{(s^1_{i1}, s^2_{i1}), (s^1_{i2}, s^2_{i2}), \ldots, (s^1_{in}, s^2_{in})\},$$

where $(s^1_{ij}, s^2_{ij})$ (for $j = 1, 2, \ldots, n$) represents the subtrees in $S_i$ and $s^1_{ij}$ and $s^2_{ij}$ are the first and the second subtrees, respectively. The initial population$\mathcal{P}(0)$ is randomly generated. This means that S-trees with arbitrary structures are created and their parameter values are also randomly assigned within the search ranges. In this step, a priori biochemical knowledge can be included as structural constraints. For instance, some components can be assumed to be non-zeros if the corresponding biochemical connections are already known to be vital for the living system under consideration.

We can evaluate each S-tree in the population by considering how well it can reproduce the given time-series data. In other words, the fitness $F_i(S_i)$ of each S-tree is defined as a sum of the relative squared errors, which is to be minimized:

$$F_i(S_i) = \sum_{i=1}^{n} \sum_{t=1}^{T} \left( \frac{X_i(t) - \hat{X}_i(t)}{X_i(t)} \right)^2, \quad (1)$$

where $T$ is the number of sampling points in the given time-course data for each $X_i$, $X_i(t)$ is the experimentally measured biochemical quantity (e.g. concentration) of $X_i$ at the sampling time $t$, and $\hat{X}_i(t)$ is the estimated biochemical quantity obtained by the numerical integration of the corresponding differential equations represented by $S_i$. Note here that we do not use the penalty term to enforce the sparse solutions. Instead, we limit the model complexity by the structural constraints based on biological a priori knowledge. For instance, the cascade constraint induces $n - 1$ terminal nodes ($g_{ij} \neq 0$ if $j = i - 1$ for $i = 2, \ldots, n$). There should be another $n$ terminal nodes considering all of the diagonal elements in $h$ ($h_{ii} \neq 0$ for $i = 1, \ldots, n$). The number of the other terminal nodes is parameterized and limited by $b_{\max}$ which actually means the maximum number of feedback/feedforward loops in the system. In preliminary studies, we had set this to 1 or 2 for the sparse candidates and then gradually increased this until we had a satisfactory result. Although not considered in this paper, the adaptive learning method by automatic balancing of model-complexity factor (Zhang and Mühlenbein, 1995) can be useful when we have little a priori information on the system.

Two major variation operators are applied to a pair of randomly selected parent S-trees ($S_p$ and $S_q$) in order to generate an offspring. First, a crossover operator swaps two corresponding subtrees chosen from the parents. The crossover that takes place at depth one results in the exchange of two corresponding differential equations of the S-system (Fig. 3) whereas that at depth two results in the trade of the first or second term of the corresponding equations. If the $c$-th subtree is selected, the offspring $S'_p$ and $S'_q$ from the crossover at depth one are denoted as follows:

$$(S'_p, S'_q) = \text{Crossover1}(S_p, S_q) \quad (2)$$

with

$$S'_p = \{(s^1_{p1}, s^2_{p1}), \ \ldots, (s^1_{qc}, s^2_{qc}), \ \ldots, (s^1_{pn}, s^2_{pn})\},$$
$$S'_q = \{(s^1_{q1}, s^2_{q1}), \ \ldots, (s^1_{pc}, s^2_{pc}), \ \ldots, (s^1_{qn}, s^2_{qn})\},$$

On the other hand, the crossover at depth two produces the following offspring:

$$(S'_p, S'_q) = \text{Crossover2}(S_p, S_q) \quad (3)$$

with

$$S'_p = \{(s^1_{p1}, s^2_{p1}), \ \ldots, (s^1_{qc}, s^2_{pc}), \ \ldots, (s^1_{pn}, s^2_{pn})\},$$
$$S'_q = \{(s^1_{q1}, s^2_{q1}), \ \ldots, (s^1_{pc}, s^2_{qc}), \ \ldots, (s^1_{qn}, s^2_{qn})\},$$

or

$$S'_p = \{(s^1_{p1}, s^2_{p1}), \ \ldots, (s^1_{pc}, s^2_{qc}), \ \ldots, (s^1_{pn}, s^2_{pn})\},$$
$$S'_q = \{(s^1_{q1}, s^2_{q1}), \ \ldots, (s^1_{qc}, s^2_{pc}), \ \ldots, (s^1_{qn}, s^2_{qn})\},$$

This kind of crossover operation can pass the good structures as well as the corresponding parameter values from the parents to the offspring. Second, we mutate the offspring $S'_p$ and $S'_q$ (Fig. 4) by randomly deleting a terminal

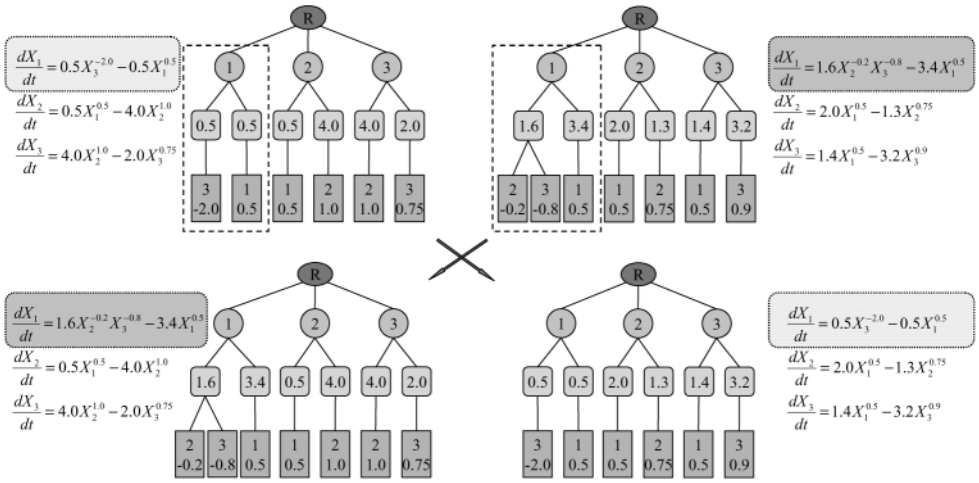

**Fig. 3.** An example of the crossover at depth one ($n = 3$, $m = 0$). Two subtrees are randomly chosen from the parents and then swapped with each other, which results in the exchange of the corresponding differential equations in the S-system. In this way, the good structures can be passed onto the offspring together with their parameter values.

