Inference of Genetic Networks using S-system: Information Criteria for Model Selection

Nasimul Noman and Hitoshi Iba
Department of Frontier Informatics
The University of Tokyo
Chiba 277-8561, Japan
{noman,iba}@iba.k.u-tokyo.ac.jp

ABSTRACT

In this paper we present an evolutionary approach for inferring the structure and dynamics in gene circuits from observed expression kinetics. For representing the regulatory interactions in a genetic network the decoupled S-system formalism has been used. We proposed an Information Criteria based fitness evaluation for model selection instead of the traditional Mean Squared Error (MSE) based fitness evaluation. A hill climbing local search method has been incorporated in our evolutionary algorithm for attaining the skeletal architecture which is most frequently observed in biological networks. Using small and medium-scale artificial networks we verified the implementation. The reconstruction method identified the correct network topology and predicted the kinetic parameters with high accuracy.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and genetics; I.2.8 [**Problem Solving, Control Methods, and Search**]: Heuristic methods

General Terms

Algorithms, Design, Performance

Keywords

Genetic network, S-system, Reverse engineering, Information criteria

1. INTRODUCTION

In the last few decades various types of models for representing gene regulatory networks have been proposed [3, 5, 9, 15, 22] as well as many algorithms have been developed to trace genetic interactions from expression data. Different genetic network models differ in terms of details of biochemical interactions incorporated, discrete or continuous gene expression level used, deterministic or stochastic approach applied, etc [6]. And these criteria define how closely the model can represent genetic interactions. Generally, detailed biochemical modeling is very useful for capturing the precise mechanism in common regulatory pathways. However as we try to approach from a more abstract to a more real representation the complexity of the model increases accordingly. And with the increase of the model complexity the data requirement for learning the model parameters also increases. Therefore a genetic network model is desirable that triggers a compromise between these two contradictory requirements.

The S-system model [21] of gene networks is based on the Biochemical System Theory (BST) - a generalized framework for modeling and analyzing biological systems [19, 20]. The model is organizationally rich enough to reasonably capture various dynamics and mechanisms that could be present in a complex system of genetic regulation. S-system is a dynamic model for biochemical pathways, having a good compromise between accuracy and mathematical flexibility. Nevertheless, inferring the genetic networks using S-system is occluded by the number of the parameters ($2N(N+1)$, where N is the number of genes in the network) that has to be estimated. In order to deal with the problem of high-dimensionality, decoupling of the original model was performed and has been used successfully in gene network reconstruction [12, 14, 17, 26].

In this work we have used the decoupled form of the S-system model for representing gene regulatory networks and for reconstruction we used an evolutionary algorithm based on *Trigonometric Differential Evolution* (TDE). For fitness evaluation of the candidate models we have used an Akaike's Information Criteria (AIC) based fitness function. For obtaining the sparse architecture we embedded a hill climbing local search process in our algorithm. We tested the proposed method using artificial gene regulatory networks of different dimensions. Experiments showed that the proposed approach can estimate the correct network structure and precise kinetic parameter values. The next section of the paper describes the S-system model both in its canonical form and decoupled form. The third section discusses the traditional fitness estimation methods and presents the proposed fitness evaluation criterion for evaluating the candidate networks. In section four, our algorithm for inferring S-system model based gene networks is described. Section five reports the experiments with the results to verify the effectiveness of the proposed algorithm and the fitness function. Some general discussions are presented in section six and finally section seven concludes the paper.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

2. THE S-SYSTEM MODEL FOR GENE NETWORKS

2.1 Canonical Form

Savageau proposed the S-system model [21] as a set of tightly-coupled non-linear differential equations and the systematic structure of the model is given by

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{N} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{N} X_j^{h_{ij}} \quad (1)$$

where N is the number of genes in the system and X_i is the expression level of i-th gene. The terms g_{ij} (h_{ij}) represent the strength of the regulation exerted by X_j on the synthesis (degradation) of X_i. Therefore, the first term in right-hand side of (1) represents all influences that increase X_i, whereas the second term represents all influences that decrease X_i. An exponent of zero for any X_j means that variable has no direct influence on the rate of the corresponding aggregate process; a positive exponent means they are positively correlated, and a negative exponent means they are negatively correlated. The set of parameters that defines a S-system model is: $\Omega\{\alpha, \beta, g, h\}$. In a biochemical engineering context, the non-negative parameters α_i , β_i are called *rate constants*, and real-valued exponents g_{ij} and h_{ij} are referred to as *kinetic orders*. It is known that biological networks are sparse[4], which means the number of regulators that have effect on a single gene is relatively small; so many of the *kinetic orders* are zero in real condition.

Usually the inference method tries to estimate the set of model parameters Ω such that it minimizes the Mean Squared Error (MSE) between the experimentally obtained gene expression levels and the gene expression levels numerically calculated by solving (1). But for an N-dimensional network the number of system parameters to be estimated is $2N(N+1)$ which increases quadratically with the network size. Moreover, as the model is described as a system of nonlinear differential equations the regression task becomes more difficult especially for larger networks. That is why, application of the model was limited to small-scale gene regulatory networks.

