Inferring Gene Regulatory Networks by Incremental Evolution and Network Decomposition

Wei-Po Lee[1, *]　　　　Yu-Ting Hsiao[1]

[1]Department of Information Management,
National Sun Yat-sen University, Kaohsiung, Taiwan

Abstract　Constructing genetic regulatory networks from expression data is one of the most important issues in systems biology research. However, building regulatory models manually is a tedious task, especially when the number of genes involved increases with the complexity of regulation. To automate the procedure of network construction, we develop a methodology to infer S-systems as regulatory systems. Our work also deals with the scalability problem by an incremental evolution strategy and a network decomposition method with several data analysis techniques. To verify the presented approaches, experiments have been conducted and the results show that they can be used to infer gene regulatory networks successfully.

Keywords　gene regulatory network; genetic algorithm; incremental evolution; gene clustering; network decomposition

1　Introduction

Gene regulatory networks (GRNs) play key roles in cellular metabolism during the development of living organisms. They dynamically orchestrate the level of expression for each gene in the genome by controlling whether and how the gene can be transcribed into RNA [1]. To investigate into the system dynamics of GRNs, biologists and computational scientists have been working on creating and exploring predictive dynamical models of complex biological systems in living cells. With the network models, we can uncover some complex behavior patterns by constructing networks from measured time series data, and then analyzing and studying the interactions between interconnected components in a network.

Traditionally, to reconstruct a gene regulatory network from experimental data, one can begin with building an initial model, simulating the system behaviors for a variety of experimental and environmental conditions, and then comparing the predictions with the observed gene expression data to give an indication of the adequacy of the model. If the experimental data is considered reliable, and the observed and predicted system behavior does not match the data, the model must be revised. The activities of manually constructing and revising models of the regulatory network, simulating the behavior of

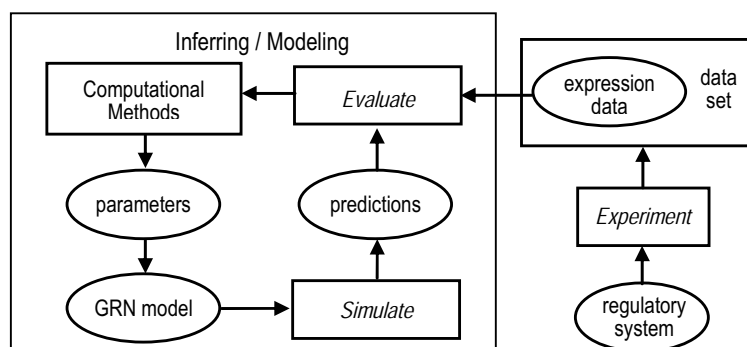---

*wplee@mail.nsysu.edu.tw

Figure 1: The computational modeling of a gene regulatory network

the system, and testing the resulting predictions are repeated until an adequate model is obtained.

As the above procedure for network modeling takes a considerable amount of time, an automated procedure is thus advocated. Reverse engineering is a paradigm with great promise for analyzing and constructing gene regulatory networks [3][4]. It is an effective way to utilize experimental data to determine the underlying network of a given model. The procedure involves altering the gene network in some way, observing the outcome, and using mathematics and logic (i.e., computational methods) to infer the underlying principles of the network. Fig. 1 illustrates the procedure of a reverse engineering approach with computational methods for modeling GRNs from measured expression data.

In this work, we establish a methodology that takes S-system model to represent GRNs and exploits evolutionary algorithms to reconstruct regulatory systems from gene expression profiles. We also propose two approaches to deal with the scalability problem. One is *incremental evolution* that involves a dynamic selection strategy to iteratively fix values for some of the system parameters and evolve others, and then gradually the overall solution can be obtained. The other is *network decomposition* in which a clustering-based method with some data analysis techniques for feature extraction is applied to develop GRNs hierarchically. To verify the presented approaches, three series of experiments have been conducted to demonstrate how it works.

2 Background

In the work of GRN reconstruction, many models have been proposed. They can mainly be categorized into two types that use discrete and continuous variables respectively. The first type of GRN models assumes that genes only exist in discrete states. In this approach, the approximation is usually implemented by Boolean variables in which the gene is in either on or off state. This type of models includes Boolean networks and Bayesian networks. Boolean networks are easy to simulate in a cheaper computational cost, but they are not able to capture some system behaviors [3][5]. Bayesian networks explicitly establish probabilistic relationships between nodes [6][7]. They have rich statistics and probability semantics, but learning network structure for such models is computationally expensive. In addition, Bayesian models are inherently static. As the

directed network graphs are acyclic by definition, there can be no auto-regulation and no time-course regulation.

