Construction of Genetic Network Using Evolutionary Algorithm and Combined Fitness Function

Ando Shin[1] ando@iba.k.u-tokyo.ac.jp

Hitoshi Iba[2] iba@iba.k.u-tokyo.ac.jp

[1] Dept. of Electronics, School of Engineering, University of Tokyo, Japan
[2] Dept. of Frontier Informatics, School of Frontier Science, University of Tokyo, Japan

Abstract

This paper proposes a method to capture the dynamics in gene expression data using S-system formalism and construct genetic network models. Our purposed method exploits the probabilistic heuristic search and divide-and-conquer approach to estimate the network structure. In evaluating the network structure, we attempt a primitive integration of other knowledge to the statistical criterion. The Z-score is used to analyze the robust and significant parameters from stochastic search results. We evaluated the proposed method on artificially generated data and *E.coli* mRNA expression data.

Keywords: gene network, genetic algorithms, differential equation model, S-system, operon map, *E.coli*

1 Introduction

Micorarray experiments provide highly parallel observation of the cell activities. It allows us to observe the cellular dynamics along with the static behavior under various environments. The analysis of such behavior is one of the key elements in the functional analysis of the genome.

A large amount of knowledge on various biological systems, e.g., gene regulation, metabolic regulations, and signal transduction are being continually accumulated over the years, though there remains a large portion that is not well understood. The network models are frequently used to give general and comprehensive view of such mechanisms. The construction of a network model which can describe the behavior of the biochemical system is an important but difficult task addressed in recent bioinformatics.

This paper focuses mainly on constructing a genetic network automatically from gene expression data. The purpose of genetic network is to represent the regulation rules underlying the gene expression. It is regarded as an abstract mapping of the more complicated biochemical network which includes other components such as proteins, metabolites, etc. The genetic network may be used as the guidance for further biological experiments to explore higher level of interaction. Its practical usages include finding drug for transcriptional targets and biological engineering.

There are various models for analyzing gene regulations [2]. Analysis of gene expression using differential equation model [1, 4, 6, 15, 20, 23] is one of the frequently used approaches. The use of S-system has been proposed by [22] and [15, 23] for analysis of the time series gene expression data. It employs a very general formalism which allows us to capture the non-linear and general dynamics of the gene regulation. Since the constructing genetic network involves abstraction and mapping of unformalized interactions, the generality of the formula is essential.

Exhaustive or analytical methods for finding mathematically optimal model are generally unrealistic in this type of problem, due to the complexity of the problem and significant noise level. The analytical solution for S-system [11] likewise consumes significant amount of time, thus the use of genetic algorithm has been proposed [15, 9, 10] as more feasible approach. In the GA process, the kinetic parameters of an S-system model, i.e., a candidate gene network structures are represented by individuals. The individuals are optimized based on certain selection criteria. The Akaike's Information Criteria (AIC), derived from maximum likelihood model is used for selecting graph structures. AIC is a reliable estimate of the disparity between the true model and the candidate model and is less computationally intensive.

A previous work have used artificial S-system model in simulated experiment [15]. It showed that the model parameters can be estimated very precisely from the artificially generated data for small networks. Problem remains when applying to microarray data with significant noise level, as the estimated structure tends to include too many parameters that are falsely positive.

Since GA is a probabilistic search, repeated trials are required to assure statistically soundness. Instead of aiming to estimate the precise parameters, we used Z-score [7] to analyze which parameter is more significant and less diverse than the others. The analysis by Z-score is more qualitative, but is sufficient for comparison with the existing knowledge, or suggesting new regulation.

The divide-and-conquer approach also implemented in this approach, first estimates the subgraphs and then the entire graph. The dimensionality is reduced in the sub-problems, where regulation of one gene is estimated. This approach has shown more accuracy and efficiency to the conventional implementation.

Furthermore, we introduce a primitive integration of other genomic information in evaluation function. Many existing database stores enormous amount of knowledge including annotations, motifs, operon maps, etc. It is important to model gene network in accordance with gene expression and such knowledge, to determine larger and more complex correlations.