2.2 Decoupled Form

As mentioned earlier, in order to deal with problem of high-dimensionality and to facilitate the regression task, decoupling of the original model has been performed [12, 14]. This decoupled S-system model allows its application to larger gene network inference problems. Using the suggested decomposition strategy the original optimization problem is divided into N sub-problems [12, 14]. In each of these subproblems the parameter values of gene i ($\alpha_i, \beta_i, g_{ij}$ and h_{ij}) are individually estimated for capturing the dynamics of gene i. In other words, this disassociation technique divides a *2N(N+1)* dimensional optimization problem into N sub-problems of *2(N+1)* dimensions. In i-th sub-problem for gene i, $X_i^{cal}(t)$ is calculated by solving the following differential equation instead

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{N} Y_j^{g_{ij}} - \beta_i \prod_{j=1}^{N} Y_j^{h_{ij}} \quad (2)$$

For solving the system of differential equation (1) we need the concentration levels X_j $(j = 1, \cdots, N)$ each of which are numerically integrated. But in the decoupled formalism, while solving the differential equation (2) in i-th subproblem (corresponding to gene i), the concentration level $Y_{j=i}$ is obtained by solving the differential equation whereas the other expression levels $Y_{j\neq i}$ are to be estimated directly from observed time-series data. The optimization task for the tightly coupled S-system model is not trivial because Eq. (1) is non-linear in all relevant cases, thus requiring iterative optimization in a larger parameter space, where 95% of the total optimization time is expended in numerical integration of the differential equations [26]. Therefore such disassociation could be very useful in reducing the computational burden. Moreover the experimental results showed their usefulness in estimating the network parameters [12, 14, 17]. In this work we have applied linear spline interpolation [18] for direct estimation of expression levels $Y_{j\neq i}$.

3. MODEL EVALUATION CRITERIA

3.1 Generic Fitness Evaluation Function

We need some measure for evaluating different candidate models that are encountered while searching for the set of optimal parameters for the target network. As mentioned earlier, the most commonly used evaluation criterion is the discrepancy between the numerical solution of the differential equation and the observed system dynamics. Tominaga *et al.* gave the MSE as the fitness evaluation function which should be minimized by Genetic Algorithm (GA) [25]. But the search space is notoriously multimodal and easily traps a search algorithm in some local optima that is capable of reproducing the almost same time-course. Since a single set of time course data can not give any general conclusion about the overall behavior of a complex dynamic system [24], use of multiple sets of time course data was found more useful. And using multiple set of dynamics, the MSE based fitness evaluation function for the canonical problem becomes

$$f^{MSE} = \sum_{k=1}^{M} \sum_{i=1}^{N} \sum_{t=1}^{T} \left\{ \frac{X_{k,i}^{cal}(t) - X_{k,i}^{exp}(t)}{X_{k,i}^{exp}(t)} \right\}^2 \quad (3)$$

where $X_{k,i}^{exp}(t)$ is the experimentally observed expression level of gene-i in the k-th set of time courses at time t and $X_{k,i}^{cal}(t)$ is the numerically calculated expression level of gene-i in the k-th set of time series at time t. M is the set of time series used, T is the number of sampling points of the experimental data. In this form of optimization problem the search algorithm tries to find a set of parameters that minimizes f^{MSE}.

In the decoupled form the relative error for the expression levels of each gene is considered individually for evaluating the candidate set of parameters for that particular gene. In other words, the sum of squared relative errors between experimental and calculated gene expression levels of gene i is used as the fitness function in subproblem i. So the objective function of the subproblem corresponding to the i-th gene becomes

$$f_i^{MSE} = \sum_{k=1}^{M} \sum_{t=1}^{T} \left\{ \frac{X_{k,i}^{cal}(t) - X_{k,i}^{exp}(t)}{X_{k,i}^{exp}(t)} \right\}^2 \quad (4)$$

And in subproblem i we try to estimate the parameters $\Omega_i = \{ \alpha_i, \beta_i, g_{ij}, h_{ij} \ (j = 1 \cdots N) \}$ for gene i that minimizes f_i^{MSE}.

3.2 Attaining Skeletal Network Structure

Generally, very few genes or proteins interact with a particular gene in biological networks [4]. But one major difficulty in the S-system based network inference process is detecting the skeletal system architecture that generates the experimentally observed dynamics. Because of the high degree-of-freedom of the model there exist many local minima in the search space that mimic the time-courses very closely. Therefore any method attempting to reproduce the time dynamics only, often gets stuck to some local optimum solution and fails to obtain the skeletal structure [10]. Kikuchi *et al.* suggested to penalize the fitness function by using all the *kinetic orders* (i.e. g_{ij} and h_{ij}) of the network [10]. Use of such *pruning term* or *penalty term*, based on Laplacian regularization term, in the basic fitness function of (3) was useful for finding a sparse network architecture in the canonical optimization problem [10, 16]. But because of high dimensionality these fitness functions have been applied to small scale networks only.

Based on the same notion, Kimura *et al.* added another more effective *penalty term* to the objective function of (4) for obtaining sparse network structure in the decoupled form of the problem [11, 12]

$$f_i = \sum_{k=1}^{M}\sum_{t=1}^{T}\left\{\frac{X_{k,i}^{cal}(t) - X_{k,i}^{exp}(t)}{X_{k,i}^{exp}(t)}\right\}^2 + c\sum_{j=1}^{N-I}(|G_{ij}| + |H_{ij}|) \tag{5}$$

