Available online at www.sciencedirect.com
ScienceDirect
Journal of Bionic Engineering 8 (2011) 98–106

Combination of Neuro-Fuzzy Network Models with Biological Knowledge for Reconstructing Gene Regulatory Networks

Guixia Liu[1], Lei Liu[1], Chunyu Liu[2], Ming Zheng[1], Lanying Su[1], Chunguang Zhou[1]

1. *College of Computer Science and Technology*, *Jilin University*, *Changchun* 130012, *P. R. China*
2. *Affiliated Hospital of Changchun University of Chinese Medicine*, *Changchun* 130021, *P. R. China*

Abstract

Inferring gene regulatory networks from large-scale expression data is an important topic in both cellular systems and computational biology. The inference of regulators might be the core factor for understanding actual regulatory conditions in gene regulatory networks, especially when strong regulators do work significantly. In this paper, we propose a novel approach based on combining neuro-fuzzy network models with biological knowledge to infer strong regulators and interrelated fuzzy rules. The hybrid neuro-fuzzy architecture can not only infer the fuzzy rules, which are suitable for describing the regulatory conditions in regulatory networks, but also explain the meaning of nodes and weight value in the neural network. It can get useful rules automatically without factitious judgments. At the same time, it does not add recursive layers to the model, and the model can also strengthen the relationships among genes and reduce calculation. We use the proposed approach to reconstruct a partial gene regulatory network of yeast. The results show that this approach can work effectively.

Keywords: neuro-fuzzy network, biological knowledge, regulators, gene regulatory networks

doi: 10.1016/S1672-6529(11)60008-5

1 Introduction

One of the hottest topics in genome science is the interaction between genes. The study of Gene Regulatory Network (GRN) is focused to understand metabolic pathways and bioprocesses. As a system, a GRN comprises of biomolecular components (genes, mRNA and proteins) which interact with each other. These interactions determine gene expression levels, that is, determine the rate of gene transcription to mRNA. In general, each mRNA molecule can be translated into a specific protein (or set of proteins). On one hand, some proteins serve only to activate other genes in nuclei, which are thought of the transcription factors that are the main players in regulatory networks. Transcription factors which transcribe genes into mRNAs, can be considered as input signals. When transcription factors bind to promoter regions adjacent to the regulated gene, they recruit RNA polymerase to perform transcription function. On the other hand, proteins that are translated from the mRNAs, can be considered as output signals. Some proteins act as transcription factors themselves to upregulate or downregulate gene expressions. These courses form feedback loops in the network, in which direct or indirect self-regulation happens[1–3]. With the rapid development in recent years, microarray data has become an important resource for bioinformatics research increasingly. Based on time series expression data obtained from DNA microarrays regulatory networks, we can identify the complicated regulatory relationships, uncover the regulatory patterns in the cell, and obtain a systematic view for biological process. GRN models can be used to identify genetic diseases and to estimate the effects of medications. One of the most challenging tasks is to reconstruct interactional structures and to confirm mechanisms in cellular systems from available experimental data. Due to the lack of the experimental data and prior knowledge, it is hard to verify regulatory relations, which are required to construct the regulatory network in traditional methods. Therefore, there is still a lot of work to do on gene networks construction[4,5].

Many computational approaches have been proposed to reconstruct GRNs based on large-scale

Corresponding author: Lei Liu
E-mail: jackie0421@yahoo.cn

microarray data which is retrieved from such biological experiments as Boolean networks[6], differential equations[7], linear combination and weighted model[8], Bayesian networks[9] and neural networks[10]. But all the existing regulatory network models have some inevitable drawbacks. For example, Boolean network models are used simply to determine the discrete model, but they are relatively rough, fixed, and have low accurate results. The differential equation model is a continuous network model, and it describes the relationships of gene impacts and changes. Although this model can reflect the genetic continuous dynamic relations better, it is difficult to establish differential equations in the right forms. It is thought that the linear combination and weighted model in the respect of establishing the relationship among genes is linear, but in fact, the relationship among genes is often very complex and nonlinear. While Bayesian network models are attractive due to dealing with stochastic aspects of gene expression and noisy measurements, but they also have the disadvantage of minimizing dynamic gene regulation. Recently, there have been some attempts to apply neural network models to investigate gene networks for better results.