The second type of GRN models uses continuous variables to simulate fully biochemical interactions with stochastic kinetics. One of the popular continuous variables models is based on differential equations that can describe more accurately the system dynamics of a GRN [8][9]. Compared to discrete variables models, the differential equations models can represent the underlying physical phenomena due to its continuous variables. In addition, there are many theories of system analysis and of control design on dynamical systems to support this type of models. The other commonly used continuous variables model is neural network-based model, among which the recurrent neural networks are the most successful ones [10][11]. The models of continuous variables are continuous in time, and their non-linear characteristics provide information about the principles of control and natural interactions of elements of the modeled system.

As mentioned above, different computational methods have been advocated to reconstruct network models (i.e., to determine network structures and parameters) from the expression data correspondingly. From the literature (e.g., [2][3]) it can be seen that work in modeling GRNs shared similar ideas in principle. However, depending on the research motivations behind the work, different researchers have explored the same topic from different points of view; thus the implementation details of individual work are different. Therefore, instead of subjectively arguing which approach is better to offer for network reconstruction, our work here mainly focuses on how to model large scale networks. We establish a methodology that takes non-linear differential equations-based model to represent GRNs and exploits genetic algorithms (GAs) to infer regulatory systems from collected expression data. Different from other works in network reconstruction, we also propose two approaches to tackle the scalability problem. The following sections describe how we employ GA with scalable methods to model large GRNs.

3   Inferring Gene Regulatory Networks

3.1   Network Model

In a GRN, the network structure is an abstraction of the chemical dynamics of this system, describing the manifold ways in which one substance affects all the others to which it is connected. The network nodes are genes that can be regarded as functions obtained by combining basic functions upon the inputs. As can be seen, the behavior expressions of a GRN network are in fact coordinated patterns of activity in time and space. Therefore this kind of networks can be regarded as dynamical systems that are perturbed by their interaction with the environment. To have such characteristic, it is important that the chosen network model for modeling the expression data must be able to produce intrinsic dynamical behavior. Differential equations-based system models are appropriate choices to work as regulatory networks, as they can accurately simulate the corresponding system dynamics.

Many models based on differential equations have been proposed, including the traditional linear ordinary differential equations and the non-linear power law ones. S-system is a kind of power law model. It consists of a particular set of tightly coupled non-linear differential equations in which the component processes are characterized by power law functions. The S-system model has been considered suitable to characterize biochemical

network systems and capable to simulate the regulatory system dynamics. In this work, we adopt this model to represent GRNs. In the S-system model, the systematic structure can be described as:

$$\frac{dX_i}{dt} = \alpha_i \prod_{j=1}^{N} X_j^{g_{i,j}} - \beta_i \prod_{j=1}^{N} X_j^{h_{i,j}}$$

Here X_i is the expression level of gene i and N is the number of genes in a genetic network. The non-negative parameters α_i and β_i are rate constants that indicate the direction of mass flow. The real number exponents $g_{i,j}$ and $h_{i,j}$ are kinetic orders that reflect the intensity of interaction from gene j to i. The above set of parameters defines an S-system model.

It should be noted that the above non-linear ordinary differential equations are hard to solve. To infer an S-system model, it is necessary to estimate all of the $2N(N+1)$ parameters simultaneously. As can be observed, it is too difficult for the traditional optimization methods to determine the large number of parameters involved in a GRN, especially when there is only limited number of data samples available. Though many intelligent computing techniques for parameter approximation, such as evolutionary algorithms [8][9], have been proposed to derive the solutions, more efficient approaches are still in need to resolve the high dimensional problem. Dimension reduction provides a useful strategy to deal with problems with high dimensional solution space. Therefore, in this work we not only employ Genetic Algorithms to infer networks, but also take the strategy of dimension reduction to develop two approaches to evolve large gene networks. One is incremental evolution that derives the overall solutions from partial solutions gradually. The other is to tackle the problem in a "divide and conquer" manner that involves a network decomposition procedure to reduce the task complexity. The details are described in the sections below.

3.2   Inferring GRNs by GA with Local Search

To evolve GRNs, we take a direct encoding scheme in which the key network parameters used to define a system model (as described in the above equation) are represented as a linear string chromosome of floating numbers. The tournament selection strategy is used here to choose parents for reproduction. Also the genetic operators useful for real-coded GA, including arithmetical crossover and non-uniform mutation [12], are used to change numerical values of the chromosomes to evolve the parameters. As in a curve-fitting problem, the goal here is to minimize the accumulated discrepancy between the gene expression data recorded in the dataset (desired values) and the values produced by the model determined by GAs (actual values). That is, the fitness function is defined directly as the mean squared error over the time course as:

$$f = \sum_{k=1}^{N} \sum_{t=1}^{T} \left\{ \frac{X_{k,i}^{a}(t) - X_{k,i}^{d}(t)}{X_{k,i}^{d}(t)} \right\}^2$$

in which $X_{k,t}^{a}$ is a desired expression level of gene i at time t, $X_{k,t}^{d}$ is an actual value obtained from the model, N is the number of genes in the network, and T is the number of the data points measured for a gene. A small penalty term measuring the connection between

genes can be added to the fitness function to reduce the search space, but it is not used in this work as that is not the main focus here.