where G_{ij} and H_{ij} are given by rearranging g_{ij} and h_{ij}, respectively, in ascending order of their absolute values (i.e., $|G_{i1}| \leq |G_{i2}| \leq \cdots \leq |G_{iN}|$ and $|H_{i1}| \leq |H_{i2}| \leq \cdots \leq |H_{iN}|$). And I is the maximum allowed cardinality (in-degree) of the network and c is the penalty constant. The superiority of this *penalty term* lies in including the maximum cardinality of the network. And thereby, this *pruning term* will penalize only when the number of genes that directly affect the i-th gene is higher than the maximum allowed in-degree I, thereby will cause most of the genes to disconnect when this penalty term is applied. However, very few genes affect both activation and repression of a specific gene. Therefore designing the penalty term considering both synthetic and degradative regulations together rather than separately will be more effective. Because such penalty will penalize whenever total number of regulators (whether synthetic or degradative) is greater than maximum allowed cardinality. Therefore, Noman and Iba suggested a further modification to the penalty term of (5) as follows [17]

$$f_i = \sum_{k=1}^{M}\sum_{t=1}^{T}\left\{\frac{X_{k,i}^{cal}(t) - X_{k,i}^{exp}(t)}{X_{k,i}^{exp}(t)}\right\}^2 + c\sum_{j=1}^{2N-I}(|K_{ij}|) \tag{6}$$

where K_{ij} are the *kinetic orders* (i.e. g_{ij} and h_{ij}) of gene i sorted in ascending order of their absolute values. Use of (6) instead of (5) as fitness function can identify the zero valued parameters increasingly and thus obtain the skeletal network structure more precisely.

3.3 Proposed Fitness Evaluation Criterion

Information criteria provide a simple method to choose from a range of competing models. Many information criteria are available such as AIC, BIC, HQ, GCV, FPE etc, but it is not clear which one is the best for a given selection task and none perform well for all model selection problems. However, Akaike's Information Criteria (AIC) [1] is most commonly used in statistical modeling to show disparity between the true model and the estimated one. Suppose $\varepsilon_{k,i}(t)$ is the error between the experimental and calculated expression level of gene-i in the k-th set of time course at instant t, i.e. $\varepsilon_{k,i}(t) = (X_{k,i}^{cal}(t) - X_{k,i}^{exp}(t))$. If we assume $\varepsilon_{k,i}(t)$ is normally distributed with mean $\mu_i = 0$ and standard deviation σ_i, which are constant for all sets of dynamics of gene-i and over time, then the probability density function of $\varepsilon_{k,i}(t)$ is given by

$$p.d.f_{\varepsilon} = \frac{1}{\sqrt{2\pi\sigma_i^2}} exp\left\{-\frac{(X_{k,i}^{cal}(t) - X_{k,i}^{exp}(t))^2}{2\sigma_i^2}\right\} \tag{7}$$

The log-likelihood Λ_i of the expression data of gene-i for a set of parameters Ω_i for gene-i is

$$\Lambda_i(\Omega_i, \sigma_i) = -\frac{1}{2\sigma_i^2}\sum_{1}^{TM}(X_{k,i}^{cal}(t) - X_{k,i}^{exp}(t))^2 - \frac{TM}{2} ln(2\pi\sigma_i^2) \tag{8}$$

and the maximum likelihood estimate of σ_i^2 is obtained from

$$\hat{\sigma}_i^2 = \frac{1}{TM}\sum_{1}^{TM}(X_{k,i}^{cal}(t) - X_{k,i}^{exp}(t))^2 \tag{9}$$

The log-likelihood of the estimated model is obtained by substituting (9) into (8).

Different information criteria are formulated as a penalized log-likelihood and particularly AIC is defined as [1]

$$AIC = -2\Lambda + 2\Phi \tag{10}$$

where Φ is the is the number of parameters included in the model. When AIC is used for selecting among the alternative models then the model with lowest AIC value is chosen. This original form of AIC has been used for model selection by Ando and Iba [2]. The second term of AIC is the *penalty term* which penalizes for addition of model parameters. However many modification or extension of the *penalty term* has been suggested resulting in various modified forms of AIC. For obtaining a network model with sparse connectivity among the components we propose the following fitness evaluation criterion for subproblem-i corresponding to gene-i

$$f_i^{AIC} = -2\Lambda_i + 2\Phi_i + c\sum_{j=1}^{2N-I}(|K_{ij}|) \tag{11}$$

As mentioned in section 3.2, this additional penalty term in (11) was designed to penalize a model only if the number of regulators included is higher than the maximum allowed for the network. Therefore, as long as the number of regulators is smaller than the maximum in-degree allowed, this additional penalty term will have zero effect in model selection. But it will interfere with the regular AIC fitness function only when the number of genes that directly influence the gene under consideration is higher than maximum allowed in-degree and will assist it in finding a sparse network architecture. This penalty term also introduces another parameter c in the fitness function but the value of this parameter can be chosen in a very easy empirical way as will be explained later. Using the fitness function given in (11) both the discrepancy in the expression levels and degree of freedom is considered for model selection as well as the sparse network structure is searched. Furthermore, in our experiment we have found that without this penalty

term the pure AIC alone cannot identify the precise skeletal network structure as will be shown later.

4. INFERENCE METHOD

Due to the complexity of the problem, finding an optimal solution using analytical techniques is not feasible because it would need significant amount of time and computational power. Evolutionary Computation (EC) has proven itself as a useful technique for exploring complex and high dimensional search spaces. Therefore, many of the real world problems involving finding optimal parameters, which might be difficult for traditional methods, are ideal for EC. Consequently the problem of network reconstruction has seen many applications of EC [8, 10, 13, 27]. In the following subsections we describe an extended evolutionary approach, based on Trigonometric Differential Evolution (TDE), for estimating the parameters for target genetic network.

