Modeling Large S-System using Clustering and Genetic Algorithm *

Sungwon Jung[1] Kwang H. Lee[2] Doheon Lee[2]
[1]*Department of Electrical Engineering & Computer Science, KAIST, Daejeon, Republic of Korea*
[2]*Department of BioSystems, KAIST, Daejeon, Republic of Korea*
Email: swjung@biosoft.kaist.ac.kr, khlee@biosoft.kaist.ac.kr, dhlee@biosoft.kaist.ac.kr

ABSTRACT: When we want to find out the regulatory relationships between genes from gene expression data, *dimensionality* is one of the big problem. In general, the size of search space in modeling the regulatory relationships grows in $O(n^2)$ while the number of genes is increasing. However, hopefully it can be reduced to $O(kn)$ with selected k by applying divide and conquer heuristics which depend on some assumptions about genetic network. In this paper, we approach the modeling problem in divide-and-conquer manner. We applied clustering to make the problem into small sub-problems, then hierarchical model process is applied to those small sub-problems.

1 INTRODUCTION

Recently, gene expression data is used widely to infer the functional relationships between genes. There are many approaches to use microarray gene expression data for various objectives. One of such topics is about genetic networks. Inference methods of genetic network can be distinguished in two categories according to whether the network includes dynamics information or not. For dynamic genetic network, time-series gene expression data is being used widely. However, inferencing dynamic genetic networks is difficult because there are many parameters in the dynamics model. To reduce the search space into manageable size, we use heuristic divide and conquer algorithm to model genetic networks. To show the effectiveness of this approach, we apply our idea to learn target system with S-system model.

2 PROPOSED APPROACH

To overcome the huge search space problem in fitting target genetic network model using some gene activity information, we considered following widely accepted assumptions.

- Co-regulated genes or closely regulated genes can be found to a certain extent.
- Gene regulatory network is sparse. So the genes which regulate some specific gene are just a few.
- Several 'similar' gene activity information can be regressed into one representative.

For 1st assumption, there are several methods to identify co-regulated genes including clustering. For 2nd assumption, it is generally common that the number of genes which regulate some specific gene is small compared to entire number of genes in genetic network even though a few special 'hub' genes may regulate with many genes. And for 3rd assumption, we adopt clustering and try to make representatives of clusters even though it is not still clear what is good regression method.

Based on these assumptions, we propose divide and conquer approach to reduce the search space in the process of fitting target genetic network.

2.1 Proposed divide and conquer strategy

2.1.1 Overall procedure

The objective of the divide and conquer strategy proposed here is to reduce the search space into manageable size when we fit the target genetic network model.

Proposed divide and conquer approach is composed of following steps.

- Step 1 : Build hierarchical groups of genes while each group includes closely co-regulated genes.
- Step 2 : Repeat following fitting procedure from the root to the bottom of the hierarchy.
  - Find regulatory relationships between groups on current level.
  - For each group G_i on current level, select other groups which regulate G_i more than some threshold degree while restricting the number of selected groups.
  - Decompose the groups into G_i as new element named *abstrat element.*
  - Go into every G_i which is not an abstract element and repeat step 2 by treating elements in it as groups. If G_i includes only one gene X_i, take the regulatory relationships on G_i as those of X_i.

Each steps will be discussed in following subsections.

2.1.2 Building hierarchical groups

From 1st assumption, we group each genes into several groups which include at most k predefined number of elements. This k is determined according to the distribution of connectivity degree in target network and feasibility of modeling systems

*This work was supported by National Research Laboratory Grant (2005-01450) from the Ministry of Science and Technology. We would like to thank CHUNG Moon Soul Center for BioInformation and BioElectronics and the IBM-SUR program for providing research and computing facilities.

which have at most k variables in them. When the used model is complex, too many variables in the model can make the fitting process very hard. So, k may be determined to as large value as possible while it does not make the fitting process infeasible.

If the number of result groups is more than k after grouping, those groups are re-grouped again in the same manner hierarchically until the resulting number of groups is equal or less than k. This procedure builds hierarchical tree structure of genes that each node has at most k children.

2.1.3 Finding regulatory relationships between groups

With the hierarchically grouped structure of genes from previous step, the procedure to find regulatory relationships between groups is conducted repeatedly on each hierarchy level.

Let's assume that there are m ($<= k$) groups $G_1, G_2, ..., G_m$ on some level. For these m groups, model fitting algorithm is applied to find regulatory network between those groups. The model fitting algorithm can be one of the search algorithms like genetic algorithm, but should be selected carefully because proper search method may be different according to type of data of the problem characteristics, and so on.