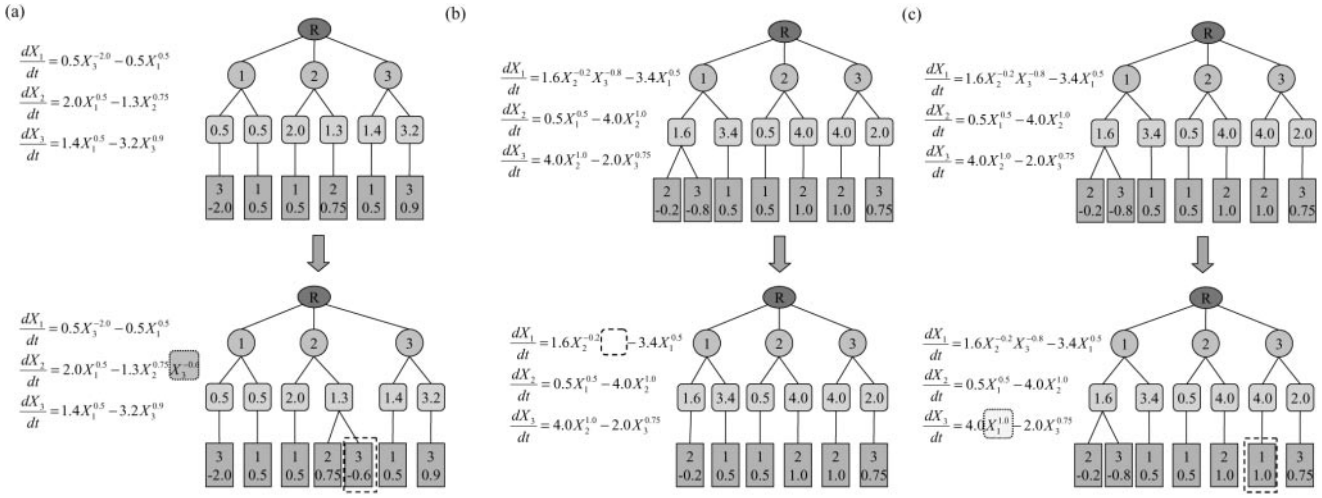

**Fig. 4.** Examples of the mutation operators such as (**a**) insertion of a new terminal node (**b**) deletion of a randomly chosen terminal node and (**c**) replacement of one index of a terminal node with another one ($n = 3, m = 0$). These operators are limited to modify the structure of S-trees by changing only one terminal node at a time. Hence, these can be regarded as local searches in the overall structure space.

node and inserting a new one, or by replacing one index of a terminal node with another one:

$$(S_p'') = \text{Mutation}\ (S_p') \quad \text{and} \quad (S_q'') = \text{Mutation}\ (S_q'), \tag{4}$$

where Mutation $\in$ {Insertion, Deletion, Replacement}. Conceptually, these operators perform local searches in the structure space. The fitness value of an offspring created by the crossover and mutation (according to their probabilities) is evaluated by Equation (1).

In general, the embedded search process for numeric constants is helpful in improving the performance of GP (Sugimoto *et al*., 2005), particularly for the regression problems. For this purpose, Gaussian mutations have been introduced since they can generate more effective searches than the crossover and mutation operations that have been commonly used in genetic algorithms (Fogel, 1990). In our problems, however, the Gaussian mutations require a large computational cost due to the numerical integrations for the fitness evaluation in every searching step. To overcome this difficulty, we have employed the Gaussian mutation-based hill-climbing search while acknowledging the possibility of a local optimum and thereby resulting in sometimes poorer performance than that of the traditional Gaussian mutation. Parameters $\alpha_i$, $\beta_i$, $g_{ij}(\neq 0)$, $h_{ij}(\neq 0)$ for all $i, j$ in the offspring $S_p''$ and $S_q''$ are changed just once by Gaussian mutation, i.e. a random number sampled from the standard normal distribution is added. These changes are accepted only if the resulting fitness value is better than before (i.e. if the estimation error is reduced). This kind of tree-based evolutionary algorithm for both structure search and parameter estimation has been successfully applied to learning the neural trees for a given time-series data (Zhang *et al*., 1997).

We employ here the restricted tournament selection (RTS) originally proposed in Harik (1995) to prevent the premature convergence on a local-optimum structure and to make a pool of multiple topology candidates without separating the search procedure for the structure and that for the parameter estimates as was done in Morishita *et al*. (2003). In the RTS, a subset of the current population is selected for each newly created offspring. The size of these subsets is fixed to some constant called an window size. Then the new offspring competes with the most similar member of the

subset. Since the window size is set to the population size in our proposed implementation procedure, each offspring is compared with all of the S-trees in the population. If the new one is better then it replaces the corresponding individual; otherwise, the new one is discarded. For the similarity measure, we calculate the structural Hamming distances between the new offspring and each of the individuals in the population by using the binary representation of the matrices $g$ and $h$.

Although not considered in this paper, decoupling the dynamic processes by estimating the slopes (Voit and Almeida, 2004) and applying the modified collocation approximation (Tsai and Wang, 2005) or decomposing the large network inference problem into subproblems (Kimura *et al*., 2005; Maki *et al*., 2002) can be easily incorporated into the proposed scheme to avoid the computationally expensive numerical integrations for the fitness evaluations. These techniques are helpful in increasing the size of biochemical networks to which the proposed method can be successfully applicable since they provide the separate and parallel analysis of each subtree in S-tree representations.