4.1 Trigonometric Differential Evolution (TDE)

One of most recent evolutionary optimization approaches is Differential Evolution(DE) proposed by Storn and Price [23]. Because of its effectiveness and efficiency it has been successfully applied to many fields where we need to find the global optimal solution of the problem, e.g. pattern recognition, communication, engineering etc. Introducing another new operator called Trigonometric Mutation Operation (TMO), Fan and Lampinen have extended the basic algorithm for higher convergence speed and greater robustness [7]. This modified DE algorithm, known as Trigonometric mutation DE (TDE), is used as the optimization agent in the core of our algorithm.

Like other *Evolutionary Algorithms* (EAs) TDE is a population-based search heuristic. Each population consists of a certain number of individuals where each individual represents a candidate solution for the problem. A new generation (an instance of population) is created from the current generation and the new one replaces the current one. Thus producing and replacing new generations in an iterative manner, TDE searches for the optimal solution of the problem. We explain the search procedure for sub-problem i.

For estimating the solution for sub-problem i an initial population of random individuals is created where each individual consists of parameters Ω_i=$\{$ α_i, β_i, g_{ij}, h_{ij} $(j = 1, \cdots, N)$ $\}$ for gene i. Then the fitness of each individual is evaluated using Eq. (11). Then new individuals are generated by the combination of randomly chosen individuals from the current population. Specifically, for each individual x^i_G, $i = 1, \cdots, P$, three other random individuals x^j_G, x^k_G and x^l_G (such that j, k and $l \epsilon \{1, \cdots, P\}$ and $i \neq j \neq k \neq l$) are selected from generation G; P is the number of individuals in G. Then a new trial individual y^i_G (i.e. a new candidate solution) is generated using probabilistic mutation operation according to the following equations

$$y^i_G = x^j_G + F(x^k_G - x^l_G) \tag{12}$$

$$y^i_G = (x^j_G + x^k_G + x^l_G)/3 + (p_k - p_j)(x^j_G - x^k_G) + (p_l - p_k)(x^k_G - x^l_G) + (p_j - p_l)(x^l_G - x^j_G) \tag{13}$$

where

$$p_j = |f(x^j_G)|/p' \qquad p_k = |f(x^k_G)|/p' \qquad p_l = |f(x^l_G)|/p'$$
$$\text{and} \qquad p' = |f(x^j_G)| + |f(x^k_G)| + |f(x^l_G)|$$

F is called the *scaling factor* or *amplification factor*. Eq. (12) represents the regular mutation operation in DE and Eq. (13) represents TMO proposed by Fan and Lampinen [7]. This TMO is applied with probability M_t and the regular one is applied with probability $(1 - M_t)$. In order to achieve higher diversity the mutated individual y^i_G is mated with the current population member x^i_G using a *crossover* operation to generate the *offspring* y^i_{G+1}. The parameters of solution y^i_{G+1} are randomly inherited from x^i_G or y^i_G determined by a parameter called *crossover factor* CF, i.e. if $r \leq CF$ (where r is a uniform random number in $[0, 1]$) then it is inherited from x^i_G otherwise from y^i_G. Finally the offspring is evaluated and replaces its parent x^i_G in next generation if and only if its fitness is better than that of its parent. This is the *replacement* process for producing new generation. And this process is repeated until a solution satisfying our criteria is found or a maximum number of generations have elapsed.

In TDE, the trigonometric mutation operation, a rather greedy search operator, makes it possible to straightforwardly adjust the balance between the convergence rate and the robustness through the newly introduced parameter, M_t. The greediness of the algorithm can be tuned conveniently by increasing or decreasing M_t. Experimental results have shown that TDE has good convergence properties, outperforms other well known EAs [7] and is effective in genetic network inference [16]. Because of these admirable properties, we have chosen TDE as optimization tool in our algorithm (explained in the next section) for the gene network reconstruction problem.

4.2 Proposed Algorithm

In this section we present the optimization algorithm that we have designed for estimating the parameters of S-system model of genetic networks. We explain the algorithm taking the sub-problem corresponding to gene i as an example.

For identifying the most robust regulatory interactions in the network and kinetic parameters for the regulations we applied double optimization in our algorithm. In double optimization a second phase of optimization is performed on different local solutions obtained in the first phase. Double optimization is useful for identifying essential parameters automatically and hence was found useful for detecting robust regulatory interactions in genetic networks [2, 10]. The two phases of our algorithm are as follows:

Phase 1: At first, we perform Γ repeated trials of optimization of the fitness function of (11) starting from different random initial solutions. In each of these trials, we performed optimization using a modified TDE algorithm with a hill climbing local search procedure (explained later). Each of these trial runs gives a solution of the sub-problem i.e. a set of parameters for the target gene. However some optimization trials may converge to some local optimum and may fail to infer the actual parameter set.

Phase 2: Since we assume some solutions in *Phase 1* are possibly local solutions, they may not identify all the target regulations and the parameter values may be significantly different in different solutions. Therefore, in order to obtain a more robust network structure and accurate pa-

rameter values we perform another optimization on the elite individuals from different trials of *Phase 1*. We select the best individual from each of the Γ trials and some randomly initialized individuals as initial population and perform optimization using the same fitness function and algorithm.