In this paper, we apply a neuro-fuzzy network model and optimize weight values with genetic algorithm. By this approach we obtain strong regulators to regulate known genes, which include 7 genes whose regulators are indefinite, and get fuzzy rules during calculation. Finally we reconstruct a partial gene regulatory network using the obtained strong regulators and fuzzy rules, and satisfactory results are obtained from experiments.

2 Method

2.1 Model

It is very popular that neural network is used for inferring. For example, dynamic neuro-fuzzy network model, whose network structure and rules are constructed during on-line learning, is flexible[15]. But the determined rules are not necessarily accurate. Recurrent neuro-fuzzy network model is better in dealing with the series data than other neuro-fuzzy networks, but it is more complex in calculation[16].

The proposed model is a six-layer, two-input and single-output neuro-fuzzy network, which has the advantages of both neural networks and fuzzy systems. As a hybrid neuro-fuzzy architecture, it can not only infer the fuzzy rules, which are suitable for describing the regulatory conditions in regulatory networks, but also explain the meaning of nodes and weight value in the neural network. It can get useful rules automatically without factitious judgments[17,18]. At the same time, no recursive layers are added to the model, but the model can still strengthen the relationships among genes and reduce calculation. The proposed neuro-fuzzy network model is shown in Fig.1.

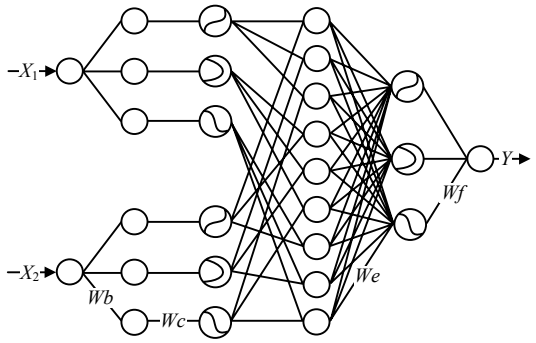


Fig. 1 The neuro-fuzzy network model.

We use O_i^j and I_i^j to respectively represent the output and input of the *i*-th node in the *j*-th layer.

In the first layer, each node represents an input linguistic variable (a linguistic input variable). There is no calculation taking place in this layer, and each node transfers the value of an input variable to the next layer.

$$O_i^1 = I_i^1 = X_i \ (i = 0,1), \tag{1}$$

where X_i represents gene.

We make input linguistic variable turn into fuzzy output in the second and third layers. Each node in the second layer needs to do a conversion as

$$O_i^2 = I_i^2 - Wb_i = X_i - Wb_i \ (i = 0,\ 1,\ \ldots\ ,5). \tag{2}$$

where Wb_i is the weight from I_i to O_i in the second layer.

In the third layer, input and output variables define three fuzzy sets (*big*, *mid*, *small*), which present large, medium and small respectively. The membership functions are represented as

$$\begin{aligned} big(x) &= \mathrm{sigmoid}(\alpha_1(x-\beta_1)) \\ &= 1/\left(1+\exp(-\alpha_1(x-\beta_1))\right), \end{aligned} \tag{3}$$

$$mid(x) = \exp(-(x-\beta_2)/\alpha_2), \tag{4}$$

$$\begin{aligned} small(x) &= 1-\mathrm{sigmoid}(\alpha_3(x-\beta_3)) \\ &= 1-1/\left(1+\exp(-\alpha_3(x-\beta_3))\right), \end{aligned} \tag{5}$$

where α_i and β_i are factors in fuzzy membership functions. At the same time the output of the third layer is shown as

$$\begin{aligned} O_i^3 &= \text{sigmoid}(Wc_i \times I_i^3) \\ &= \exp(-I_i^j / Wc_i) \\ &= 1 - \text{sigmoid}(Wc_i \times I_i^j)\ (i = 0,1,...,5), \end{aligned} \quad (6)$$

where Wc_i is the weight from I_i to O_i in the third layer.