For two groups G_i and G_j, a regulation relation that G_j regulates G_i with degree of d will be represented as follows:

$$Regulate(G_j, G_i, d) \tag{1}$$

2.1.4 Decomposition of abstract elements

Before going into lower level of hierarchy, the regulation relations with abstract elements are decomposed first. Let's assume that some abstract element A, which is from upper level, regulates G_i with regulation degree d.

$$Regulate(A, G_i, d) \tag{2}$$

And suppose that A is composed of inner lower elements a_1, a_2, ..., a_q:

$$A = \{a_1, a_2, ..., a_q\} \tag{3}$$

The decomposition of regulation relation is defined as following. When a group of genes A, $A = \{a_1, a_2, ..., a_q\}$, regulates a gene or group G_i, the regulation relation can be decomposed into as following.

$$\begin{aligned} Regulate(A, G_i, d) \quad &= Regulate(a_1, G_i, d_{a_1}) \\ &Regulate(a_2, G_i, d_{a_2}) \quad ..., \\ &\quad Regulate(a_m, T, d_{a_m}) \end{aligned} \tag{4}$$

The notation means proper regression of regulation. By decomposition process described in this section, the upper level regulations are split into lower level regulations. This makes the final solution be constructed with not abstracted groups but actual genes finally.

2.1.5 Inserting abstract elements

After model fitting and decomposition process on current level, other groups which regulate G_i are determined for each group G_i. These regulating groups are every $G_j (j \neq i)$ which regulate G_i more than some regulation degree threshold θ. Let's assume that G_i includes p elements which are subgroups of G_i:

$$G_i = \{g_1, g_2, ..., g_p\},\ (p \leq k) \tag{5}$$

If the number of groups which regulate G_i more than threshold θ is t, those t groups are also able to be in G_i because it is determined that they regulate G_i, so the elements of G_i. Now, those t groups are inserted into G_i as *abstract elements*. Finally, the G_i will be like following after inserting abstract elements:

$$G_i = \{g_1, g_2, ..., g_p, a_1, a_2, ..., a_t\} \tag{6}$$

a_j are abstract elements.

2.2 Search space analysis

Let's assume that there are total n genes in target genetic network which is unknown yet. Generally, the number of parameters of genetic network model is $O(n^2)$ without any knowledge on genetic network. Our objective is to reduce that order.

Let k be the maximum number of elements in a group when the proposed divide and conquer approach is applied to find target genetic network. By hierarchical grouping procedure, the hierarchy tree of groups of genes is constructed while each node has at most k elements(groups or genes). We assume that the number of abstract elements in a group is small constant, C. So the number of parameters in the process of finding genetic network model in one group is:

$$O((k+C)^2) = O(k^2) \tag{7}$$

Decomposition procedure finds the regulation degree of each regulations of lower level elements in some group(which is an abstract element). It is to find every d_{a_i} in right hand side of the equation (4). We can assume that the number of parameters in decomposing one group which is an abstract element is

$$O(k) \tag{8}$$

because one group may have at most k elements in it.

In worst case, decomposition process of one abstract element can occur at most k times because there are maximum k non-abstract elements in the group. So, the total number of parameters of decomposition for one abstract element is

$$kO(k) = O(k^2) \tag{9}$$

Because there are constant C abstract elements in a group, the total number of parameters in decomposing every abstract element in one group is:

$$C \cdot O(k^2) = O(k^2) \tag{10}$$

By (7) and (10), the total number of parameters in one group is

$$O(k^2)+O(k^2)=O(k^2) \tag{11}$$

When we build hierarchical tree of groups from n genes by limiting the maximum number of elements to k, the total number of nodes(groups) is

$$1+k+k^2+...+k^{(log_k n-1)}=\frac{k^{log_k n}-1}{k-1}=\frac{n-1}{k-1} \tag{12}$$

Finally, the total number of parameters to be fitted is determined by multiplication of (11) and (12).

$$\frac{n-1}{k-1}\cdot O(k^2)=O(kn) \tag{13}$$

So, the total number of parameters can be reduced to $O(kn)$. If $k \ll n$, it goes to $O(n)$.