If the solutions obtained from different trials of *Phase 1* are local solutions they retain some essential regulations. So applying another optimization on these solutions we expect to identify all the correct regulations with accurate strengths and avoid the loss of any necessary interaction.

As mentioned earlier, the solution space of the problem contains many local optima which may lead the search algorithm to wrong directions and eventually the global solution may remain undetected. For locating the global optimal solution in such a search space we need to maintain population diversity. Mutation is the operator that has been traditionally used in EAs to introduce diversity in the population. TDE does not apply any immediate mutation operation, so we occasionally apply a mutation operation in our algorithm for higher diversity in the population. If the fitness of the elite individual does not improve for G_m generations then the mutation operation is evoked which mutates all the other individuals in the current generation. We applied the Gaussian mutation with mutation probability p_m. Gaussian mutation realizes the mutation operation by adding a random value from the Gaussian distribution. For mutating the *rate constants* of an individual the random numbers are drawn from a Gaussian distribution with mean $\mu_r = 0$ and standard deviation σ_r and for mutating the *kinetic orders* the random numbers are drawn from a distribution with mean $\mu_k = 0$ and standard deviation σ_k.

4.3 Hill Climbing Local Search

Incorporating problem-dependent heuristics, such as approximation algorithms, local search techniques, specialized recombination operators, etc., are often very useful in designing an effective global optimization technique. In order to obtain the skeletal network structure efficiently we embedded a local search method in our algorithm. Our local refinement procedure performs a hill climbing search operation around the best individual and a random individual of each generation for obtaining a sparser architecture.The hill climbing search operation on an individual $Indiv$ is shown below:

HCLS (*Indiv*)

1. SORT the *kinetic orders* (i.e. g_{ij} and h_{ij}) of $Indiv$ all together in ascending order of their absolute values. i.e. $|K(i)| \leq |K(i+1)|$ for $(i = 1, \cdots, 2N-1)$ and SET $i = 1$
2. Generate a new solution $Indiv'$ from $Indiv$ by setting $K(i) = 0$
3. IF $f(Indiv') \leq f(Indiv)$ SET $Indiv = Indiv'$
4. SET $i = i + 1$ and GOTO Step 2

This hill-climbing local search process allows us to identify the non-existing regulations by mutating the *kinetic orders* to zero in the increasing order of their strength and thus helps us to identify the skeletal network structure. And the restore capability of the greedy search also allows to recover from wrong elimination of any essential regulation. Hybridizing this hill-climbing search procedure with the TDE algorithm we can identify the sparse network structure efficiently and estimate the strengths of the regulations more accurately.

5. RECONSTRUCTION EXPERIMENTS AND RESULTS

To see how successfully the proposed method can reconstruct network topology and estimate kinetic parameters we evaluate it by simulation. We used two artificial networks of different dimensions and simulated those to obtain synthetic microarray data sets. And we applied our method to reverse engineer the networks from these data sets. The details of the experiments and the outcomes follow in the subsequent sections.

5.1 Small Scale Network Inference

As a first study, we tested our approach using a well studied small scale network model NET1. The system, consisting of five genes, adequately demonstrates different types of positive and negative mode of regulatory controls among the reactants. The target parameters for the system, listed in Table 1, are the same as found in many other studies [10, 12, 17, 25]. Choosing this network model we get a chance to compare our method to early approaches.

Generally, a single time series cannot provide insight into the mechanism of a dynamic system[24]. And multiple different candidate solutions evolve if the model parameters are estimated using insufficient amounts of time series data. Therefore we used $M = 10$ sets of time series data for ensuring sufficient amount of observed gene expression levels. The sets of time-series were obtained by solving (1) on the model of Table 1. Initial concentration level for each time series was generated randomly in $[0.0, 1.0]$. Sampling 11 points from each time-course we used $10 \times 11 = 110$ gene expression levels for each gene.

5.1.1 Experimental Setup

We performed the experiment under the following setup. The search regions of the parameters were $[0.0, 20.0]$ for α_i and β_i, and $[-3.0, 3.0]$ for g_{ij} and h_{ij}. The maximum allowed cardinality I was chosen to be 5, and the penalty coefficient c was 1000.0. The parameter values for the TDE algorithm were $F = 0.5$, $CF = 0.8$ and $M_t = 0.05$, population size was 60 and the maximum number of generations in each trial of *Phase 1* and in *Phase 2* was 850. In *Phase 1* we evolved 5 ($\Gamma = 1, \cdots, 5$) independent trial solutions from which we selected elite individuals for optimization in *Phase 2*. The parameter values for the mutation phase were $p_m = 0.01$, $\sigma_r = 3.0$ and $\sigma_k = 1.2$. In *Phase 1* of the optimization, $G_m = 100$ and in *Phase 2*, $G_m = 200$ were used. Our algorithm was implemented in Java language and the time required for solving each subproblem was approximately 10 minutes using a PC with a 1700 MHz Intel Pentium processor and 512 MB of RAM.

In order to reduce the computational burden, a structure skeletalizing was applied in a similar fashion used by Tominaga *et al.* [25]. If the absolute value of a parameter is less than a threshold value δ then structure skeletalizing resets it to zero. This process reduces the computational cost as well as helps to identify the zero valued parameters. In our experiment $\delta = 0.001$ was used. We used 5 repetitions for each experiment to assure soundness of our stochastic search algorithm.