## 3 RESULTS

### 3.1 Artificial genetic network

To evaluate the performance of the proposed algorithm, we first consider an artificial genetic network employed from previous studies (Ando *et al*., 2002; Kikuchi *et al*., 2003; Kimura *et al*., 2005; Spieth *et al*., 2004). As we have to estimate a relatively large number of parameters with an insufficient dataset, we suffer from a dimensionality problem and there can be a lot of different possible network structures all of which bring about only small differences in estimating the given dataset. These false candidates can be reduced by shrinking the structural search space based on available constraints. One obvious constraint is that all the diagonal elements in the matrix $h$ are not zero ($h_{ii} \neq 0$ for $i = 1, \ldots, n$). This is because as the concentration $X_i$ is higher, $X_i$ can participate in the reaction more actively (i.e. it disappears fast). With the assumption that it is known to be a cascade system (there is usually a nominal cascade signaling pathway and more complicated feedback or feedforward loops are formed around the nominal pathway), we can impose another structural constraint on the initial population: $g_{ij} \neq 0$ if $j = i - 1$, for $i = 2, \ldots, n$. This means that every left side subtree in the $i$-th subtree has a terminal node whose index $j$ is equal to $i - 1$ except for the first subtree. The maximum number of terminal nodes (which are not caused by the structural constraints) is limited to $b_{\max}$. The individuals in the initial population are uniformly distributed over the structural search space since they are randomly generated within the structural constraints. In other words, the additional terminal nodes in every initial S-tree are equally probable with respect to the possible values $\{1, 2, \ldots, b_{\max}\}$.

To generate artificial data sets that will act as experimentally measured metabolic profiles, we apply the fourth-order Runge-Kutta method to the assumed true S-system (Fig. 5b). The initial conditions for the data generation are as follows: $n = 5$, $T = 15$, $X_1(0) = 0.7, X_2(0) = 0.12, X_3(0) = 0.14, X_4(0) = 0.16, X_5(0) = 0.18$. The size of population is assumed as 10 000 and therefore ten thousands of different S-trees are randomly generated to satisfy the structural constraints. The crossover and the mutation probabilities are set to 0.9 and 0.6, respectively. The search ranges of the parameters are set as [0.0, 15.0] for $\alpha_i$ and $\beta_i$, and [−3.0, 3.0] for $g_{ij}$ and $h_{ij}$. These search ranges were set identical to those of the previous study (Kikuchi *et al*., 2003) for a fair comparison. The proposed scheme is terminated after $5 \times 10^5$ iterations. All the computational experiments in this paper were performed in the Linux system with AMD Athlon MP 2800+ processor and 2 GB memory.

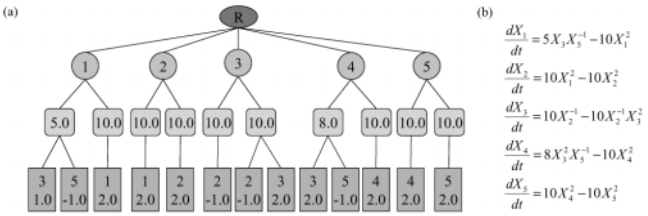


**Fig. 5.** (**a**) The true S-tree representation for the artificial genetic network ($n = 5, m = 0$), and (**b**) the corresponding S-system. By using this true network, we generate the dynamic profiles which resemble the experimentally measured data.

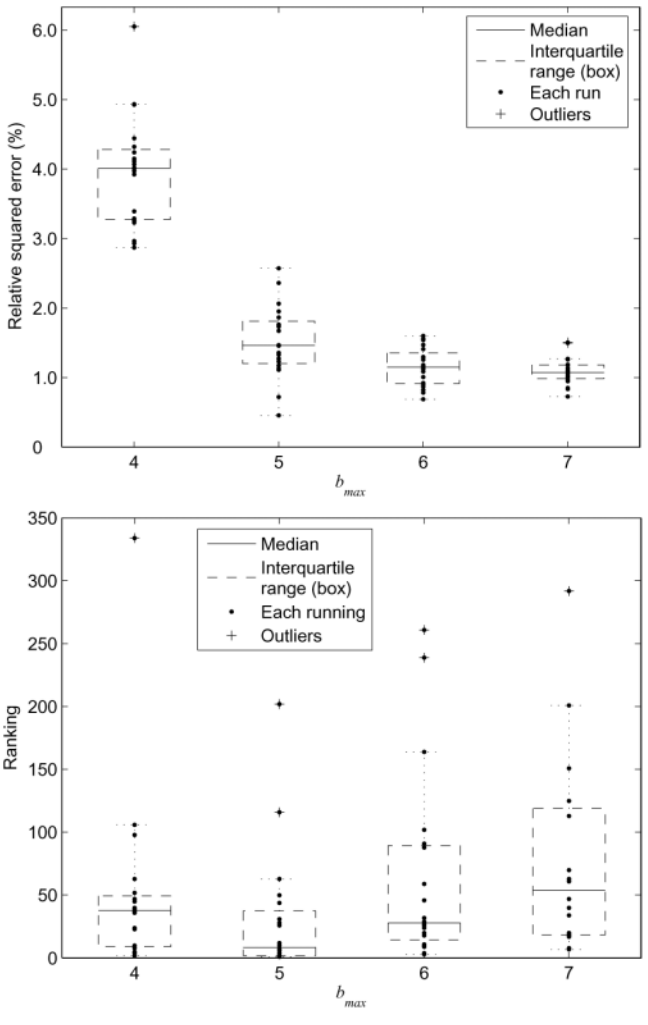


**Fig. 6.** Errors and rankings of the individuals that have the true structure over 20 independent runs for each value of $b_{\max}$ which is the maximum number of terminal nodes resulting from the structural constraints. The relative squared errors decrease on average. However, the rankings according to the fitness values over the final population do not show significant increase even though the number of complex candidates grows exponentially.