Fuzzy rules are inferred in the fourth and fifth layers. The output of each node in the fourth layer is as follows

$$O_i^4 = \prod I_i^4,\ i = 0,1,... ,8, \quad (7)$$

The conversion formula for the fifth layer is

$$O_i^5 = \text{sigmoid}(\sum (I_i^5 \times We_{ij}))\ (i = 0,1,2), \quad (8)$$

where We_{ij} is the weight from I_i to O_i in the fifth layer.

The sixth layer is responsible for the defuzzification operation. It implements sum function by using the cancroids non-fuzzy method. The output of each node in sixth layer is shown below

$$O_i^6 = \sum (Wf_i \times I_i^6) / \sum I_i^6, \quad (9)$$

where Wf_i is the weight from I_i to O_i in the sixth layer.

2.2 Algorithm implementation

In the algorithm implementation, we use genetic algorithm to train network. Chromosomes are made up of weights Wb_i, Wc_i, We_{ij} and Wf_i, the fitness $S(i)$ of the i-th chromosome in group is

$$S(i) = 1/E(i), \quad (10)$$

$$E(i) = \frac{1}{2}\sum_k (O - T)^2, \quad (11)$$

where k is the number of samples, T is teacher signal, and O is the actual output of network to which this chromosome corresponds. The chromosome number of initial group is selected as 200, and the chromosomes with higher fitness are selected as fathers and mothers. We reserve the top 10 fitness chromosomes in each generation, and there are 10 chromosomes mutating in each generation. Then we select 90 pairs of father chromosomes to carry out cross operation. The network error is normally between 1 and 2 when the genetic operation arrives at the 20th generation. At this time, the network is easily to fall into local minimum value, and the error does not change any more. Our approach is to modify gene fragments, and meanwhile modify the value at 10 places of the chromosome randomly. Then we check the network error of the corresponding chromosome to see whether it decreases or not. If it decreases, we preserve the modified chromosome; otherwise the experiment goes on. Each chromosome is experimented 100 times, so we can make sure that the network error can jump out the local minimum. The network error will decline below 1 after the algorithm runs 20 generations. Training error curve is shown in Fig. 2.

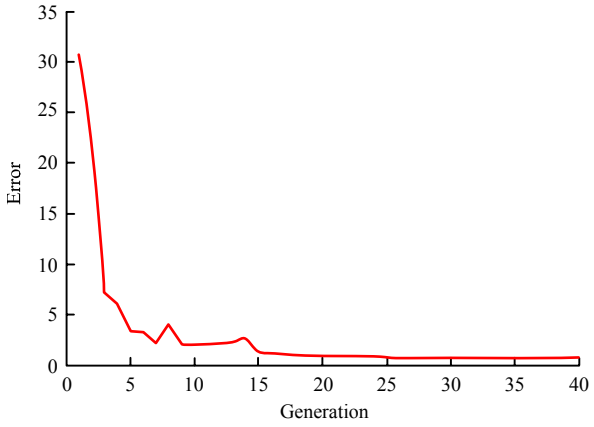


Fig. 2 Error curve of neural network.

2.3 Extracting fuzzy rules

We record any change of weight We_{ij} in the process of implementing algorithm. The growing of We_{ij} is extracted with the decrease in the network error. These obtained weights contribute to the convergence of network, so the corresponding fuzzy rules are needed. There are multiple obtained fuzzy rules, for example, we get two rules, Rule 1: If regulator Y is big, then gene X_1 is big and gene X_2 is mid, it corresponds to a weight marked as W_1; Rule 2: If regulator Y is big, then gene X_1 is mid and gene X_2 is small, and it corresponds to a weight marked as W_2. If the increased value of W_1 is greater than that of W_2, we will keep Rule 1 and give up Rule 2.

3 Experimental results

3.1 Profiles introduction

We applied yeast (*Saccharomyces cerevisiae*) cell-cycle gene expression data[19] (download from http://genome-www.stanford.edu/cellcycle/) reported by Spellman *et al.*[20], which is made up of 77 samples collected at different time points of the cell-cycle. The

data come from six different experiments: alpha-factor, CDC15 arrest, CDC28 arrest, elutriation, CLN3 and CLB2. Missing values are filled with zero. After normalization, the anterior 60 samples were used as training data of the neuro-fuzzy network, and the fuzzy rules inferred were tested for the consistency of the other 17 samples.