Because traditional GAs are global search methods that mainly concentrate on exploring the solution space without taking local information, they lack the ability of local fine-tuning. Therefore, many researchers have proposed to combine GA with a local search technique to exploit the local information to determine the promising search direction in the search space. To enhance the search performance in solving the above optimization problem, we have implemented the simplex method, a popular algorithm for numerical solution of the linear programming problem [13], as a local search technique with the GA.

3.3 Incremental Evolution

In the task of inferring GRNs, when the number of genes in a regulatory network and the interactions between the genes increase in respect to the increasing functional complexity the network has to deal with, the number of network parameters will increase rapidly and thus makes the search difficult. That is, the direct GA described above can easily get trapped in an unfruitful region of the search space. To deal with the scalability problem, we adopt the concept of incremental evolution. The underlying principle is that a population is first evolved to solve an easier version T' of the original complex task T, in which the solution region of T is more accessible from region T'. More task versions with incremental complexity can be arranged so that the original task can be achieved progressively.

The major focus to realize incremental evolution is to formulate a scheme to transfer the goal task into another task that is more evolvable. In the process of task transformation, the underlying structure of the environment and the goal of the overall task must be preserved. This can be achieved by arranging the task sequence manually or alternatively by an automated procedure. In this work, we modify the cutting plane mechanism used in the high-dimension function optimization problem [14], to develop an adaptive strategy to perform incremental evolution automatically.

During the process of evolution, our strategy intends to fix some gene variables (i.e., the network parameters related to these genes), and to evolve the other gene variables. That is, to evolve partial solution incrementally and then to gain the overall solution gradually. The selection of gene variables to be fixed needs to consider their individual discriminating ability. Fixing a gene variable with higher discriminating ability in an earlier stage can have better chances to direct the candidate solutions to the potential final solution and it is thus more likely to converge to a better solution. Therefore, to select the gene variables to be fixed, we define an evaluation function f_i (different from the fitness function f, which is the summation of all f_i) to record the accumulated mean squared error for each gene variable X_i, and choose gene candidates accordingly. With the above considerations, the main steps of our GA-based modeling are modified in the following ways:

1. Initializing a population in which an individual is constituted by all network parameters.
2. Running an evolution experiment and calculating the evaluation values f_i for each gene variable X_i.

3. Adding gene variables with evaluation values less than a threshold ε (i.e., $f_i < \varepsilon$) to a candidate list, and then choosing k gene variables with smallest evaluation values from the candidates to fix.

4. Repeating the above steps a few times to make better decisions (i.e., to find a better candidate list).

5. Adding a small constant to k (to fix more gene variables), if more candidates are obtained in step 3.

6. Going back to step 2 and evolving other gene variables.

In the above flow, the threshold (tolerance) can be adjusted to construct a more appropriate candidate list. If there is no candidate to be selected and fixed in step 3, the evolutionary process then operates as in the original GA. As indicated in step 5, when more and more gene variables produce desired behaviors (i.e., with very small error), the number of gene variables to be fixed will increase gradually. The experimental section will show that this strategy can efficiently improve the search and obtain better solutions.

3.4 Network Decomposition

In addition to the above search-based technique, we also develop a clustering-based method to decompose search space to solve more complicated reconstruction task. Clustering is a useful exploratory technique for the analysis of gene expression data. The hypothesis of using gene clustering is that gene in a cluster may share some common functions or regulatory elements and they can thus be considered and modeled together. In our method, a clustering technique is firstly employed to group the genes into tightly coupled small-scale networks, based on the analysis of their corresponding expression data, and the small networks can be decomposed again in the similar way until the resulting networks can be directly modeled. Then the small networks are directly evolved from the expression data. Once all the small networks have been obtained, they can be regarded as self-contained system components of the original system, and assembled together manually or by the learning algorithms described.

In our current work, the self-organization feature map (SOM) method is adopted for gene clustering. Before a clustering method is applied to the expression data, some features on the data set have to be decided so that the clustering method can find the relationships between the data accordingly. As there are no predefined data features to be selected in gene expression profiles, a feature extraction procedure needs to be performed. Here we use the wavelet transform (WT) technique to extract data features from the waveforms derived from the gene expression data of different time points.