Table 1: S-system parameters for network model NET1

Gene i	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}
1	5.0	0.0	0.0	1.0	0.0	-1.0	10.0	2.0	0.0	0.0	0.0	0.0
2	10.0	2.0	0.0	0.0	0.0	0.0	10.0	0.0	2.0	0.0	0.0	0.0
3	10.0	0.0	-1.0	0.0	0.0	0.0	10.0	0.0	-1.0	2.0	0.0	0.0
4	8.0	0.0	0.0	2.0	0.0	-1.0	10.0	0.0	0.0	0.0	2.0	0.0
5	10.0	0.0	0.0	0.0	2.0	0.0	10.0	0.0	0.0	0.0	0.0	2.0

Table 2: Inferred parameters for network model NET1

Gene i	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}
1	4.990	0.000	-0.008	0.980	-0.004	-0.997	10.003	1.978	0.000	0.000	0.000	0.000
2	10.051	1.995	0.004	0.009	0.002	-0.002	10.060	0.000	1.998	0.012	0.000	0.001
3	9.936	-0.004	-1.001	-0.001	0.000	0.000	9.937	-0.004	-1.001	2.007	0.000	0.001
4	8.032	0.000	-0.011	1.949	0.000	-0.996	10.153	0.000	0.007	0.000	1.972	0.000
5	10.011	0.000	0.003	0.023	2.002	-0.009	9.992	0.006	0.000	0.002	0.000	1.990

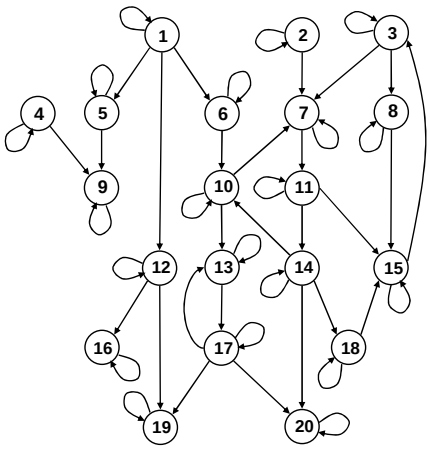

Figure 1: Structure of Genetic Network NET2

5.1.2 Result

Table 2 shows the parameters estimated by our algorithm in a typical run. As shown in Table 2 our method was able to attain the exact network topology and parameter values were almost the same as the target values. Many of the zero valued parameters were identified correctly and the values of the others are close enough to zero to indicate (possible) false positive interactions.

5.2 Medium Scale Network Inference

In this experiment we investigate the performance of our algorithm using a 20 gene network NET2. The topology of the network was created randomly with a maximum in-degree limit and then we formulated the network in S-system formalism. Figure 1 shows the network structure and Table 3 contains the parameters which were chosen arbitrarily. We simulated the network model NET2 with random initial concentrations chosen from $[0.0, 1.0]$ to generate 20 sets of synthetic microarray data. We used 11 samples from each time course data and used them for inferring the model parameters.

5.2.1 Experimental Setup

Since ours is a stochastic search process we employed our algorithm to reconstruct the target network in five repeated runs under the following conditions. The search regions of the parameters were $[0.0, 20.0]$ for α_i and β_i, and $[-3.0, 3.0]$ for g_{ij} and h_{ij}. The population size was 210 and the maximum number of generations in each trial of *Phase 1* and in *Phase 2* was 2400. Other conditions were the same as in Sec. 5.1.1. The average time for solving each sub-problem was approximately 13.5 hours using a PC with a 1700 MHz Intel Pentium processor and 512 MB of RAM.

Table 3: Target parameters for NET2

α_i β_i	10.0
$g_{i,j}$	$g_{3,15} = -0.7$, $g_{5,1} = 1.0$, $g_{6,1} = 2.0$, $g_{7,2} = 1.2$, $g_{7,3} = -0.8$, $g_{7,10} = 1.6$, $g_{8,3} = -0.6$, $g_{9,4} = 0.5$, $g_{9,5} = 0.7$, $g_{10,6} = -0.3$, $g_{10,14} = 0.9$, $g_{11,7} = 0.5$, $g_{12,1} = 1.0$, $g_{13,10} = -0.4$, $g_{13,17} = 1.3$, $g_{14,11} = -0.4$, $g_{15,8} = 0.5$, $g_{15,11} = -1.0$, $g_{15,18} = -0.9$, $g_{16,12} = 2.0$, $g_{17,13} = -0.5$, $g_{18,14} = 1.2$, $g_{19,12} = 1.4$, $g_{19,17} = 0.6$ $g_{20,14} = 1.0$, $g_{20,17} = 1.5$, other $g_{i,j} = 0.0$
h_{ij}	1.0 if $(i = j)$, 0.0, otherwise

Table 4: Estimated parameters for NET2

Gene 1	$\alpha_1 = 9.90$, $g_{1,3} = 0.002$, $g_{1,20} = 0.001$, $\beta_1 = 9.87$, $h_{1,1} = 1.007$, $h_{1,3} = 0.009$, $h_{1,13} = 0.003$
Gene 5	$\alpha_5 = 9.99$, $g_{5,1} = 1.026$, $g_{5,2} = -0.007$, $g_{5,20} = -0.007$, $\beta_5 = 10.03$, $h_{5,1} = 0.030$, $h_{5,5} = 0.990$, $h_{5,9} = -0.007$, $h_{5,13} = 0.007$, $h_{5,15} = -0.006$, $h_{5,16} = -0.003$
Gene 15	$\alpha_{15} = 12.01$, $g_{15,8} = 0.413$, $g_{15,11} = -0.904$, $g_{15,18} = -0.849$, $\beta_{15} = 12.804$, $h_{15,8} = -0.058$, $h_{15,15} = 0.876$
Gene 20	$\alpha_{20} = 7.56$, $g_{20,7} = -0.002$, $g_{20,14} = 1.246$, $g_{20,17} = 1.717$, $g_{20,20} = -0.181$, $\beta_{20} = 7.60$, $h_{20,7} = -0.010$, $h_{20,10} = 0.001$, $h_{20,17} = -0.142$, $h_{20,19} = -0.002$, $h_{20,20} = 1.03$