The biological knowledge was integrated in our method. The knowledge includes:

(1) There are 104 genes which are regulated in cell-cycle of the yeast and they were substantiated by scientists through distinct ways. However, only 97 genes were analyzed by using the methods of the researchers in Stanford University, and the evidence of the other 7 genes has not been found yet.

(2) This cell-cycle includes five periods: M/G1, G1, S, S/G2 and G2/M. The genes in each period have been clustered by Spellman *et al*., and their paper also showed us the functions of genes[20].

(3) There are 149 regulators for 97 genes in the (Saccharomyces Genome Database) SGD database[21], and each gene usually has 6 regulators.

3.2 Results

(1) The calculating results of the strong regulators of 97 genes

There are two inputs X_1 and X_2, and single output Y in our neuro-fuzzy network model. This model describes the intensity of relations among these 3 genes. If the inputs X_1, X_2 and output Y are in the same period and have similar functions, or they have regulation relationships, the error gained from training samples will be the minimum and the expected curve is more similar with real curve in test. In our experiments, the input genes X_1, X_2 were selected from the same period and they have same or similar functions, and Y is selected from the regulator sets of this period. We got $Y_{\min}$ whose calculated network error was the least (minimum) for each pair of X_1, X_2, which is the strong regulator of X_1 and X_2. If $Y_{\min}$ belong to the regulator subsets of X_1 and X_2, then we consider the result is right, otherwise, it is wrong.

As mentioned above 104 genes and 149 regulators are divided into 5 parts according to the cell-cycle periods, as shown in Table 1. There are only 97 genes whose regulator sets are recorded in SGD database. Our first experiment is to calculate the strong regulators of these genes, and results show that only 8 genes do not belong to their own regulator sets, the accuracy is 91.75%.

Table 1 The number of genes and regulators in each period

Period	Number of genes	Number of regulators
M/G1	19	46
G1	53	49
S	8	5
S/G2	9	27
G2/M	15	22
Sum	104	149

(2) The calculating results of the strong regulators of 7 genes which do not have regulator records in the SGD database

These 7 genes are: CSD2/CHS3, CDC8, DPB3, PRI1, RAD17, CWP2 and TIR1. Among them, CSD2/CHS3, CDC8, DPB3, PRI1 and RAD17 are in G1 phase, CWP2 and TIR1 are their strong regulators. CWP2 and TIR1 were calculated by the neuro-fuzzy network model from their regulator sets which consists of 53 regulators. CWP2 and TIR1 are in S/G2 phase, HCM1 is their strong regulator which was calculated by the neuro-fuzzy network model from its regulator set which is made up of 27 regulators. The result needs to be further validated by biological experiments.

(3) Using the strong regulators of 104 genes and fuzzy rules to reconstruct a partial gene regulatory network of yeast

Each gene we researched has about 6 regulators in the SGD database. Our method can compute regulators' degree of each gene and extract the strong regulator whose regulatory phenomenon is most obvious.

We can use the fuzzy rules to describe conditions of regulation. For example, there is a rule: if the expression of regulator Y is big, then the expression of gene X_1 is big and that of gene X_2 is small, which means that the expression of gene X_1 is induced and that of gene X_2 is repressed when regulator Y is highly expressed. But sometimes the expression of gene X_1 or X_2 is mid, which means that the expression of regulated gene is in the middle express level, which is not obviously increased or decreased. Simple regulation is shown in Fig. 3.

The results of regulation and the regulatory modules in Saccharomyces Genome Database[22] are compared with each other. There are 70 regulation sides in common, and 59 sides are consistent with the records in the database above. The accuracy is 84.28%. The regulatory network we have reconstructed is shown in Fig. 4.