The WT theory has been widely used in many signal-processing applications [15]. WT decomposes a signal into a set of basis functions called wavelets. It involves representing a time function in terms of simple and fixed building blocks, termed wavelets. These building blocks are actually a family of functions derived from a single generating function (i.e., the mother wavelet) by translation and dilation operations. It is known that the WT is more suitable in analyzing non-stationary signals, since it is well localized in time and frequency [16]. With its important ability on data manipulation, WT can compress an original signal that consists of many data points, into a few parameters called wavelet coefficients that characterize the behavior of the signal. The wavelet coefficients can be computed by using the discrete wavelet transform. The computed wavelet coeffi-

cients provide a compact representation that shows the energy distribution of the signal in time and frequency. Therefore, the wavelet coefficients derived from the time-varying gene regulatory signals can be used as features of the signals for gene clustering.

Fig. 2 is the typical result of wavelet transform for a certain gene (produced by MAT-LAB Wavelet toolbox), in which s is the original gene expression data, a_4 is the wavelet approximation (taken from the Daubechies function with wavelets of order 4) by the relevant subsequences, and d_1 to d_4 are the wavelet detailed subsequences (with four levels multi-resolution analysis). The coefficients of the high frequent wavelet subsequences are then used as data features for SOM clustering.
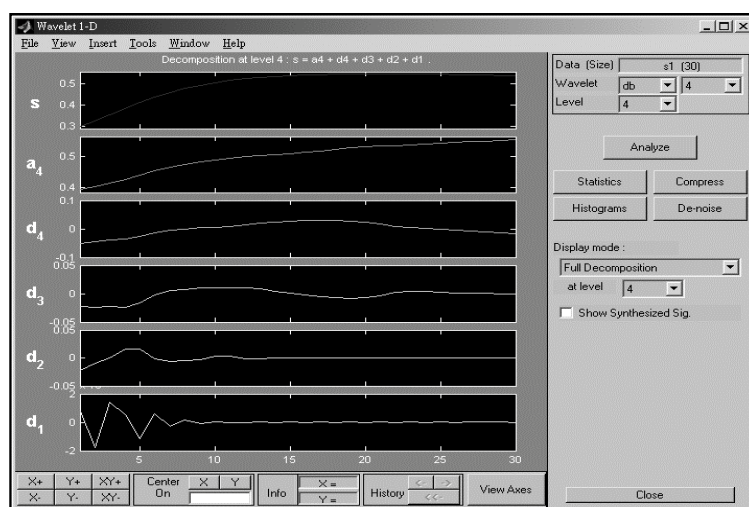

Figure 2: The wavelet transform for the expression data

4   Experiments and Results

To evaluate the proposed approaches, we conduct three series of experiments. The first series is to examine whether GA can evolve the S-system model from a given set of time series data. The second series is to investigate whether the proposed incremental evolution strategy can be used to reconstruct relatively large size GRNs. Finally, in the third series of experiments our approach is coupled with a gene clustering technique to model complicated networks with even more gene nodes.

4.1   Evolving GRNs

The data set used in the first series of experiments is the one reported in [17], which is the expression data of a metabolic network consisting of three substances (X_1, X_2, and X_3 in the equations below). As described in [17], the target network is a part of the biological phospholipids pathway, and their experimental data was derived from the E-cell simulation environment (i.e., a software package for cellular and biochemical modeling and simulation, see [18]). This network can be described approximately as:

$$\dot{X}_1 = -10.3176X_1X_2$$
$$\dot{X}_2 = 9.7149X_1X_3 - 17.5084X_2$$
$$\dot{X}_3 = -9.7018X_1X_3 + 17.4766X_2$$

The GA presented in section 3.2 was used to learn a network model from the expression data. Fig. 3 shows the network behaviors of the original and evolved networks, in which the x-axis represents time step (for collecting data) and y-axis, the concentration of different gene components. Fig. 4 is the fitness curve of the best individuals during a typical run in evolving networks. They indicate that the network model can be evolved successfully in which almost identical system behaviors can be obtained.
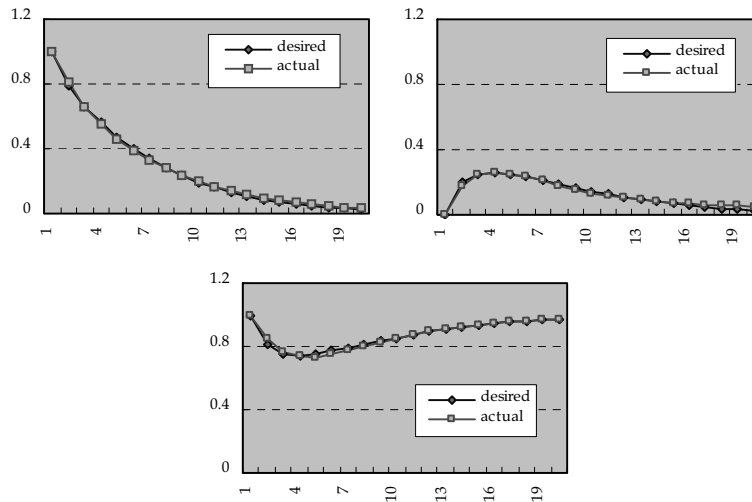
Figure 3: Behaviors of the target and evolved networks