**Table 1.** True versus estimated parameter values for the artificial genetic network

| $i$ | $\alpha_i/\hat{\alpha}_i$ | $g_{i1}/\hat{g}_{i1}$ | $g_{i2}/\hat{g}_{i2}$ | $g_{i3}/\hat{g}_{i3}$ | $g_{i4}/\hat{g}_{i4}$ | $g_{i5}/\hat{g}_{i5}$ |
|---|---|---|---|---|---|---|
| 1 | 5.0/8.5854 | 0.0/0.0000 | 0.0/0.0000 | 1.0/0.5886 | 0.0/0.0000 | −1.0/−0.7083 |
| 2 | 10.0/9.7709 | 2.0/2.1013 | 0.0/0.0000 | 0.0/0.0000 | 0.0/0.0000 | 0.0/0.0000 |
| 3 | 10.0/13.7629 | 0.0/0.0000 | −1.0/−0.8140 | 0.0/0.0000 | 0.0/0.0000 | 0.0/0.0000 |
| 4 | 8.0/8.3954 | 0.0/0.0000 | 0.0/0.0000 | 2.0/1.9352 | 0.0/0.0000 | −1.0/−0.9917 |
| 5 | 10.0/9.4643 | 0.0/0.0000 | 0.0/0.0000 | 0.0/0.0000 | 2.0/1.9546 | 0.0/0.0000 |
| | $\beta_i/\hat{\beta}_i$ | $h_{i1}/\hat{h}_{i1}$ | $h_{i2}/\hat{h}_{i2}$ | $h_{i3}/\hat{h}_{i3}$ | $h_{i4}/\hat{h}_{i4}$ | $h_{i5}/\hat{h}_{i5}$ |
| 1 | 10.0/13.7959 | 2.0/1.3778 | 0.0/0.0000 | 0.0/0.0000 | 0.0/0.0000 | 0.0/0.0000 |
| 2 | 10.0/10.0117 | 0.0/0.0000 | 2.0/2.2341 | 0.0/0.0000 | 0.0/0.0000 | 0.0/0.0000 |
| 3 | 10.0/13.9742 | 0.0/0.0000 | −1.0/−0.8120 | 2.0/1.9543 | 0.0/0.0000 | 0.0/0.0000 |
| 4 | 10.0/10.1264 | 0.0/0.0000 | 0.0/0.0000 | 0.0/0.0000 | 2.0/2.1038 | 0.0/0.0000 |
| 5 | 10.0/9.7222 | 0.0/0.0000 | 0.0/0.0000 | 0.0/0.0000 | 0.0/0.0000 | 2.0/2.1300 |

The proposed algorithm has successfully identified the true structure of the original system in Figure 5 and estimated the parameter values within a fairly small error range.

By releasing the limitation for the maximum number of terminal nodes except for nodes caused by structural constraints (i.e. allowing more complex candidate topologies), we can test whether the proposed scheme can identify the true structure. The results from 20 independent executions for each value of $b_{\max}$ are summarized in Figure 6. Note that, in all experiments, we obtain the true structure without the threshold or the coefficient for the sparse network. That is, since S-trees can intrinsically represent the sparse structures, we do not need to append the complexity term to the fitness function for enforcing parsimony. In addition, the relative squared errors decrease on average. Furthermore, the rankings according to the fitness values over the individuals in the final population do not significantly increase in spite of the number of complex candidates, which grows exponentially. This implies that the proposed scheme is not only able to assemble good primitive structures by recombination of the promising substructures but is also able to well preserve the sparse structures. Table 1 shows the best parameter estimates obtained among all the experiments. Throughout simulation with these parameter estimates, we further confirm that the identified system recovers most of the true dynamics of the original system (Fig. 7).

To test the noise-tolerance of the proposed algorithm, we have added ±1, 2, 5, 10 and 15% uniformly distributed random noise to the true synthetic data. Except that $b_{\max}$ is fixed at 5, other settings are same as in the previous experiments. The execution time for each run is ∼7.3 h. As shown in Figure 8, the best and the median rankings of the individuals having the true structure over 20 independent runs are similar up to the 10% noise ratio while the worst rankings grow slowly for 5 and 10% noise. For 15% ratio, however, we observe that the rankings are scattered over 500 with a sudden rise in the worst case. Hence, we can conclude that the proposed algorithm is robust within ±10% random noise.

## 3.2 Yeast fermentation pathway

As an example of real biochemical networks, we consider an S-system model of the ethanol production by yeast (*Saccharomyces cerevisiae*) which has been extensively studied for both practical and academic reasons (Voit, 2000). Among the numerous previous investigations under different experimental settings and conditions, we choose the following system employed from (Vera *et al*., 2003):

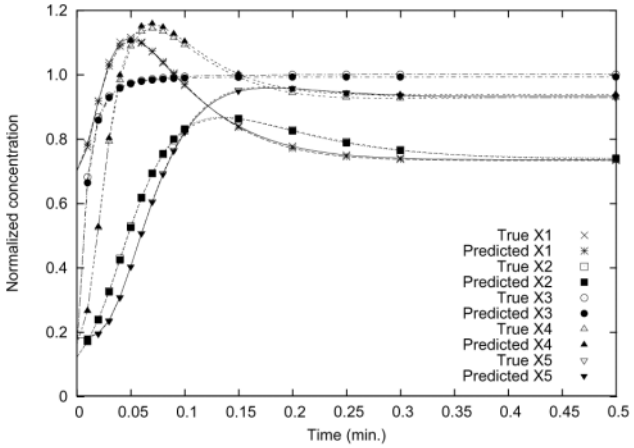


**Fig. 7.** The original versus simulated time-series data for the artificial genetic network. The identified system successfully recovers most of the true dynamics of the original system.

$$
\begin{aligned}
\frac{dX_1}{dt} &= 1.0006X_2^{-0.0492}X_6 - 1.6497X_1^{0.5582}X_5^{0.0465}X_7 \\
\frac{dX_2}{dt} &= 1.6497X_1^{0.5582}X_5^{0.0465}X_7 \\
&\quad - 0.5793X_2^{0.5097}X_5^{-0.2218}X_8^{0.8322}X_{11}^{0.1678} \\
\frac{dX_3}{dt} &= 0.4536X_2^{0.4407}X_5^{-0.2665}X_8 \\
&\quad - 0.2456X_3^{0.4506}X_4^{0.0441}X_5^{0.092}X_9^{0.8547}X_{12}^{0.1453} \\
\frac{dX_4}{dt} &= 0.2365X_3^{0.5285}X_5^{0.0994}X_9 \\
&\quad - 2.0892X_3^{-0.0075}X_4^{0.304}X_5^{0.0484}X_{10} \\
\frac{dX_5}{dt} &= 1.406X_3^{0.2605}X_4^{0.152}X_5^{0.0739}X_9^{0.5}X_{10}^{0.5} \\
&\quad - 2.9437X_1^{0.1962}X_2^{0.1791}X_5^{0.2354}X_7^{0.3514}X_8^{0.2925} \\
&\quad \times X_{11}^{0.0589}X_{13}^{0.297}.
\end{aligned}
$$