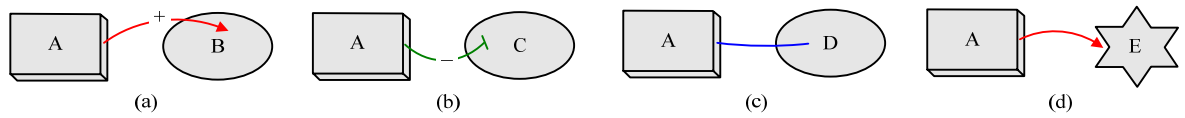
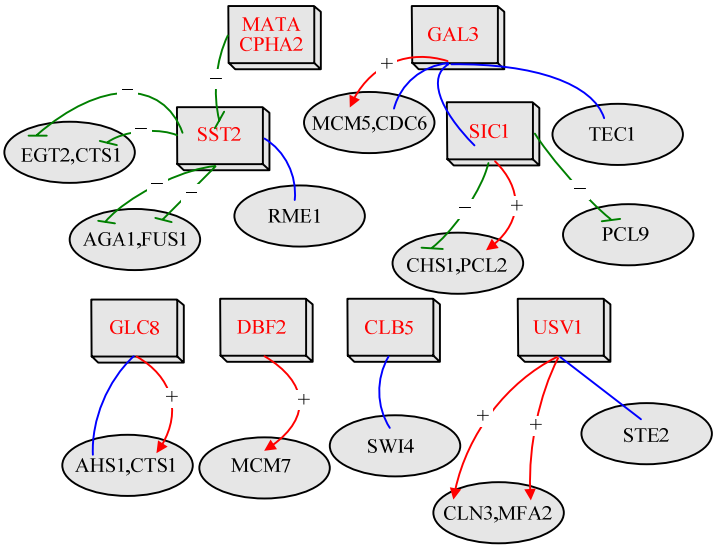
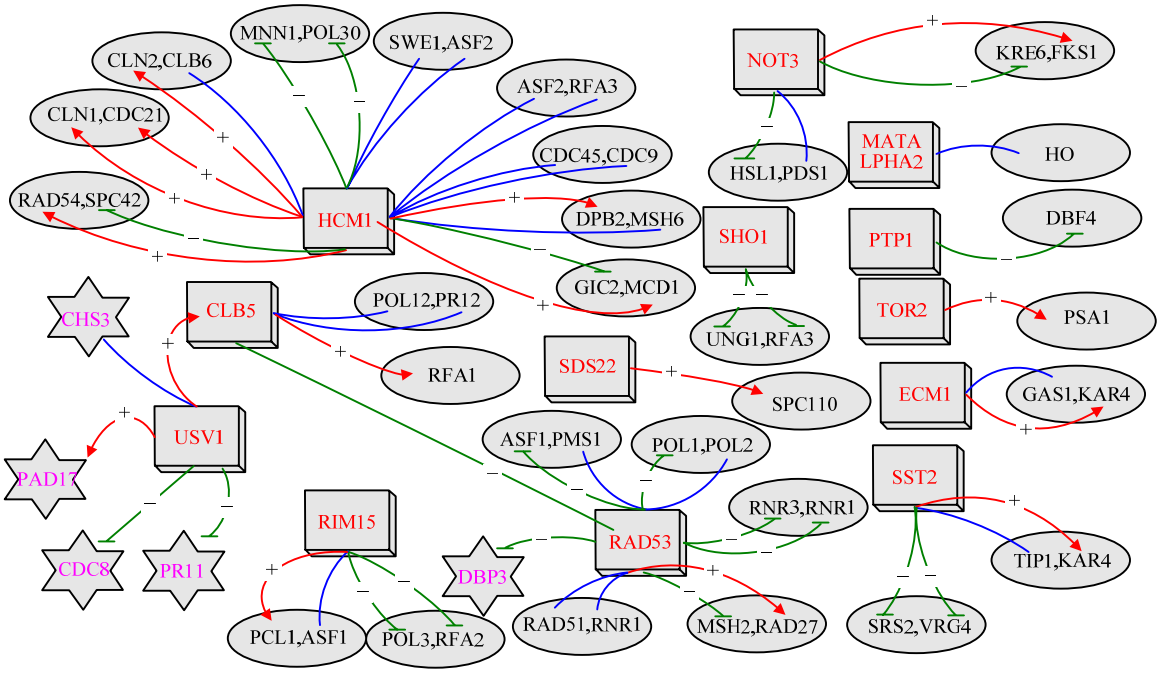


Fig. 3 Introduce the meaning of four kinds of relations. (a) Gene A makes the expression of gene B increase in biology process with "+"; (b) gene A makes the expression of gene C decrease with "−"; (c) gene D is in medium level and the regulation relationship between gene A and gene D is not obvious; (d) we do not know the regulators of gene E, which is in the form of hexagon.