4.2   Performance of Incremental Evolution

As can be expected, when the number of network parameters increases, the task of inferring network will become more and more difficult to achieve. This section demonstrates how the presented incremental evolution approach can improve the performance of inferring network for relatively large networks. The data set used in the experiment is an artificial dataset obtained from the well-known GRN simulation software Genexp (reported in [10]). A ten nodes gene network was defined and the simulation was run for 30 time steps for data collection. After that, GAs with and without the proposed incremental evolution strategy were used to evolve the network model reversely from the same data, respectively. Fig. 5 is a typical example comparing the two evolutionary approaches with and without incremental evolution strategy, which indicating their corresponding fitness curves of the best individuals during the runs. As can be seen in this figure, incremental evolution performs better in network modeling. Fig. 6 presents the data collected from the simulation and the expression data generated by the network inferred by the incremental evolution for the example. Again, the x-axis and y-axis represent time step and
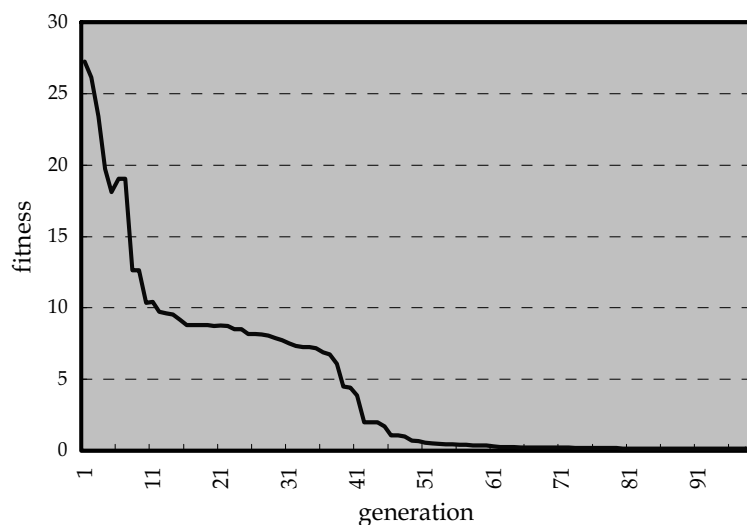
Figure 4: The fitness curve in evolving a three nodes network

the concentration of genes, respectively. Though this is not a perfect match in data fitting, it can be observed that the behavior of the evolved network is very similar to the original one, in which many of them have almost identical data sequences. This is satisfactory because the data set is small and the number of system parameters is large in fact.

4.3   Network Decomposition by Gene Clustering

To model large systems with more genes, the dataset available is usually not sufficient to determine accurately the interactions between all genes in a given data set. Hence, it is thus important to be able to construct a coarse-grained description of the system at first. This section demonstrates how the clustering method can help inferring coarse-grained network models from data. The dataset used in this set of experiments is a real experimental data set Rat CNS (central nervous system), taken from [19]. This dataset includes expression data of 112 genes collected from 9 time points of different phases (embryonic, postnatal, and adult). To reconstruct the original network from these time series data, the gene clustering method described in section 3.4, including the procedures of wavelet transform and SOM, was used to group genes. Among the 112 genes data, 103 of them were categorized into 6 different clusters and 9 genes did not belong to any cluster. One of our clusters consisting of 19 genes is very similar to the one reported in a previous study dealing with rat CNS data [19] (containing the 17 genes cluster in fact). To be consistent with the previous study and to preserve the meaning of the cluster as the original work, we decided to use the 17 genes cluster reported in [19] as the target network to be reconstructed.