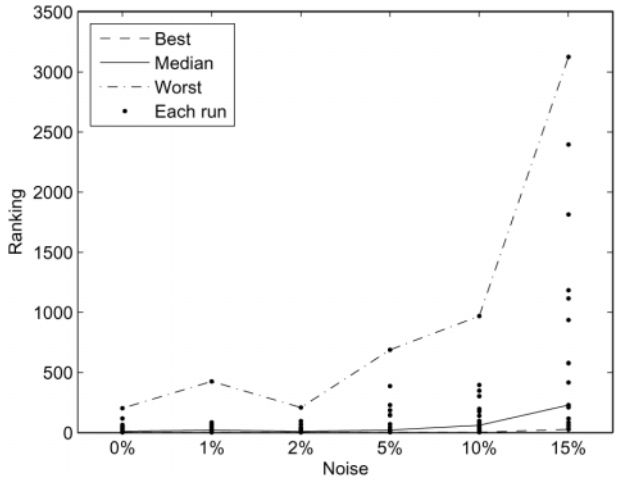


**Fig. 8.** The change of the rankings of the individuals that have the true structure in the final population as the noise ratio increases. In all experiments, we have set the maximum number of terminal nodes, $b_{\max}$ to 5. From this result, we can confirm that the proposed algorithm is robust within ±10% noise ratio.

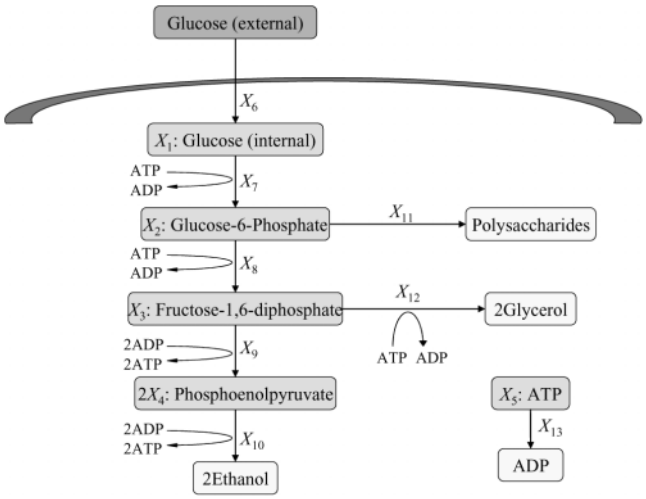


**Fig. 9.** Illustration of the anaerobic fermentation pathway in an yeast (Vera *et al*., 2003). This model refers to anaerobic and nongrowing conditions with glucose as the sole carbon source in the absence of nitrogen. Five dependent and eight independent variables are involved in this system.

This model refers to anaerobic, non-growing conditions, with glucose as the sole carbon source and in the absence of nitrogen. As shown in Figure 9, this metabolic pathway involves five dependent variables: glucose ($X_1$), glucose-6-phosphate ($X_2$), fructose-1,6-diphosphate ($X_3$), phosphoenolpyruvate ($X_4$) and ATP ($X_5$). The model also contains eight independent variables with the following steady-state values: glucose uptake ($X_6$; 47.5 mM min$^{-1}$), hexokinase ($X_7$; 24.1 mM min$^{-1}$), phosphofructokinase ($X_8$; 53.9 mM min$^{-1}$), glyceraldehyde 3-phosphate dehydrogenase ($X_9$; 91.4 mM min$^{-1}$), pyruvate kinase ($X_{10}$; 18.1 mM min$^{-1}$), polysaccharide storage ($X_{11}$; 82.9 mM min$^{-1}$), glycerol production ($X_{12}$; 92.4 mM min$^{-1}$) and ATPase ($X_{13}$; 1.0 mM min$^{-1}$). Here, we only consider the parameter values related to the dependent variables with the assumption that the enzyme activities (the parameter values for independent variables) are known.

Upon the two structural constraints from the previous example, we add one more constraint regarding the ATP conversion, which is involved in every reaction step except glucose uptake. This means that $g_{i5}$ for $i = 2, \ldots, n$ and $h_{i5}$ for $i = 1, \ldots, n-1$ should not be zero. Hence, by randomly generating the initial population, we append the terminal nodes of indices 5 to every S-tree for $g_{25}, \ldots, g_{55}$ and $h_{15}, \ldots, h_{45}$. A total of 10 sets of time-course data ($T = 20$) are generated in the same way as for the previous experiment with 10 arbitrary initial concentrations whose ranges are [0.0, 10.0]. All experimental settings for identification of this yeast fermentation pathway are similar to that of the previous artificial genetic network except that the maximum number of iterations is $10^6$ and $b_{\max}$ is set to 6 in this case. The search ranges of the parameters are [0.0, 3.0] for $\alpha_i$ and $\beta_i$, and [−1.0, 1.0] for $g_{ij}$ and $h_{ij}$. These rather compact ranges were chosen to prevent any over-fitting of the given training data when some parameter values rapidly grow, although a few parameter values could be far larger than the others in real stiff problems. The computational time for this experiment is ∼79 h.

Not having the skeletalizing parameters, the proposed scheme can automatically detect the sparse primitives of a biochemical network and identify the underlying structure. This is verified from the fact that the comparatively less important parameters with non-zero values did not disappear as shown in Table 2. We note that these connections can be lost during the explicit skeletalizing procedure in which some kinetic orders near zero (or below a threshold) are set to zero. It is of course difficult to determine the optimal threshold value. If the threshold is too large then the true connections exhibiting only small effects can be missed during the procedure. On the other hand, if it is too small then we cannot retrieve the inherent sparse structure. It is also noteworthy that the proposed S-tree based GP can find feasible solutions without the penalty coefficient for a simplified network structure. In previous studies, this value should be chosen carefully since an improper value can impair the success rate of attaining good candidates for a true network structure (Kikuchi *et al*., 2003).