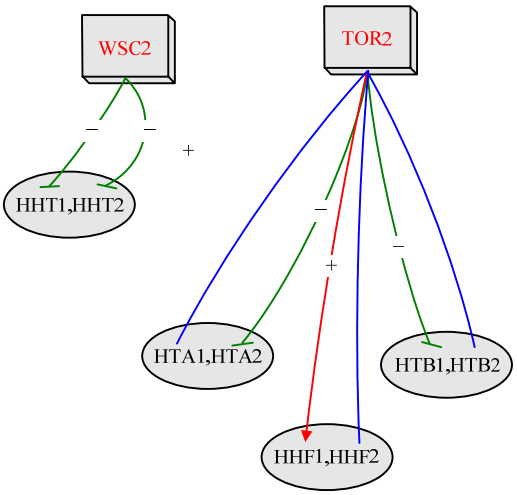


(a) M/G1 period

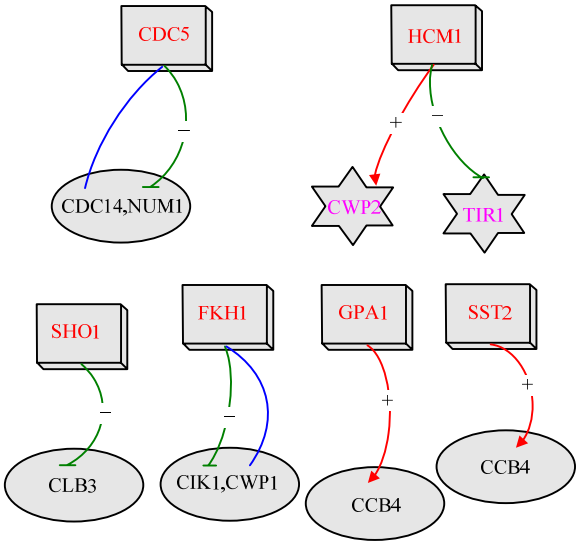


(b) G1 period

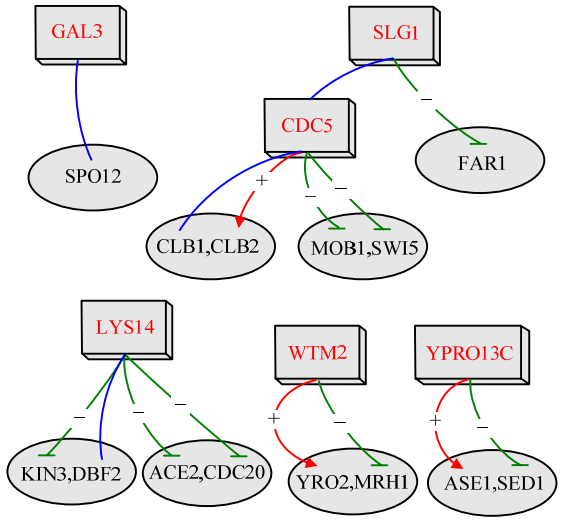
Fig. 4 Reconstructing a gene regulatory network of 104 genes based on the combination of neuro-fuzzy network model and biological knowledge: It shows the express situation of regulated genes when expressions of strong regulators increase. The entire network consists of networks of 5 periods during cell cycle of yeast. (a) M/G1 period: there are 19 genes and 8 strong regulators. (b) G1 period: there are 53 genes, including 5 purple genes whose regulators are unknown and 13 strong regulators. (c) S period, there are 8 genes and 2 strong regulators. (d) S/G2 period, there are 9 genes, including 2 purple genes whose regulators are unknown and 6 strong regulators. (e) G2/M period, there are 15 genes and 6 strong regulators. The genes in pane are regulators, the genes in ellipse are 97 regulated genes, while 7 genes in hexagon are the genes without regulators in record.



(c) S period



(d) S/G2 period



(e) G2/M period

Fig. 4 Continued.