Since the genes within the same cluster have been closely related, it is not practical to group them by the same clustering method again. Therefore, once the above target network (i.e., the one with 17 genes) has been determined, the genes were decomposed into four sub-groups according to the mutual information (often used to distinguish the
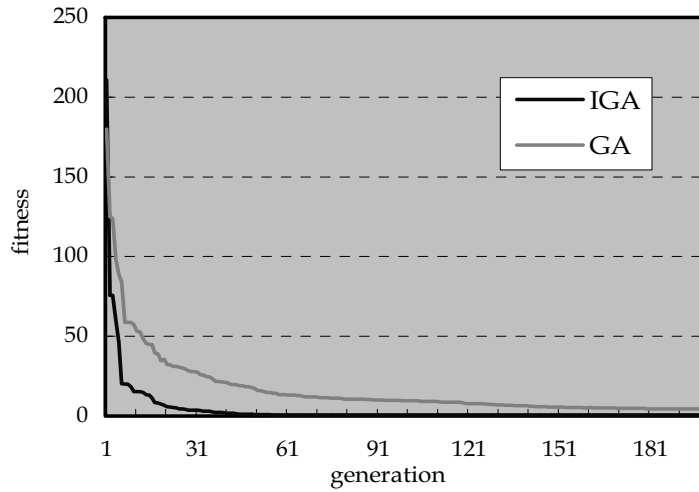
Figure 5: Performance of GAs with and without an incremental evolution strategy

close relationships between genes) between them, in which some genes belonged to more than one sub-group. Table 1 lists the details of the decomposed results in which the four sub-groups have 8, 5, 4, 5 nodes respectively. Then the GA was employed to build the 4 subnets and afterward, the target network. Fig. 7 shows four sets of behaviors of the original (left) and evolved (right) networks group by group. Again, very similar behaviors between the two sets of networks can be obtained. It indicates that the proposed network decomposition approach can be efficiently and successfully used to model networks with relatively large size.

Table 1: The details of each sub-group

sub-group	Gene Names	#genes
a	NFH, NFM, MOG, GRg1, NGF, Afgf, GFAP, cfos	8
b	S100beta, mGluR1, CNTF, GFAP, cfos	5
c	ChAT, NMDA2A, Bfgf, MOG	4
d	mAChR4, cjun, IP3R2, GFAP, cfos	5

5 Conclusions and Future Work

To construct GRNs from gene expression profiles is one of the most important issues in systems biology research. Many models have been proposed to simulate GRNs, and different computational methods have also been developed to reconstruct networks. In this work, regardless of which model and method is most suitable for network reconstruction, we mainly emphasize the importance of establishing a practical approach that can model GRNs and is scalable for inferring large-scale networks. As S-systems can work as
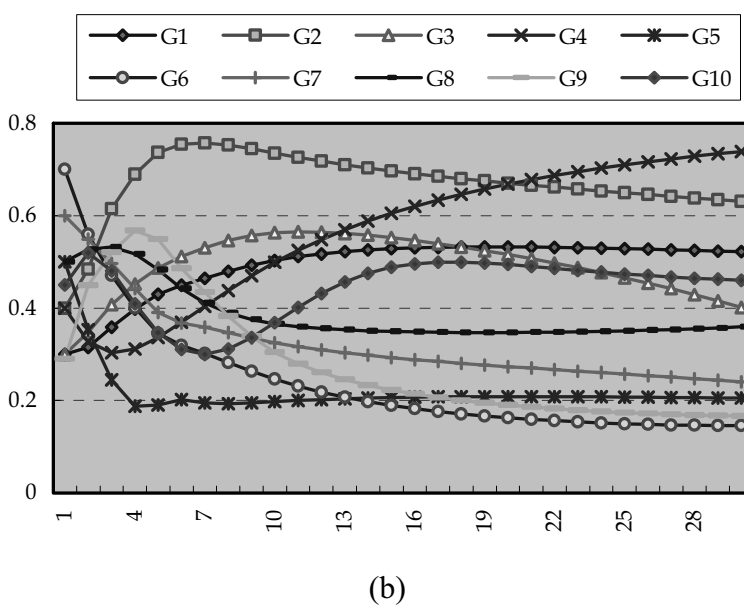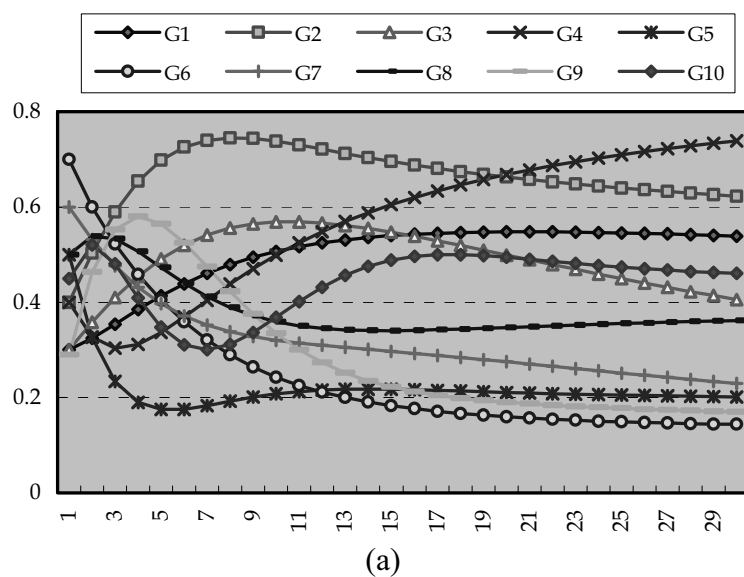
(a)