We used our method in Monte-Carlo simulation and also for analysis of *E.coli* data obtained from Stanford Microarray Database[1] and *E.coli* Genome Project[2].

This paper is structured as follow: section 2 describes the S-system formalism used as a general model of the genetic network. Further, AIC formula for evaluating a given candidate network and genetic algorithm implementation for estimating the network model are described. Finally, the integration of biological knowledge in evaluation function is proposed. Section 3 describes the setup of Monte-Carlo simulation and its analysis by Z-score. Section 4 describes the *E.coli* expression data and its analysis using the proposed approach.

2 Method

2.1 The Genetic Network Model and Evaluation Function

The time series expression data are obtained in form of data vector $\vec{x}(t_k)$, where $k(1 < k < T)$ is the timepoint of measured gene expression. $\vec{x} = \{x_1, ..., x_N\}$ indicates the snapshot of expression levels for N genes. We assume that gene expressions are regulated by a nonlinear model in the form:

$$d\vec{x}/dt = s(\vec{x}) \tag{1}$$

We use the S-system [22] as the function s, described as follows:

$$\frac{dx_i}{dt} = \alpha_\iota \prod_{j=1}^{N} x_j^{g_{ij}} - \beta_i \prod_{j=1}^{N} x_j^{h_{ij}} \tag{2}$$

The S-system is a type of power-law formalism based on ordinary differential equations, in which the component processes are characterized by two power-law functions terms. The two terms in (2)

[1] `http://genome-www4.stanford.edu/MicroArray/SMD/`

[2] `http://www.genome.wisc.edu/`

each represent the productive and inhibitory regulation influencing the variable at left-hand side of the equation. The parameters that define the S-system are: $M\{\alpha, \beta, g, h\}$. Biologically, *alpha*, β are interpreted to kinetic constants, and the g and h are considered as reaction order in biological engineering terms.

The error ε, between true expression $S(x)$ and observed expression x is inevitable in the microarray data. We assume ε as normally distributed error determined by standard deviation σ, which is constant over time and among all genes.

$$\varepsilon_k = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\{\frac{(x_k - S(x_k))^2}{2\sigma^2}\} \tag{3}$$

The log-likelihood Λ of the given set of time-ordered expression $X = \{x(0),,,x(T)\}$ for specific S-system model M and σ^2 is :

$$\Lambda(M, \sigma) = -\frac{1}{2\sigma^2} \sum_1^T (x - S(x))^2 - \frac{T}{2} ln(2\pi\sigma^2) \tag{4}$$

while the maximum likelihood estimate of the σ^2 is

$$\hat{\sigma}^2 = \frac{1}{T} \sum_1^T (x - S(x))^2 \tag{5}$$

The log likelihood of the estimated model is obtained by substituting (5) to (4).

As previously mentioned, AIC was used to select candidate models for its reliability in regression and timeseries prediction [19], and also for its computational efficiency. The AIC is formalized as (6) using log likelihood Λ and degree of freedom in the model Φ,

$$AIC = -2\Lambda + 2\Phi \tag{6}$$

Using AIC as the evaluation function, parsimonious pressure is applied in the model selection process. Both the error in expression and degree of freedom is considered to avoid overfitting to the noise.

2.2 Genetic Algorithm

Since the number of parameters in the model is in the order of n^2 to the number of variables n, the exhaustive search of the possible models are unrealistic for actual biochemical networks. We have used genetic algorithms (GA) to find the model which gives the smallest AIC. The GA is known to be a robust approach to generating network structures and models [3, 25]. The GA optimizes the set of parameters, i.e., the kinetic order matrices $G = \{g_{ij}\}, H = \{h_{ij}\}$, of the S-system instances. Note that the kinetic constants are not included as part of the structure in the following experiments, but can be determined by this method or through other means as well.

We have implemented a real-coded GA [12] and real-coded messy GA [5], known to be more efficient than conventional binary GA. Following are brief description of GA terms. An individual is an instance of S-system model parameters. A population consists of a certain number of individuals. A generation is an instance of GA populations. A generation produces a new generation by which it is replaced. The iterated production and replacement of the generation constitutes the GA.