### 3.3 SOS DNA repair system in *E.coli*

To demonstrate the usefulness of the proposed algorithm for real experimental biological data, we consider the transcriptional network of SOS response to DNA damages in *E.coli* (Sutton *et al*., 2000). About 30 genes are known to be involved in this DNA damage tolerance and repairing process (e.g. recA, lexA, umuD, uvrA, etc.). In a normal state, LexA protein acts as a repressor which binds to the promoter regions of those genes. Once DNA damages take place, RecA proteins sense them and form nucleoprotein filaments by binding to single strand DNAs. They activate the SOS repair system by inducing LexA autocleavage. That is, the decrease of the LexA concentration causes the de-repression of the SOS genes (Fig. 10). As the damage is repaired, LexA begins to be accumulated and then represses the SOS genes again. Four time-course real data of the main eight genes of this system were collected from Ronen *et al*. (2002) based on the measurements using the green fluorescent proteins (GFPs). Every profile (http://www.weizmann.ac.il/mcb/UriAlon/Papers/SOSData/) consists of 50 measurements evenly spaced by 6 min including the initial

**Table 2.** True versus estimated parameter values of the yeast fermentation pathway

| $i$ | $\alpha_i/\hat{\alpha}_i$ | $g_{i1}/\hat{g}_{i1}$ | $g_{i2}/\hat{g}_{i2}$ | $g_{i3}/\hat{g}_{i3}$ | $g_{i4}/\hat{g}_{i4}$ | $g_{i5}/\hat{g}_{i5}$ |
|---|---|---|---|---|---|---|
| 1 | 1.0006/0.6113 | 0.0000/0.0000 | −0.0492/−0.0107 | 0.0000/0.0000 | 0.0000/0.0000 | 0.0000/0.0000 |
| 2 | 1.6497/1.4432 | 0.5582/0.6322 | 0.0000/0.0000 | 0.0000/0.0000 | 0.0000/0.0000 | 0.0465/−0.0363 |
| 3 | 0.4536/0.3689 | 0.0000/0.0000 | 0.4407/0.5313 | 0.0000/0.0000 | 0.0000/0.0000 | −0.2665/−0.3004 |
| 4 | 0.2365/0.1464 | 0.0000/0.0000 | 0.0000/0.0000 | 0.5285/0.9056 | 0.0000/0.0000 | 0.0994/−0.0881 |
| 5 | 1.4060/1.2827 | 0.0000/0.0000 | 0.0000/0.0000 | 0.2605/0.2253 | 0.1520/0.2366 | 0.0739/−0.0514 |
| | $\beta_i/\hat{\beta}_i$ | $h_{i1}/\hat{h}_{i1}$ | $h_{i2}/\hat{h}_{i2}$ | $h_{i3}/\hat{h}_{i3}$ | $h_{i4}/\hat{h}_{i4}$ | $h_{i5}/\hat{h}_{i5}$ |
| 1 | 1.6479/0.9903 | 0.5582/0.7560 | 0.0000/0.0000 | 0.0000/0.0000 | 0.0000/0.0000 | 0.0465/0.0865 |
| 2 | 0.5793/0.4723 | 0.0000/0.0000 | 0.5097/0.6706 | 0.0000/0.0000 | 0.0000/0.0000 | −0.2218/−0.4061 |
| 3 | 0.2456/0.2067 | 0.0000/0.0000 | 0.0000/0.0000 | 0.4506/0.5729 | 0.0441/0.0227 | 0.0920/0.0890 |
| 4 | 2.0892/1.4889 | 0.0000/0.0000 | 0.0000/0.0000 | −0.0075/−0.0342 | 0.3040/0.4647 | 0.0484/−0.0161 |
| 5 | 2.9437/2.1809 | 0.1962/0.3712 | 0.1791/0.0107 | 0.0000/0.0000 | 0.0000/0.0000 | 0.2354/0.3781 |

The proposed algorithm has successfully identified the true structure of the original system while some parameters were estimated with opposite signs.

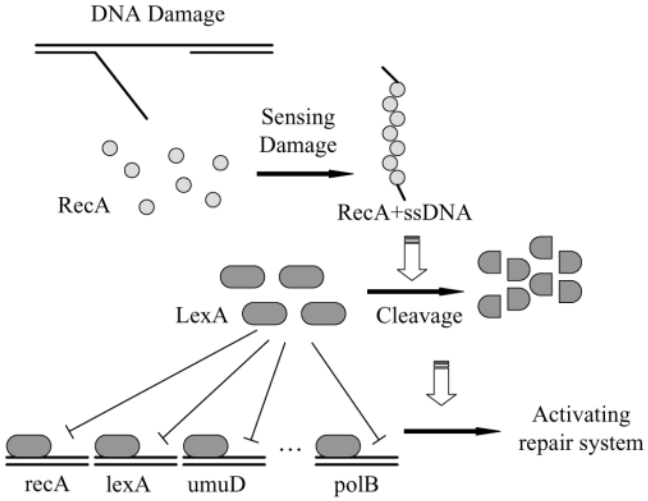


**Fig. 10.** SOS DNA repair system in *E.coli* (Sutton *et al*., 2000). Once DNA damages are detected by RecA proteins, the induced LexA autocleavage makes the system activated.

concentrations which are all zeros. As a real dataset to test the proposed algorithm, we choose six genes (uvrD, lexA, umuD, recA, uvrA and polB) in the fourth dataset.

As in the previous two cases, we impose some constraints on the candidate network structures to shrink the broad structural space. Except the self-regulation ($h_{ii} \neq 0$ for all $i$), we do not pay attention to the other elements in the matrix $h$ (i.e. $h_{ij} = 0$ if $i \neq j$). In addition, each gene is assumed to be related to one or two other genes in the system. From these constraints, the randomly generated initial individuals should have one or two terminal nodes for $g$ and one node for $h$ in each subtree. We also limit the total number of terminal nodes in one S-tree up to 13. The search ranges of the parameters are [0.0, 5.0] for $\alpha_i$ and $\beta_i$, and [−2.0, 2.0] for $g_{ij}$ and $h_{ij}$. We set the same initial concentration for all genes to $10^{-3}$ although the true values are all zeros. This is because the S-system cannot produce any other profile from the original initial condition of all zeros. In this example, we adopt a mean squared error as the fitness value since the relative squared error becomes invalid when the measured concentration is 0.

After $2 \times 10^6$ generations in ∼35 h, the proposed algorithm has successfully identified the inhibition of other genes by lexA although the suppressive effect on recA was missed as shown in Table 3. The activation of recA by uvrA is false since this is a target gene and thus cannot play the role as a regulator. However, the regulation of lexA by recA was detected while their relationship is indirect (recA→RecA ⊣ LecA ⊣ lecA). With the identified parameter values, we have simulated the dynamic behavior of the system (Fig. 11). In spite of the noisy experimental data, we confirm that the general trends of the expression patterns for all genes are rather well recovered.