3 S-SYSTEM MODELING WITH PROPOSED APPROACH

3.1 S-system model

There are n genes X_1, X_2, ..., X_n and time-course expression data D_1, D_2, ..., D_n accordingly. The simplified form of S-system model describes the dynamics of some variable X_i as follows:

$$\frac{dX_i}{dt}=\alpha_i\prod_{j=1}^{n}X_j^{g_{ij}}-\beta_i\prod_{j=1}^{n}X_j^{h_{ij}} \tag{14}$$

S-system model has the form which is easy to understand and intuitive. However, the major disadvantage of S-system model is its large number of parameters to be estimated. The number of parameters is $2n(n+1)$, where n is the number of state variables. Several approaches to use S-system model to describe the dynamics of biosystems have been proposed. A technique for the dynamic modeling of complex biosystems by combining genetic algorithm and the S-system has been proposed [1] [2]. However, estimation of the S-system parameters is too difficult with the conventional simple GA. There were another improvement in using GA to predict the parameters of S-system [3]. They improved an evaluation function of GA that aims at eliminating futile parameters by adding the sum of the absolute values of model parameters to the conventional error function. With this and additional ideas, they succeeded to increase the predictable number of parameters using GA and showed its performance by fitting 5 variable S-system example. However, the use of GA on S-system is still very limited to the case of such a few variables.

When the number of variables in a S-system is increased more than about 5, it is very difficult to find even sub-optimal solutions. This is because of the huge increase of entire search space and the sensitivity of large size S-system.

3.2 Proposed method for S-system with GA

In this section, divide and conquer algorithm using clustering and GA to estimate S-system parameters is proposed. Basically, this algorithm follows the procedure which is mentioned earlier. In addition to the basic algorithm structure, it needs following key methods for learning S-system with clustering and GA.

- Hierarchical group structure building algorithm
- Making representative time-course for a group
- Parameter search algorithm for S-system
- Parameter search algorithm in the decomposition process.

For hierarchical group structure building, hierarchical clustering of time-course gene expression has been used. For two parameter search procedures, GA has been used.

The specific steps is as follows:

1. Hierarchical clustering of gene expression data.
2. Making restructured cluster hierarchy such that each node has at most k children.
3. From root, using GA to estimate S-system between clusters. In each cluster, decompose abstract elements.
4. Putting at most $k-1$ regulating element into the regulated cluster as abstract elements.
5. Repeating step 3 and 4 until leaf node is encountered.

Hierarchical clustering and hierarchy restructuring is done easily by using normal hierarchical clustering and cutting proper levels to reset the size of clusters. We will describe the GA to learn S-system between clusters and to decompose abstract elements in following subsections.

3.2.1 GA for learning S-system between clusters

For learning a S-system between clusters, we use averaged gene expression data as a representative for each cluster. With those representative data, we apply following evaluation function in the process of GA [3].

$$E=\sum_{i=1}^{n}\sum_{t=1}^{T}\left(\frac{X_i'(t)-X_i(t)}{X_i(t)}\right)^2+cnT\left\{\sum_{i,j}|g_{ij}|+\sum_{i,j,i\neq j}|h_{ij}|\right\} \tag{15}$$

$$Fitness=\frac{1}{E} \tag{16}$$

n is the number of variable in the S-system. T is the number of sampling points of the time-course data. $X_i'(t)$ is the numerically calculated time-course at time t of a state variable X_i, and $X_i(t)$ represents the experimentally observed time-course at time t of X_i.

3.2.2 GA for decomposition of abstract elements

In the decomposition process, the regulation relation with an abstract element should be decomposed into relations of children of abstract element. Let's assume that target gene X_i is regulated by some abstract element A_j. The regulation relation of X_i is expressed as follows:

$$\frac{dX_i}{dt} = \alpha_i X_1^{g_{i1}} \times \ldots \times A_j^{g_{ij}} - \beta_i X_1^{h_{i1}} \times \ldots \times A_j^{h_{ij}} \qquad (17)$$

When $A_j = \{a_1, a_2, \ldots, a_p\}$, the decomposition procedure decomposes $A_j^{g_{ij}}$ into following form to keep the form of S-system description:

$$A_j^{g_{ij}} = a_1^{g_1} \times a_2^{g_2} \times \ldots \times a_p^{g_p} \qquad (18)$$

Equation (18) requires that the multiplication of time-course values of decomposed elements should meet the time-course value of $A_j^{g_{ij}}$. To estimate those parameters, GA is used here. The evaluation function to fit those exponents in the decomposed result is as follows:

$$E = \sum_{t=1}^{T} \left(A_j^{g_{ij}}(t) - \prod_{l=1}^{p} a_l^{g_l}(t) \right)^2 \qquad (19)$$

$$Fitness = \frac{1}{E} \qquad (20)$$

3.2.3 Selecting abstract elements

To go into the lower level of a cluster, other clusters those regulate the cluster are inserted as abstract elements. To reduce the increase of search space, only at most $k-1$ elements are inserted into the regulated cluster. The degree of regulation d_{ij} of X_j onto X_i is defined as follows:

$$d_{ij} = |g_{ij}| + |h_{ij}| \qquad (21)$$

With this degree of regulation, at most $k-1$ elements are selected as abstract elements those have regulation degree larger than some threshold θ.