In inference of network structure, the divide-and-conquer approach [9, 10] reduces the complexity of the problem by dividing the task into sub-problems. The divide-and-conquer approach consists of two phases.

In the first phase, the subgraphs of the entire network are estimated. n two-level hierarchical graphs are prepared. Each subgraph includes one of the genes as a child node and includes all genes as the potential parents, i.e., its regulators. Each subgraph is optimized by GA. In this phase, Each GA individual represents $2n$ variables, which are the kinetic orders of divided graphs. The variables are a candidate for a column of the kinetic order matrices, $\vec{G} = \{g_{i0}, ..., g_{in}\}, \vec{H} = \{h_{i0}, ..., h_{in}\}$.

In the second phase, the elite solutions from the first phase are combined to create candidate graph for the whole gene network. In this phase, an individual represents an array of $2n^2$ variables. The variables are the candidate for the kinetic order matrices, G, H. The GA optimization is performed multiple times in each phase to produce robust results.

We have tested different combination of GA implementations for each phase. The best performance was achieved when using extension of UNDX+MGG [17, 24] for the first phase, and extended real-coded messyGA [8] for the second phase.

Each of the evolutionary algorithms used in this approach has different characteristics. UNDX crossover preserves the statistical properties of the generation (i.e. means and deviation of the parameters), assuming that fitness landscape is normally distributed and variables are independent. The messy GA emphasizes the linkage between variables to generate 'schema', and explores the combination of schema for optimal solutions [13].

UNDX is designed for and have been successfully applied to 20-30 variable problems. On the other hand, the extended messy GA has been empirically shown to be applicable to 30 node problems. The combined approach arguably scales to more than 30-node problems, barring unknown strong dependence among variables or very complicated fitness landscape.

Since the realistic biochemical network have much larger dimension, it would require alternative measures to limit dimension of the search, e.g., incorporating genomic knowledge from sequential data or considering the modularity of the network.

Further, the local optimization algorithm is implemented within the GA. The combination of greedy search and GA, sometimes called memetic algorithm, is known to perform better [18]. The greedy search proceeds as follow: The kinetic orders of trivial value are mutated to 0. The AIC score of the mutated model is compared with that of the original model. If the score is improved by mutation, the new model is maintained. Otherwise, previous parameter values are restored. Each parameter is mutated once, sequentially. Similarly, the "gradual optimization strategy" [15] have been used to force the deletion of smaller values. Compared to this strategy, our approach requires more evaluation but can recover from over-deletion.

The overall procedure for estimating the model by GA is as follow:

- Phase 1: (repeat for each considered gene)
  - initialization: prepare candidate subgraphs with $2n$ parameters
  - repeat the following
  - evaluation: calculate AIC score for each individual
  - selection: select individuals based on its score
  - genetic operation: apply genetic operators to selected individuals to create a new generation.
- Phase 2:
  - initialization: combine elite individuals from Phase 1 to create candidate models
  - repeat the following
  - evaluation: calculate AIC score for each individual
  - selection: select individuals based on its score
  - genetic operation: apply genetic operators to selected individuals to create a new generation.

2.3 Integrating Prior Knowledge and Environmental Changes

With recent technological advances, abundance of biological and genomic information have become available in various databases, e.g., biological pathway database, sequence motifs, annotations, protein secondary and tertiary structure. The integration of multiple genomic information is one of the future goals of bioinformatics. Reflecting such knowledge may benefit the inference of gene network structure in many ways. First, though microarray analysis is essential in understanding cell dynamics, the experiment is expensive and there may not be sufficient data to provide enough information to determine the gene network. Secondly, incorporating prior knowledge would increase the precision of the inference and reducing the effect of noise and the false positive predictions. Further, the knowledge may limit the search space and number of local optima, thus increasing the efficiency of the search. Finally, the new found knowledge may be evaluated in the context of prior knowledge.

Previous works proposed to incorporate biological expert knowledge interactively into the gene network modeling process [10]. They basically used the knowledge to reduce the dimensionality of the problem [10, 16], by fixating on known substructure. Our approach, on the other hand, assigns weights to the individual pathways and adjusts the evaluation of the candidate model according to the knowledge. This would enable us to consider the level of confidence for individual knowledge and possibly rectify the errors in given knowledge. This would be preferable because some of the existing database include preliminary results and are prone to errors.