(b)

Figure 6: Behaviors of the target (a) and evolved (b) networks
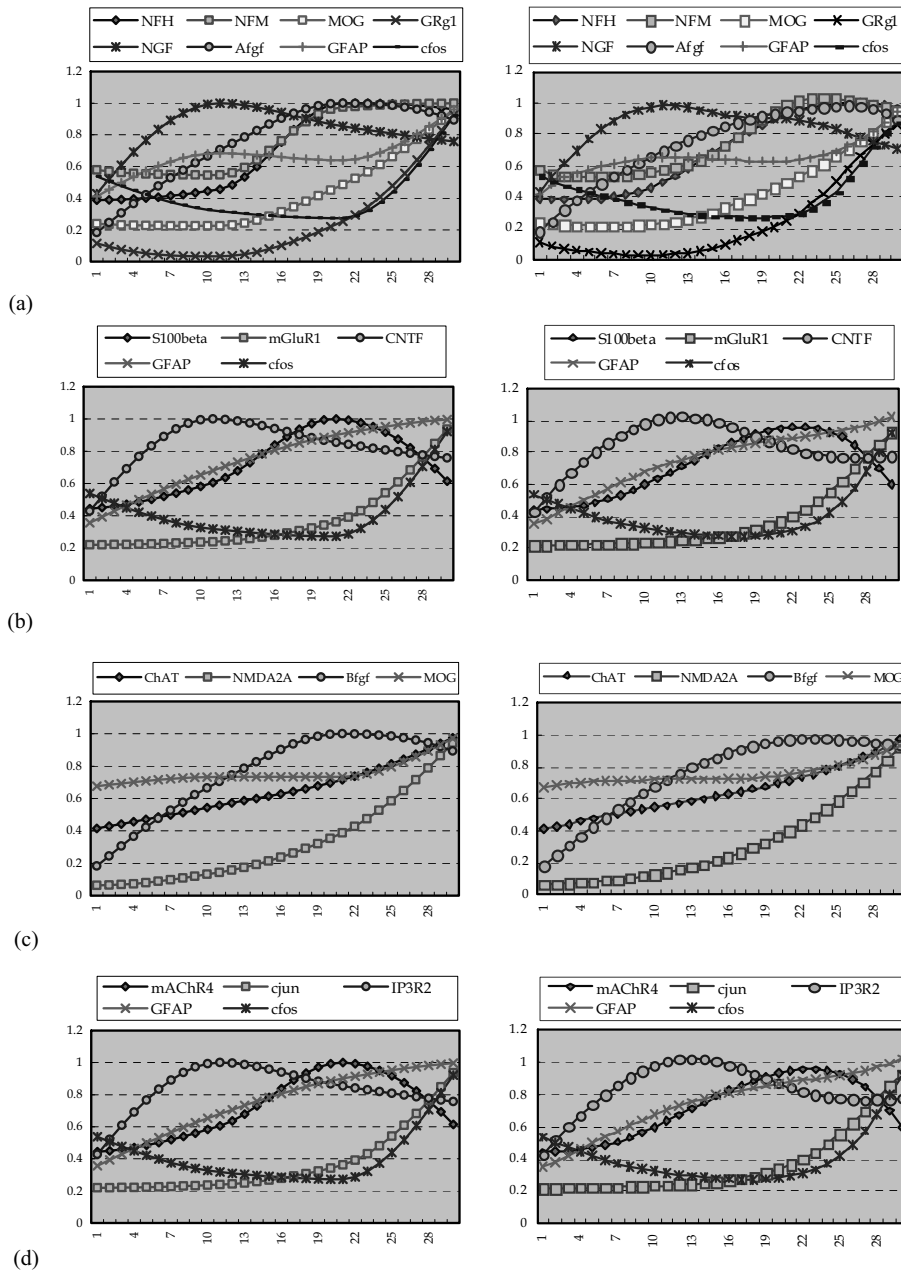
(a)

(b)

(c)

(d)

Figure 7: The behaviours of the desired and evolved networks for the four sets of genes

dynamical systems as GRNs do, our work has adopted this model to represent GRNs, and implemented an evolutionary mechanism to infer network models from expression data. In order to deal with the scalability problem, an incremental evolution strategy and a network decomposition method have been proposed to infer large networks progressively and hierarchically. To verify the presented approaches, experiments have been conducted to demonstrate how they work for the inference of GRNs. The results have shown that our approaches can be used to infer networks from measured expression data successfully.