4 Discussion

When calculating the relation among X_1, X_2 and Y with neuro-fuzzy network model, we find that there are four relations between X_1, X_2. Relation 1: X_1 and X_2 are in the same period and have similar functions, and Y is their regulator. Relation 2: X_1 and X_2 do not have similar functions and Y is the regulator of X_1 or X_2. Relation 3: X_1 and X_2 are in the same period and have similar functions, Y is not the regulator of them. Relation 4: X_1, X_2 and Y are in the same period and they also have same or similar functions.

Comparing the above relations among input genes X_1, X_2 and the output gene Y, we can find that the error gained from training is minimum and the expected curve is more similar to real curve than other relations when X_1, X_2 and Y obey Relation 1. The curves of the four kinds of relations are shown in Figs. 5 to 8.

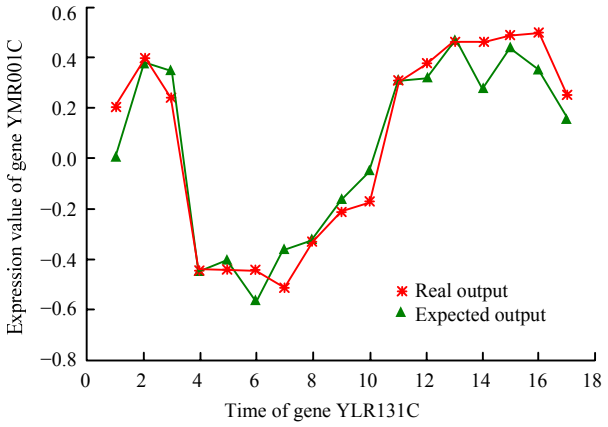


Fig. 5 Test curves of Relation 1. Total error is 0.7232, the average error of a single data is 0.012.

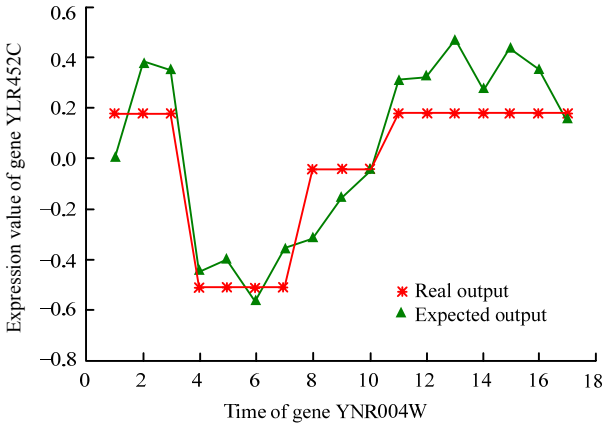


Fig. 6 Test curves of Relation 2. Total error is 0.4893, the average error of a single data is 0.008.

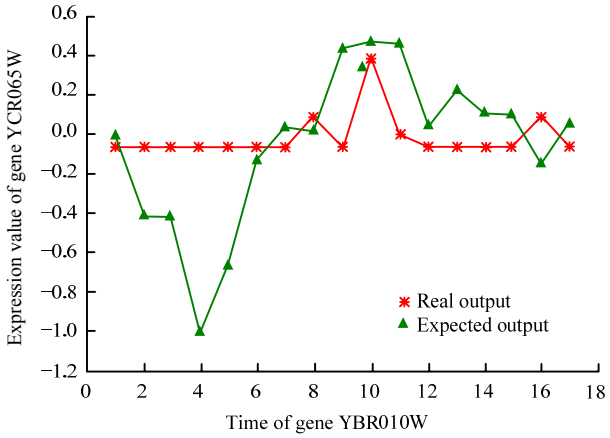


Fig. 7 Test curves of Relation 3. Total error is 2.4893, the average error of a single data is 0.043.