The knowledge score KW of a candidate network is determined by sum of knowledge weight on its individual pathways.

$$KW = \sum_i \sum_j \delta(g_{ij}, h_{ij}) \times k_{ij} \tag{7}$$

$$\begin{aligned} \delta\,(x, y) &= 0 \quad \{x = 0 \wedge y = 0\} \\ &= 1 \quad \{x \neq 0, y \neq 0\} \end{aligned} \tag{8}$$

The weight k_{ij} for each pathway from gene i to j is determined from existing knowledge. The $k_{ij} = 1$ is assigned if no knowledge available for a particular pathway. The score increases for every known pathway included in the candidate network. The knowledge matrix $K = \{k_{ij}\}$ is prepared prior to the modeling process.

The fitness function of the candidate network is adjusted as follows:

$$\begin{aligned} Fitness &= AIC/KW \quad \{AIC > 0\} \\ &= AIC \times KW \quad \{AIC < 0\} \end{aligned} \tag{9}$$

The common databases mainly signify existence of relations, but note that this approach can also incorporate the absence of influence as well.

3 Simulated Experiment

3.1 Analysis by Z-score

The performance of our approach was first evaluated using Monte-Carlo simulation. Acyclic network structures were created randomly as the true model in the simulation. The datapoints were generated using fourth-order Runge-Kutta for equations (2) and (3) with standard deviation $\sigma^2 = 1, 2$. All kinetic constants α and β were set to 10. Additionally, only one type of regulation was allowed from a particular gene to another, i.e., one gene cannot have both promotive and inhibitory regulation from the same gene.

From each model, we generated 20 expression matrices with different noise pattern. They were used to calculate AIC score of the candidate gene network structures. Since GA is a probabilistic search method, its performance is evaluated in multiple runs. The results of GA were analyzed using Z-score.

As previously mentioned in 2.2, we used 10 repetition of the second phase to obtain 10 candidate optimal models. For each parameter p included in the models, we calculated the mean magnitude μ_p and the standard deviation S_p to derive $Z_p = \mu_p/S_p$. The Z_p quantity can be used as a signal to noise measurement to imply robust parameters.

Our approach reconstructed the network structure with very high accuracy under the noise setting $\sigma^2 = 1, 2$. We will summarize the experiment with the results from a typical target model. Table 1 shows the list of false positive prediction for structure of the typical model Fig.1. The average Z score of the true positive predictions were 1.64. We can see that Z score of the false prediction are disperse and has significantly smaller value.

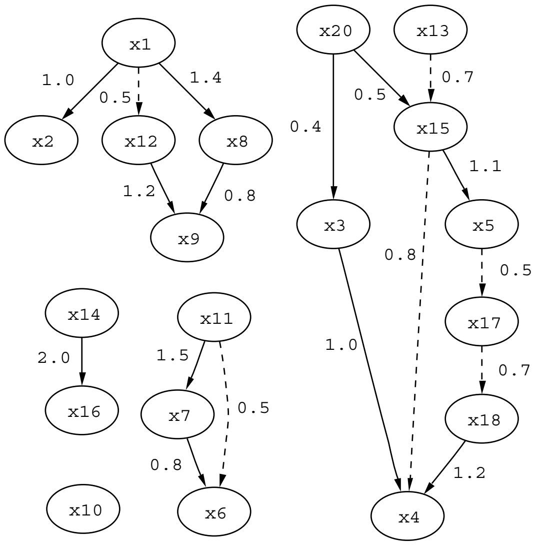

Figure 1: A typical sample of the simulated model. The solid/dotted line indicate positive/negative kinetic orders. Values of the kinetic order is indicated by the number by the arrows.