5.2.2 Results

In each run our method successfully predicted the exact network architecture and also determined the type of regulation (activation/inhibition) correctly. The process also estimated the kinetic parameters with high accuracy. Table 4 shows a typical estimation of parameters for genes with different number of regulators. Again it can be stated that the method successfully identified the network dynamics but also falsely predicted some regulators which can be easily ignored because of the strength of their regulations.

6. DISCUSSION

One of the major challenges the emerging field of Systems Biology facing, is identifying the sophisticated mechanism that regulates gene expression. Among different available models, S-systems has been found to provide valid representations in a large number of theoretical and practical studies. Moreover, its parameters have well-defined meanings in biological context, which makes its application more realistic for modeling metabolic networks. Nevertheless, due to its tightly coupled form it has found limited application to larger network. Reformulation of the model in decoupled form has made its application computationally tractable in networks consisting of many metabolites.

In this work we have presented a method for inferring the transcriptional regulations in a network represented in decoupled S-system formalism. Using an evolutionary algorithm, we predicted the kinetic parameters of the system from time series gene expression data. While searching for the optimal set of parameters using our evolutionary algorithm, we evaluated the candidate solutions using an AIC based fitness criterion rather the conventional MSE based fitness function. In the proposed fitness evaluation function we extended the penalty term of AIC. The purpose of this additional penalty term is to facilitate the selection of models with sparse network architecture. Following the guideline of previous works, we designed this penalty term such that it will penalize the fitness score of a candidate network model if it has more regulators than the maximum allowed in-degree of that network. Therefore, this penalty term remains silent as long as the number of regulators of a gene does not exceed the maximum given limit. Otherwise it penalizes the competing model and thus helps to identify the skeletal architecture. In this additional term there is an additional parameter c which has been given a value 1000. Choice of this parameter value is very straightforward and was determined as follows. As mentioned in section 5.1.1 our algorithm performed a structure skeletalizing for reducing the computational burden by setting a parameter to zero if its absolute value is less than $\delta = 0.001$. We want to penalize the fitness score effectively for all additional regulators (that lie beyond the threshold of maximum allowed in-degree) until their values go down below δ. Therefore such a value for the parameter 'c' is quite natural and was found useful.

To further investigate the necessity of the additional penalty term in (11) for obtaining the skeletal structure of the network we perform additional experiments. We reverse engineer the model of NET1 in the exact environment of section 5.1.1 except using the original AIC as fitness evaluation criterion. Among the five repeated runs of the experiment the best results are shown in Table 5. It can be found from Table 5 that, using the original AIC as fitness evaluation criterion, the algorithm could essentially identify the target network topology but could not estimate the parameter values very accurately. Some of the parameter values were pretty distant from the target. Moreover some false positive regulations had strengths very strong that can not be ignored. On the other hand, the same method using the fitness function of (11) could predict the parameter values with high accuracy. We believe these results can be helpful to justify the usefulness of the proposed fitness function of (11) for evaluating candidate solutions in an evolutionary approach for inferring the model parameters.

In the proposed evolutionary algorithm we used TDE as the core optimization unit. TDE has found a wide range of real world applications where we need to search for an optimal set of real valued parameters. Therefore, we designed our algorithm implanting TDE in the kernel and taking several other issues in consideration, such as: we performed double optimization for selecting robust parameter values, included a hill-climbing local search procedure for accelerating the identification of the skeletal network topology and embedded a mutation-phase to maintain the diversity in the population for finding the global optimal solution. We experimented with networks of different dimensions and properties to evaluate the performance of the proposed technique. The reconstruction method identified all the regulatory interactions with correct properties (activation or repression) and also estimated the strength of the regulations with high accuracy.

When we compared our proposed method to previous works that used MSE based scoring it was found it performed better in the experiment of reconstructing NET1. Compared to the method of [17] the estimated parameters were more accurate using the same number of gene expression data. And compared to the method of [11] the proposed technique performed better both in terms of computational efficiency and parameter estimation.

7. CONCLUSION

As microarray data is becoming more easily available, identifying the regulatory machinery in a gene circuit is becoming more desirable compared to a clustering method for grouping genes with similar patterns. Genetic network estimation using S-system model is often formulated as an optimization problem where the MSE between the estimated expression levels and experimental expression levels is used as the fitness evaluation criterion which should be minimized for identifying the optimal structure and kinetic parameters.

AIC is a long existing criterion for evaluating alternative models and making a selection among them. In this work we proposed a new information criteria based fitness evaluation function for reverse engineering genetic circuits from time series data using evolutionary algorithms. We also developed an improved evolutionary algorithm for reconstructing the underlying regulatory architecture and inferring effective kinetic parameters for the network using the proposed fitness function. Our methodology was tested in simulation using small-scale and medium networks and the method reconstructed the target networks structures exactly and estimated parameter values very accurately. Since our proposed method is purely computational, it can be readily applied to other network reconstruction problems and the proposed evolutionary algorithm is general enough for using in other model of biological network. We are currently investigating the usefulness of the proposed fitness function in estimating model parameters from data corrupted with noise and in our future work we will apply our method for inferring large-scale genetic networks from real microarray data.