## 4 CONCLUSIONS

For both structural and dynamical modeling of a biochemical network in one unified framework, we have proposed an S-tree representation that can encompass both direct mapping onto a network structure and transformation of data into a set of dynamic equations. Since S-tree modeling is intrinsically suitable for representing a sparse network, we can address the topological issue of a biochemical network as well as the issue of parameter estimation in this framework. S-tree based GP has thus been presented for identification of a biochemical network. As this algorithm has the advantage of automatically assembling the sparse primitives of a biochemical network, it has the potential to identify the underlying structure in a more efficient way. By applying the proposed technique to the identification of an artificial genetic network based on generated time-course data, we have verified its capability of finding the reasonable parameter estimates as well as unraveling the true sparse structure in a robust way even if we do not have any a priori knowledge about the exact number of underlying feedback loops in a given system. Furthermore, the S-tree based GP could find feasible solutions of the real biochemical network example without the regularization factors such as the threshold (to remove the minor connections) and the coefficient of the penalty term (for the complex

**Table 3.** The estimated parameter values of the SOS DNA repair system in *E.coli* (1: uvrD, 2: lexA, 3: umuD, 4: recA, 5: uvrA, and 6: polB)

| $i$ | $\hat{\alpha}_i$ | $\hat{g}_{i1}$ | $\hat{g}_{i2}$ | $\hat{g}_{i3}$ | $\hat{g}_{i4}$ | $\hat{g}_{i5}$ | $\hat{g}_{i6}$ | $\hat{\beta}_i$ | $\hat{h}_{i1}$ | $\hat{h}_{i2}$ | $\hat{h}_{i3}$ | $\hat{h}_{i4}$ | $\hat{h}_{i5}$ | $\hat{h}_{i6}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 3.0201 | 0.0000 | −1.0090 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 3.1120 | 0.5878 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 2 | 0.0384 | 0.0000 | −0.2271 | 0.0000 | 1.8836 | 0.0000 | 0.0000 | 4.9993 | 0.0000 | 1.1379 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| 3 | 1.2459 | 0.0000 | −1.2592 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.7401 | 0.0000 | 0.0000 | 1.0995 | 0.0000 | 0.0000 | 0.0000 |
| 4 | 3.7721 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 1.2127 | 0.0000 | 0.1042 | 0.0000 | 0.0000 | 0.0000 | 1.6184 | 0.0000 | 0.0000 |
| 5 | 3.3552 | 0.0000 | −1.4562 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.2163 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 1.3217 | 0.0000 |
| 6 | 1.6719 | 0.0000 | −0.9869 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 4.9999 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.4618 |

The proposed algorithm has successfully identified the inhibition of other genes by lexA although the suppressive effect on recA was missed.

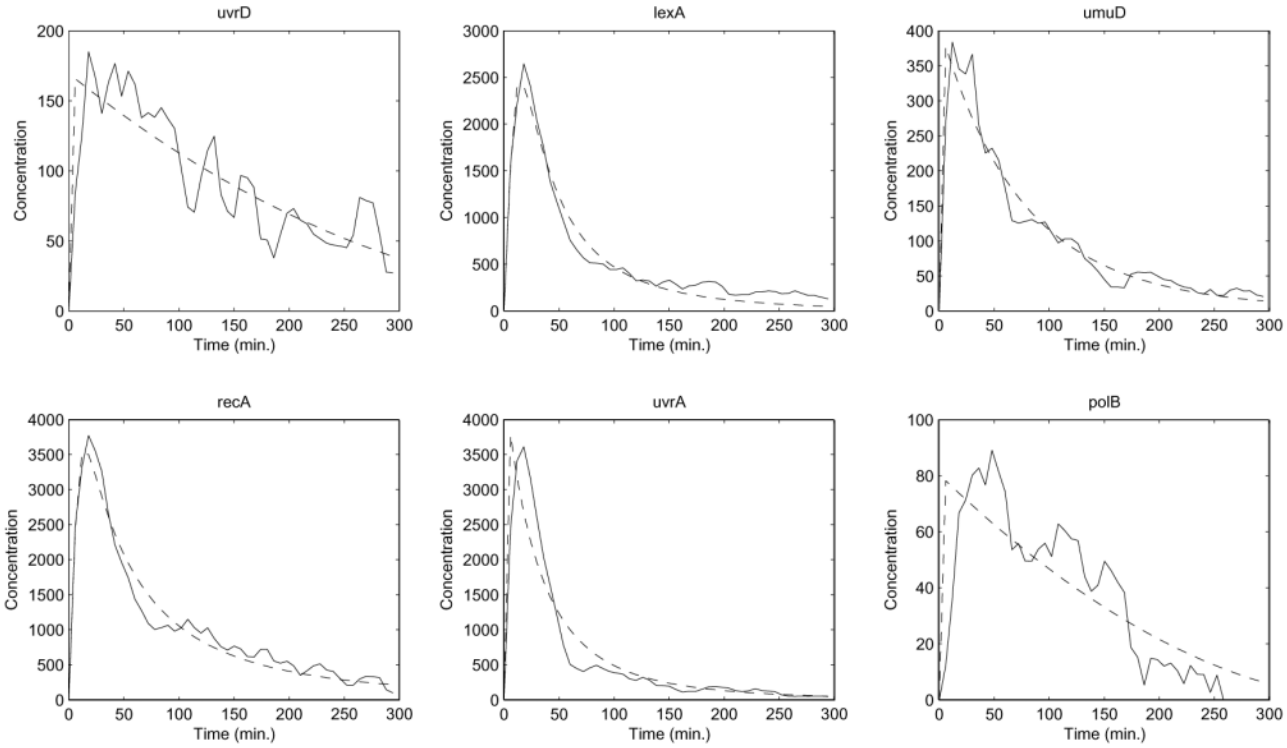


**Fig. 11.** The measured (solid lines) versus simulated (dash lines) time-course profiles for the SOS response to DNA damages in *E.coli*. In spite of the noisy experimental data, the general trends of expression profiles of all genes were rather well recovered.

structures). This means that the proposed algorithm can well identify the network structure by assembling the promising substructures and can also keep the sparse networks within the population in an appropriate way. Besides keeping diversity in the main population, we also note that different plausible network topologies are attained. The highly ranked networks can be good candidates for representing the underlying biochemical network of a living system and thereby provide a good basis for investigating the unknown interactions of the constituent components. One major difficulty in applying the proposed algorithm to real large-scale networks lies in the fitness evaluation as it requires time-consuming numerical integrations over all of the candidates. There are however some techniques available to reduce such computational complexity (see Section 2.2) and these can be easily incorporated into the proposed algorithm to enhance the applicability of the proposed algorithm for more large-scale biochemical networks.