3.2 Knowledge-integrated Evaluation Function

Next, we tested the effect of knowledge-integrated fitness function. The following experiment is performed under assumption that knowledge about p percentages of the pathways, i.e., nonzero kinetic order values g_{mn}, h_{mn}, has been predicted via alternative method. The knowledge is given in a form of knowledge matrix K, whose elements are $k_{mn} = r$ for informed pathways and $k_{ij} = 1$ for other pathways. The knowledge weight $r = 1, 1.5, 2, 2.5, 4$ were tested and the best results were obtained with $r = 2$.

Table 2 shows the sensitivity of the best candidate network when adjusted fitness value was used. The model in Figure 1 were used for simulation. We compared the cases for $p = 10, 20, 50$. The sensitivity improves significantly with the increased amount of given information.

Further experiments explore how the incorrect information affect the prediction. The q percent false information were included into the knowledge matrix, i.e., $\{k_{mn} = 1, g_{mn} = 0, h_{mn} = 0\}$. In the following experiment, percentage of incorrect information q is 5. The result is shown in Table 3. We compared the result using knowledge weight $r = 1.0, 1.5, 2, 2.5$ in terms of sensitivity of the optimal candidate structure, and also the percentage of corrected knowledge, i.e., the pathway in optimal graph that correctly predicted $g_{ij} = h_{ij} = 0$, despite the given information of $k_{ij} = r$.

Table 1: False positive table ($\sigma^2 = 2$).

Source	Dest.	Z	Source	Dest.	Z	Source	Dest.	Z
X1	X4	0.0161	X7	X2	0.0846	X13	X3	0.431
	X6	0.187		X3	0.773		X9	0.0271
	X10	0.103958		X5	0.274	X14	X1	0.0867
	X19	0.116		X9	0.379		X3	0.224
X2	X4	0.0543		X10	0.0534		X4	0.0244
	X8	0.078		X14	0.0123		X6	0.0943
	X9	0.0954		X16	0.133		X9	0.206
	X16	0.448	X8	X1	0.27		X10	0.145
	X20	0.00501		X10	0.0389		X12	0.0808
X3	X5	0.318		X16	0.0447		X18	0.163
	X9	0.0773	X10	X3	0.205	X15	X6	0.266
	X11	0.00409		X6	0.0296		X9	0.335
	X18	0.00622		X8	0.0051	X16	X4	0.0968
X4	X6	0.0673		X14	0.0695		X6	0.045
	X19	0.498	X11	X4	0.194		X8	0.351
X5	X2	0.00998		X5	0.0717		X10	0.499
	X8	0.0469		X8	0.603		X19	0.0246
	X10	0.224		X9	0.303	X17	X1	0.0721
	X19	0.0702		X15	0.0308		X9	0.375
X6	X9	0.412		X20	0.0109		X16	0.205
	X13	0.0828	X12	X4	0.0726	X19	X3	0.0605
				X10	0.0805		X6	0.0577
				X12	0.17		X8	0.00202
				X19	0.0622		X14	0.266
						X20	X19	0.0404

Table 2: Sensitivity of optimal fitness model vs. amount of knowledge ($r = 2$).

Knowledge	0%	20%	50%
Average Sensitivity	54%	72%	88%

Table 3: Sensitivity and correction percentage vs. knowledge weight (knowledge about 50% of the pathway are given, 5% of the knowledge is false positive).

Weight	1.0	1.5	2	2.5
Sensitivity	64%	80%	82%	76%
Rate of correction	82%	85%	80%	73%

4 Analysis of *E.coli* Microarray Data

We applied our analysis to time-ordered gene expression data, which was published by [14] and has also been analyzed in [16]. We used the data as a benchmark to evaluate our method in comparison to above references.

The dataset focuses on the well studied regulatory process of tryptophan metabolism. It includes three time series with duration of five time points and same initial values. The data show the changes in mRNA level under starvation of tryptophan in one time series and also by overdose of tryptophan in other two. The analysis focused on expression of 14 important operons, which were selected by [16]. They belong to key operons related to tryptophan perturbation. The operon map information are available from RegulonDB[3] [21].

Note that in this experiment, only the the first phase was applied, i.e., the reconstruction of the whole network was omitted. Our assumption was that the data weren't sufficient data to reconstruct the global gene network. We estimated the parents for each of the selected genes, considering every gene as possible parents.