4 EXPERIMENT

4.1 Experimental environment

An S-system which has 10 variables is used as target system to show the ability of proposed approach. When we apply conventional GA, we couldn't get the result when there are more than 8 variables. We used following artificial S-system with 10 variables as a target system. This system is made by hand.

We sampled 10 sets of time-course from the system in Table 1 with initial values described in Table 2.

An example of used time-course data is shown in Figure 1.

The conditions of our experiment were as follows: number of sampling point for evaluation = 10, number of choromosomes in population for every GA procedure = 65, max number of generation = 35000, α_i and $\beta_i \in$ [0, 15], g_{ij} and $h_{ij} \in$ [-3, 3], weighting parameter $c = 0.15$, $k = 5$.

$$\begin{aligned}
\frac{dX_1}{dt} &= 0.12X_9^{1.4} - 2.5X_2^{2.84}\\
\frac{dX_2}{dt} &= 0.35X_2^{3} - 4.64X_7^{0.62}\\
\frac{dX_3}{dt} &= 0.47X_2^{2.7} - 8.44X_3^{2.87}\\
\frac{dX_4}{dt} &= 2.96X_6^{-0.44} - 14.99X_4^{0.62}\\
\frac{dX_5}{dt} &= 1.36X_6^{3} - 15X_2^{1.2}X_5^{1.12}\\
\frac{dX_6}{dt} &= 3.63X_5^{3} - 10.4X_6^{0.98}X_7\\
\frac{dX_7}{dt} &= 3.23X_2^{2.01}X_4^{3}X_6^{2.46}X_7^{1.29}X_9^{1.95} - 14.51X_7^{2.26}\\
\frac{dX_8}{dt} &= 0.04X_2^{-2.5} - 13.25X_2^{1.31}X_8^{0.94}\\
\frac{dX_9}{dt} &= 0.44X_5^{0.65} - 13.67X_9^{1.7}\\
\frac{dX_{10}}{dt} &= 0.23X_2^{-2.71}X_9^{1.77} - 10.72X_{10}^{1.64}
\end{aligned}$$

Table 1: 10 variable artificial target S-system

Trial	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
1	0.24	0.93	0.11	0.88	0.32	1.29	0.4	0.13	0.76	0.1
2	1.41	1.33	0.97	0.8	0.38	1.13	0.45	1.02	0.52	0.73
3	0.4	1.06	1.43	0.56	0.2	0.94	0.58	0.62	0.51	0.22
4	1.22	0.74	0.44	0.6	0.36	0.97	0.11	0.46	1.38	1.3
5	1.17	0.32	0.83	0.05	0.86	0.85	0.01	0.83	0.24	0.95
6	1.47	1.08	0.51	1.33	1.21	1.17	0.22	0.98	1.34	0.62
7	0.62	1.25	0.37	0.04	0.16	1.27	0.87	1	1.02	1.18
8	1.28	1.35	0.63	1.01	0.81	0.23	1.19	1.28	0.39	0.3
9	1.09	1.27	0.16	1.08	0.45	0.95	0.75	0.34	0.46	0.06
10	1.39	0.9	0.65	0.22	0.82	1.08	0.21	0.36	0.39	0.21

Table 2: 10 sets of initial concentrations used in our computational experiments. These values were also prepared artificially

4.2 Result

The found S-system by proposed method is shown in Table 3. The example of time-course from the found S-system with initial values of trial 6 is shown in Figure 2.