Our work presented here shows some prospects of future research. The first is to incorporate biological knowledge into our approach to construct gene regulatory networks, in addition to minimizing mean square error for the time series gene profiles. In this way, domain knowledge can be introduced to derive more meaningful solutions. It is also worthwhile to investigate into how to employ and integrate other types of learning algorithms to furthermore improve the modeling performance. Another direction is to develop new gene clustering methods that can consider more characteristics of gene regulation at the same time in feature extraction, and are suitable for gene regulatory network modeling in particular. They should be helpful in reconstructing networks hierarchically in an even more efficient way.

References

[1] Davison, E. H., Rast, J. P., Oliveri, P., et a1.: A genomic regulatory network for development. Science 295 (2002) 1669-1678

[2] deJong, H.: Modeling and simulation of genetic regulatory systems: a literature review. Journal of Computational Biology 9 (2002) 67-103

[3] Cho, K.-H., Choo, S.-M., Jung, S. H., Kim, J.-R., Choi, H.-S., Kim, J.: Reverse engineering of gene regulatory networks. IET System Biology 1 (2007) 149-163

[4] Csete, M. E., Doyle, J. C: Reverse engineering of biological complexity. Science 295 (2002) 1664-1669

[5] Mehra, S., Hu, W., Karypis G.: A Boolean algorithm for reconstructing the structure of regulatory networks. Metabolic Engineering 6 (2004) 326-339

[6] Hartemink, A. J., Gifford, D. K., Jaakkola, T. S., Young, R. A.: Bayesian methods for elucidating genetic regulatory networks. IEEE Intelligent Systems 17 (2002) 37-43

[7] Ong, I. M., Glasner, J. D., Page, D.: Modeling regulatory pathways in *E. coli* from time series expression profiles. Bioinformatics 18 (2002) s241-s248

[8] Ho, S.-Y., Hsieh, C.-H., Yu, F.-C., Huang H.-L.: An intelligent two-stage evolutionary algorithm for dynamic pathway identification from gene expression profiles. IEEE/ACM Trans. on Computational Biology and Bioinformatics 4 (2007) 648-660

[9] Noman, N., Iba, H.: Inferring gene regulatory networks using differential evolution with local search. IEEE/ACM Trans. on Computational Biology and Bioinformatics 4 (2007) 634-647

[10] Vu, T., Vohradsky, J.: Genexp: a genetic network simulation environment, Bioinformatics 18 (2002) 1400-1401

[11] Xu, R., Venayagamoorthy, G. K., Wunsch II, D.: Modeling of gene regulatory networks with hybrid differential evolution and particle swarm optimization. Neural Networks 20 (2007) 917-927

[12] Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs (1999) Springer

[13] Yen, J., Liao, J. C., Lee, B., Randolph, D.: A hybrid approach to modeling metabolic systems using a genetic algorithm and simplex method. IEEE Trans. on Systems, Man, and Cybernetics-Part B, 28 (1998) 173-191

[14] Guan, S.-U., Chen, Q., Mo, W.: Evolving dynamic multi-objective optimization problems with objective replacement. Artificial Intelligence Review 23 (2005) 267-293

[15] Mallat, S.: A theory for multiresolution signal decomposition: the wavelet representation. IEEE Trans on Pattern Analysis and Machine Intelligence 11 (1989) 674-693

[16] Daubechies, I.: The wavelet transform, time-frequency localization and signal analysis. IEEE Trans. on Information Theory 36 (1990) 961-1005

[17] Ando, S., Sakamoto, E., Iba, H.: Evolutionary modeling and inference of gene network Information Sciences 145 (2002) 237259

[18] Tomita, M., Hashimoto, K., Takahashi, K., Shimizu, T. S., Matsuzaki, Y., Miyoshi, F., Saito, K., Tanida, S., Yugi, K., Venter, J. C., Hutchison, R.: E-Cell: software environment for whole-cell simulation. Bioinformatics 15 (1999) 72-84

[19] Wen, X., Fuhrman, S., Michaels, G., Carr, D., Smith, S., Barker, J., Somogyi, R.: Large-scale temporal gene expression mapping of central nervous system development, PNAS 95 (1998) 334-339