The tryptophan level (excess/starved) were constant. The expression datapoints were supplemented with interpolation, Monte-Carlo simulation, and resampling. The data was first interpolated with polynomial splines with MDL criteria. The interpolated data were used for numerical integration of the S-system formula. While the interpolation enables the analysis of the sparse and underdetermined data, it should be carefully applied since the interpolation function inevitably biases the result of the analysis. In additon, the kinetic constants were estimated from maximum and minimum differentiation of the interpolated data.

Aside from the interpolation, the datapoints were processed for purpose of calculating the AIC score as follow: We first prepared the supplementary data, which was generated using mean and standard deviation of the original data with normal distribution function. The supplementary data were mixed with the original, to quadruple the number of datapoints. Data points were resampled from mixed data without overlap, and used to calculating AIC score of the candidate model.

The GA and Z-score analysis of the processed data is same as previously described in the simulated experiment. This experiment used operon map knowledge to adjust the fitness score. Known relations among operon member genes were weighed with $r = 2$.

We empirically set the threshold of Z-score for significant regulations at $Z_{th} = 1.2$. The results were very dispersed, possibly due to the noise in the microarray analysis or the formalism of the model. For each considered genes, only one or two genes had shown above threshold Z-score. Table 4 shows the list of such operons that appeared as the regulators of the considered genes. Notably in this table, the operons predicted to be inversely affected by trpR: trp operons, mtr, and aroH were the known trp repressors. The tnaA and tnaB constitute an operon, but tnaB had significantly weaker expression in the used data. This may have resulted in prediction that tnaB was regulated by aroH, which is one of the trp repressors. Additionally, operons aroF, aroG, aroH, and aroL were all affected by tryptophan-excess factor, which coincides with the known biological observation.

Table 4: Significant relation among considered operons in Z-score analysis. The first operons appeared in the predicted S-system models as the regulator of the operons pointed by arrows, with Z-score higher than $Z_threshold$=1.2.

tyrR → aroF	tyrR → aroG	trpR → aroH	tyrR → aroL	tyrR → aroP
aroF, trpR → tnaA	aroF, aroH → tnaB	trpR → trp	trpR → mtr	

[3] `http://www.cifn.unam.mx/Computational_Genomics/regulondb/`

5 Conclusion

We have proposed an extended GA to estimate genetic network in S-system formalism. The experiments showed the proposed method to be capable of reliable estimation in simulation and in microarray data, combined with guidance of other genomic knowledge. Modeling with differential equation, such as the S-system, can exploit the continuous value and the time order of the gene expression. On the other hand, it requires a lot of data points, which was addressed by interpolation of the sparse data in this paper. The interpolation inevitably biases the result of the analysis and should be carefully chosen for each given data set.

The Z-score provides abstract analysis of the parameters, signifying the magnitude of confidence for each predicted regulation. This is suitable for simplified comprehension and guidance for biological experiment. For purpose of precise simulation, the estimated relation should be optimized with more specific mathematical models.

The proposed GA framework is general enough to apply to other models of biological networks. Combination of several different models is in the works for more general analysis.

Our long-term goal is to reconstruct larger network, e.g., gene network of network human has more components and complicated interactions compared to primitive organisms. Such task requires significant improvement in the following: increase in accuracy, quantity, and resolution of the observation, improvement in efficiency of the genetic algorithms, additional knowledge of the domain to reduce the search space.

References

[1] Alves, R. and Savageau, M., Systemic properties of ensembles of metabolic networks: application of graphical and statistical methods to simple unbranched pathways, *Bioinformatics*, 16(6):534–547, 2000.

[2] Brazhnik, P., de la Fuente, A., and Mendes, P., Gene networks: how to put the function in genomics, *Trends. Biotechnol.*, 20(11):467–472, 2002.

[3] Chau, C.W., Kwong, S., Diu, C.K., and Fahrner, W.R., Optimization of hmm by a genetic algorithm, In *IEEE International Conference on acoustics, Speech, and Signal Processing*, 1727–1730, 1997.

[4] Chen, T., He, H.L., and Church, G.M., Modeling gene expression with differential equations, In *Pacific Symp. Biocomputing*, 29–40, 1999.