$$\begin{aligned}
\frac{dX_1'}{dt} &= 0.52X_1^{1.8}X_2^{1.34} - 3.63X_2^{2.05}\\
\frac{dX_2'}{dt} &= 3.04X_2^{0.27} - 4.6X_2^{0.26}\\
\frac{dX_3'}{dt} &= 0.28X_2^{3} - 8.19X_3^{2.98}\\
\frac{dX_4'}{dt} &= 1.87X_6^{-0.5} - 14.67X_4^{0.77}\\
\frac{dX_5'}{dt} &= -12.79X_2^{1.33}X_5^{1.24}\\
\frac{dX_6'}{dt} &= 0.11X_1^{3}X_8^{2.69} - 9.93X_6^{0.99}X_7^{1.12}\\
\frac{dX_7'}{dt} &= 15X_5^{3}X_6^{3} - 13.81X_7^{2.24}\\
\frac{dX_8'}{dt} &= 0.02X_1^{1.48}X_2^{-3}X_5^{-2.02}X_6^{-2.83}X_8^{3} - 15X_2^{2.06}X_8^{1.11}\\
\frac{dX_9'}{dt} &= 0.16X_5^{0.62}X_9^{-0.4} - 13.01X_9^{1.76}\\
\frac{dX_{10}'}{dt} &= 0.02X_2^{-3} - 10.16X_{10}^{1.69}
\end{aligned}$$

Table 3: S-system found by proposed algorithm.

Among total 28 non-zero exponents in the target S-system, 19 non-zero exponents are also in the found S-system. This is about 68% correctness of finding non-zero parameters. For the true-positive ratio, 19 of 30 non-zero exponents in the found S-system are correct non-zero exponents. So, it showed about 63% true positives. When we count the overall exponent parameters including 0 exponents, it can be said that 180 among 200 exponents are determined correctly whether they are zero or non-zero exponents. When we consider the fact that it is not easy to find even a candidate solution for large size S-system, we think that this method may provide some candidate solutions for such complex systems.

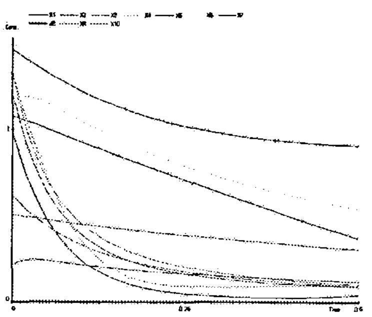

Figure 1: Time-course of trial 6

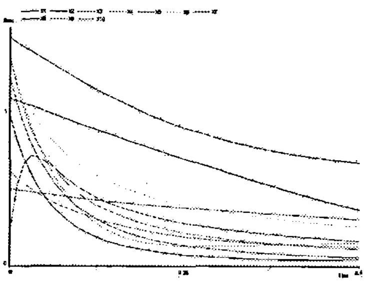

Figure 2: Time-course from learned S-system with initial values of trial 6

5 CONCLUDING REMARKS

In this paper, we proposed a divide and conquer approach to reduce the search space in fitting systems which describe genetic networks. The proposed approach makes hierarchical groups of genes and find local regulatory relationships in each group. With the concept of *abstract element* and *decomposition*, this approach tries to find the connectivity of relation between elements in different groups. The proposed approach can get a candidate system model which describes complex dynamic system. When this approach is used with selected k, the number of parameters to be found is reduced from $O(n^2)$ to $O(kn)$. We evaluated the performance of the proposed method with a artificial S-system of 10 variables.

Based on the current work, we would like to study the following issues as a further work. We just applied proposed approach to find 10 variables target S-system and need further evaluation to show the effectiveness of this approach for larger scale systems. In fact, the S-system model is not suitable for larger scale system modeling because of its high nonlinearity and high sensitivity. So some further study on larger genetic network model and application of proposed approach to those cases may be needed.

REFERENCES

[1] Tominaga, D. and Okamoto, M.. Design of caninical model complex nonlinear dynamics. In Proceedings of the International Conference on Computer Applications in Biotechnology, pages 85–90, 1998.

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

[3] Shinichi Kikuchi, Daisuke Tominaga, Masanori Arita, Katsutoshi Takahashi and Masaru Tomita. Dynamic modeling of genetic networks using genetic algorithm and S-system. Bioinformatics, 19(5):643–650, 2003.

[4] S. Dutta. An event-based fuzzy temporal logic. In Proc. 18th IEEE Intl. Symp. on Multiple-Valued Logic, pages 64–71, Palma de Mallorca, Spain, 1988.

[5] M. G. Harbour, M. H. Klein, and J. P. Lehoczky. Timing analysis for priority scheduling of hard real-time systems. IEEE Transactions of Software Engineering, 20(1):13–28, 1994.

[6] Esko Turunen. Mathmatics Behind Fuzzy Logic. Advances in Soft Computing. Verlag, Heidelberg, 1999.