8. REFERENCES

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *Second Int. Symposium on Information Theory*, pages 267–281, 1973.

[2] S. Ando and H. Iba. Construction of genetic network using evolutionary algorithm and combined fitness

Table 5: Inferred parameters for NET1 using original AIC

Gene i	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}
1	2.365	-0.610	-0.003	1.328	0.014	-1.326	7.656	2.741	0.008	-0.299	0.017	0.362
2	9.671	2.009	-0.017	0.014	0.002	-0.005	9.689	-0.082	2.067	0.023	0.001	0.000
3	7.610	-0.034	-0.990	-0.182	-0.006	-0.007	7.615	-0.042	-0.986	2.505	-0.008	-0.011
4	6.805	-0.018	-0.021	2.060	-0.088	-1.039	8.813	-0.093	-0.011	-0.232	2.297	0.141
5	9.133	-0.037	0.005	0.037	2.106	-0.085	9.108	-0.060	0.013	0.010	-0.071	2.092

function. In *Genome Informatics 14*, pages 94–103, June 2003.

[3] A. Arkin, J. Ross, and H. H. McAdams. Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected escherichia coli cells. *Genetics*, 149:1633–1648, August 1998.

[4] M. Arnone and E. Davidson. The hardwiring of development: organization and function of genomic regulatory systems. *Development*, 124(10):1851–1864, May 1997.

[5] P. D'haeseleer, X. Wen, S. Fuhrman, and R. Somogyi. Linear modeling of mrna expression levels during cns development and injury. In *Pacific Symposium on Biocomputing 4*, pages 41–52, 1999.

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

[7] H. Y. Fan and J. Lampinen. A trigonometric mutation operation to differential evolution. *Journal of Global Optimization*, 27(1):105–129, September 2003.

[8] H. Iba and A. Mimura. Inference of a gene regulatory network by means of interactive evolutionary computing. *Information Sciences*, 15(3):225–236, September 2002.

[9] S. Kauffman. *The Origins of Order: Self-organization and Selection in Evolution*. Oxford University Press, New York, 1993.

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

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

[12] S. Kimura, K. Ide, A. Kashihara, M. Kano, M. Hatakeyama, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu, and A. Konagaya. Inference of s-system models of genetic networks using a cooperative coevolutionary algorithm. *Bioinformatics*, 21(7):1154–1163, 2005.

[13] Y. Maki, D. Tominaga, M. Okamoto, S. Watanabe, and Y. Eguchi. Development of a system for the inference of large scale genetic networks. In *Pacific Symposium on Biocomputing 6*, pages 446–458, 2001.

[14] Y. Maki, T. Ueda, M. Okamoto, N. Uematsu, K. Inamura, K. Uchida, Y. Takahashi, and Y. Eguchi. Inference of genetic network using the expression profile time course data of mouse p19 cells. In *Genome Informatics 13*, page 382383, December 2002.

[15] H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano. Hybrid petri net representation of gene regulatory network. In *Pacific Symposium on Biocomputing 5*, pages 338–349, 2000.

[16] N. Noman and H. Iba. Inference of gene regulatory networks using s-system and differential evolution. In *Genetic and Evolutionary Computation Conference (GECCO) Proceedings*, pages 439–446, June 2005.

[17] N. Noman and H. Iba. Reverse engineering genetic networks using evolutionary computation. In *Genome Informatics 16(2)*, pages 205–214, December 2005.

[18] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical recipies in C, second edition*. Cambridge University Press, 1995.

[19] M. Savageau. Biochemical systems analysis. i. some mathematical properties of the rate law for the component enzymatic reactions. *Theoretical Biology*, 25:365–369, 1969.

[20] M. Savageau. Biochemical systems analysis. ii. the steady-state solutions for an n-pool system using a power-law approximation. *Theoretical Biology*, 25:370–379, 1969.

[21] M. Savageau. *Biochemical Systems Analysis. A Study of Function and Design in Molecular Biology*. Addison-Wesley Publishing Company, Reading, Massachusetts, 1976.

[22] M. Savageau. Power-law formalism: A canonical nonlinear approach to modeling and analysis. In *World Congress of Nonlinear Analysts, 92, Vol.4*, pages 3323–3334, 1996.

[23] R. Storn and K. V. Price. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, December 1997.

[24] F. Streichert, H. Planatscher, C. Spieth, H. Ulmer, and A. Zell. Comparing genetic programming and evolution strategies on inferring gene regulatory networks. In *Genetic and Evolutionary Computation Conference (GECCO) Proceedings*, pages 471–480. Springer, June 2004.

[25] D. Tominaga, N. Koga, and M. Okamoto. Efficient numerical optimization algorithm based on genetic algorithm for inverse problem. In *Proceedings of Genetic and Evolutionary Computation Conference*, pages 251–258. Van Nostrand Reinhold, July 2000.

[26] E. O. Voit and J. S. Almeida. Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics*, 20(11):1670–1681, 2004.

[27] S.-C. Wang. Reconstructing genetic networks from time ordered gene expression data using bayesian method with global search algorithm. *Journal of Bioinformatics and Computational Biology*, 2(3):441–458, 2004.