[5] David, E.G., Korb, B., and Deb, K., Messy genetic algorithms: motivation, analysis, and first results, *Complex Systems*, 3(5):493–530, 1989.

[6] De Hoon, M., Imoto, S., Kobayashi, K., Ogasawara, N., and Miyano, S., Inferring gene regulatory networks from time-ordered gene expression data of bacillus subtilis using differential equations, In *Pac. Symp. Biocomput.*, 17–28, 2003.

[7] D'Haeseleer, P., Liang, S., and Somogyi, R., Genetic network inference: from co-expression clustering to reverse engineering, *Bioinformatics*, 16(8):707–726., 2000.

[8] Goldberg, D.E., Deb, K., Kargupta, H., and Harik, G., Rapid, accurate optimizaiton of difficult problems using fast messy genetic algorithms, In *Fifth International Conference on Genetic Algorithms*, 56–64, 1993.

[9] Iba, H. and Ando, S., Inference of gene regulatory model by genetic algorithms, In *Conference on Evolutionary Computation*, 2001.

[10] Iba, H. and Mimura, A., Inference of a gene regulatory network by means of interactive evolutionary computing, *Information Sciences*, 145(3-4):225–236, 2002.

[11] Irvine, D.H. and Savageau, M.A., Efficient solution of nonlinear, *SIAM J. Numer. Anal.*, 27:704–735., 1990.

[12] Jonikow, C.Z. and Michalewicz, Z., An experimental comparison of binary and floating point representations in genetic algorithms, In *International Conference on Genetic Algorithms*, 31–38, 1991.

[13] Kargupta, H., The gene expression messy genetic algorithm, In *International Conference on Evolutionary Computation*, 814–819, 1996.

[14] Khodursky, A., Peter, B.J., Cozzarelli, N.R., Botstein, D., Brown, P.O., and Yanofsky, C., DNA microarray analysis of gene expression in response to physiological and genetic changes that affect tryptophan metabolism in *Escherichia coli*, *Proc. Natl. Acad. Sci. USA*, 97(22):12170–12175, 2000.

[15] Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K., and Tomita, M., Dynamic modeling of genetic networks using genetic algorithm and S-system, *Bioinformatics*, 19(5):643–650, 2003.

[16] Ong, I., Glasner, J.D., and Page, D., Modelling regulatory pathways in *E. coli* from time series expression profiles, *Bioinformatics*, 18(S1):S241–S248, 2002.

[17] Ono, I. and Kobayashi, S., A real-coded genetic algorithm for function optimization using unimodal normal distribution crossover, In *7th International Conference on Genetic Algorithms*, 246–253, 1997.

[18] Pablo, M., On evolution, search, optimization, genetic algorithms and martial arts: towards memetic algorithms, *Technical Report C3P 826, Caltech Concurrent Computation Program Report*, Using Smart Source Parsing, 1989.

[19] Priestley, M.B., *Spectral Analysis and Time Series*, Academic Press, London, 1994.

[20] Sakamoto, E. and Iba, H., Evolutionary inference of a biological network as differential equations by genetic programming, *Genome Informatics*, 12:276–277, 2001.

[21] Salgado, H., Santos-Zavaleta, A., Gama-Castro, S., Millan-Zarate, D., Diaz-Peredo, E., Sanchez-Solano, F., Perez-Rueda, E., Bonavides-Martinez, C., and Collado-Vides, J., RegulonDB (version 3.2): transcriptional regulation and operon organization in *Escherichia coli* K-12, *Nucleic Acids Res.*, 29(1):72–74, 2001.

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

[23] Voit, E.O. and Radivoyevitch, T., Biochemical systems analysis of genome-wide expression data, *Bioinformatics*, 16(11):1023–1037, 2000.

[24] Yamamura, H., Satoh, M., and Kobayashi, S., Minimal generation gap model for gas considering both exploration and exploitation, In *Iizuka'96*, 494–497, 1996.

[25] Yao, X., Review of evolutionary neural networks, *International Journal of Intelligent Systems*, 8:539–567, 1993.