## ACKNOWLEDGEMENTS

This work was supported by grants from the Korea Ministry of Science and Technology (Korean Systems Biology Research Grants, M10503010001-05N030100111 and M10309000002-03B5000-00110), by the 21C Frontier Microbial Genomics and Application Center Program, Ministry of Science and Technology (Grant MG05-0204-3-0), Republic of Korea, and in part by 2005-B0000002 from Korea Bio-Hub Program of Korea Ministry of Commerce, Industry and Energy. D.-Y.C. and B.-T.Z. were also supported by the same Ministry through National Research Lab (NRL) project and the Ministry of Education and Human Resources Development under the BK21-IT Program. The ICT at the Seoul National University provided research facilities for this study.

*Conflict of Interest*: none declared.

## REFERENCES


Ando,S. *et al*. (2002) Evolutionary modeling and inference of gene network. *Inf. Sci*., **145**, 237–259.

Barabasi,A.-L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet*., **5**, 101–113.

Cho,K.-H. *et al*. (2005a) A unified framework for unraveling the functional interaction structure of a biomolecular network based on stimulus-response experimental data. *FEBS Lett*., **579**, 4520–4528.

Cho,K.-H. *et al*. (2005b) Unraveling the functional interaction structure of a cellular network from temporal slope information of experimental data. *FEBS J*., **272**, 3950–3959.

Covert,M.W. *et al*. (2001) Metabolic modeling of microbial strains *in silico*. *Trends Biochem. Sci*., **26**, 179–186.

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

De Jong,H. *et al*. (2004) Qualitative simulation of genetic regulatory networks using piecewise-linear models. *Bull. Math. Biol*., **66**, 301–340.

Fogel,D.B. (1991) *System Identification Through Simulated Evolution: A Machine Learning Approach to Modeling*. Ginn Press, Needham Heights, MA, USA.

Fogel,D.B. and Atmar,J.W. (1990) Comparing genetic operators with Gaussian mutations in simulated evolutionary processes using linear systems. *Biol. Cybern*., **63**, 111–114.

Harik,G.R. (1995) Finding multimodal solutions using restricted tournament selection. In *Proceedings of the Sixth International Conference on Genetic Algorithms*. Morgan Kaufmann, pp. 24–31.

Herrgard,M.J. *et al*. (2004) Reconstruction of microbial transcriptional regulatory networks. *Curr. Opin. Biotechnol*., **15**, 70–77.

Jeong,H. *et al*. (2000) The large-scale organization of metabolic networks. *Nature*, **407**, 651–654.

Kikuchi,S. *et al*. (2003) Dynamic modeling of genetic networks using genetic algorithm and S-system. *Bioinformatics*, **19**, 643–650.

Kimura,S. *et al*. (2005) Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm. *Bioinformatics*, **21**, 1154–1163.

Koza,J.R. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, Cambridge, MA, USA.

Maki,Y. *et al*. (2002) Inference of genetic network using the expression profile time course data of mouse P19 cells. *Genome Inform*., **13**, 382–383.

Morishita,R., Imade,H., Ono,I., Ono,N. and Okamoto,M. (2003) Finding multiple solutions based on an evolutionary algorithm for inference of genetic networks by S-system. In *Proceedings of the 2003 Congress on Evolutionary Computation*, Vol. 1, pp. 615–622.

Papin,J.A. *et al*. (2003) Metabolic pathways in the post-genome era. *Trends Biochem. Sci*., **28**, 250–258.

Rao,C.V. *et al*. (2002) Control, exploitation and tolerance of intracellular noise. *Nature*, **420**, 231–237.

Ronen,M. *et al*. (2002) Assigning numbers to the arrows: Parameterizing a gene regulation network by using accurate expression kinetics. *Proc. Natl Sci. USA*, **99**, 10555–10560.

Runarsson,T.P. and Yao,X. (2000) Stochastic ranking for constrained evolutionary optimization. *IEEE Trans. Evol. Comput*., **4**, 284–294.

Schmidt,H. *et al*. (2005) Identification of small scale biochemical networks based on general type system perturbations. *FEBS J*., **272**, 2141–2151.

Segal,E. *et al*. (2003) Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data. *Nat. Genet*., **34**, 166–176.

Spieth,C., Streichert,F., Speer,N. and Zell,A. (2004) Optimizing topology and parameters of gene regulatory network models from time-series experiments. In *Proceedings of the 2004 Genetic and Evolutionary Computation Conference*, LNCS 3102, Springer, pp. 461–470.

Stelling,J. (2004) Mathematical models in microbial systems biology. *Curr. Opin. Microbiol*., **7**, 513–518.

Sugimoto,M. *et al*. (2005) Reverse engineering of biochemical equations from time-course data by means of genetic programming. *BioSystems*, **80**, 155–164.

Sutton,M.D. *et al*. (2000) The SOS response: resent insights into *umuDC*-dependent mutagenesis and DNA damage tolerance. *Ann. Rev. Genet*., **34**, 479–497.

Thieffry,D. *et al*. (1998) From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in *Escherichia coli*. *BioEssays*, **20**, 433–440.

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

Tsai,K.-Y. and Wang,F.-S. (2005) Evolutionary optimization with data collocation for reverse engineering of biological networks. *Bioinformatics*, **21**, 1180–1188.

Vera,J. *et al*. (2003) Multicriteria optimization of biochemical systems by linear programming: application to production of ethanol by *Saccharomyces cerevisiae*. *Biotechnol. Bioeng*., **83**, 335–343.

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

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

Wolkenhauer,O., Kitano,H. and Cho,K.-H. (2003) Systems biology: Looking at opportunities and challenges in applying systems theory to molecular and cell biology. *IEEE Contr. Syst. Mag*., **23**, 38–48.

Zhang,B.-T. and Mühlenbein,H. (1995) Balancing accuracy and parsimony in genetic programming. *Evol. Comput*., **3**, 17–38.

Zhang,B.-T. *et al*. (1997) Evolutionary induction of sparse neural trees. *Evol. Comput*., **5**, 213–236.