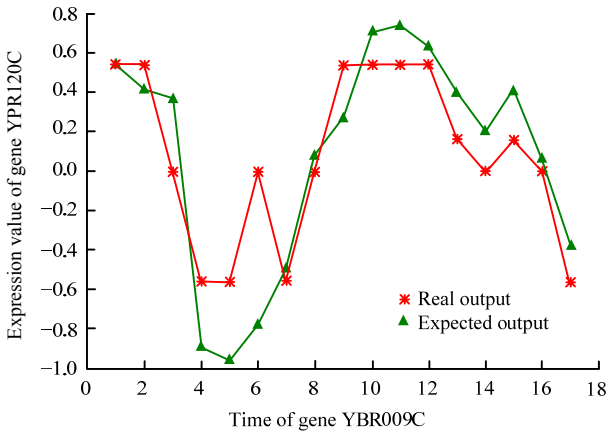


Fig. 8 Test curves of Relation 4. Total error is 2.7704, the average error of a single data is 0.046.

We carried out 20 experiments altogether with respect to the relations among genes X_1, X_2 and Y, in which we selected genes X_1, X_2 and regulator Y randomly from 5 periods in each relation and calculated their relations with our method. We could get results via comparing network errors and the similarity of the test curves. The result is good if the changing tendency of the real output curve is consistent with that of the expected output curve. The results are shown in Table 2.

Table 2 Results of four relations

Relation	Good results	Bad results
Relation 1	5	0
Relation 2	3	2
Relation 3	0	5
Relation 4	2	3

From the experiments above we can know that, in the modeling, if X_1, X_2 and Y are in Relation 1, the error gained from the model is small. The strength of the regulation and the error are in an inverse proportion. When we only calculate the relation of gene X_1 and regulator Y, we need assign X_1 to X_2.

When the calculation is completed, the set of rules will come out. The rule sets describe the relations of the regulation network. There may be conflicts in the sets because the constraint of the rules is not strong enough at the beginning.

There were 8 wrong results in the first experiment: 2 results existed in M/G1 period, 3 results appeared in G1, 1 result was in S/G2 and 2 results were in G2/M. There are two major reasons why mistakes happened in our model. One reason is that there is serious deficiency in some genetic data, which results in inability to calculate actual relations. The other reason is that there is a major regulator in each period, and the calculated error about it is always small. Therefore it is sometimes difficult to get the real strong regulator in this period.

In the second experiment, we obtained the regulators of the 7 genes. Some references also indicated that the 7 genes are regulated in cell cycle of yeast. But we are still unable to verify these regulators by now.

In the third experiment we reconstructed a gene regulatory network of 104 genes, and there were many regulation relations consistent with the regulation models in the SGD database.

5 Conclusion

In this paper, we proposed a novel approach for reconstruction of gene regulatory networks by integrating biological knowledge of a set of regulators, biology functions of genes, and a set of expression profiles. We determined the regulatory relationships among genes with fuzzy rules, such as activation, inhibition or no effect. Our approach is based on a neuro-fuzzy network model which could infer strong relations among genes. We obtained the strong regulator of each gene and their fuzzy rules, then we applied them to reconstruct partial gene regulatory network.

The number of fuzzy rules we got is usually more than one. There are two kinds of regulatory conditions. One condition is that regulated genes are influenced by its regulator which increases gene expression value, and the other is that the regulator decreases gene expression value. As a result, we reserved two kinds of fuzzy rules corresponding to the conditions mentioned above, which describe how regulators control genes. It should be

noticed that the model calculates the relations of three genes X_1, X_2 and Y each time, and we have to research thousands of genes, which is a time-consuming work Nevertheless, due to the theory of neuro-fuzzy network models, we are confident that the proposed approach could deal with even larger datasets in less time by distributing computations.

Experimental results validate the approach, which can obtain high accuracy in processing real biology data. Three works about gene regulatory networks reconstruction have been done. First, we obtained strong regulators of 97 known regulated genes; second, we got strong regulators of 7 unknown regulated genes; third, we reconstructed a partial gene regulatory network of these 104 genes with fuzzy rules.

In fact, the fuzzy rules are changing all the time, so it is not good enough to get only one rule to describe monolithic change. We need more details about the fuzzy rules. Future work will extend this model with appropriate structure and more effective algorithm which can reduce computing time and obtain time series of fuzzy rules.

Acknowledgement

This paper is supported by National Natural Science Foundation of China (Grant No. 60973092 and No. 60873146), the National High Technology Research and Development Program of China (Grant No.2009 AA02Z307), the "211 Project" of Jilin University, the Key Laboratory for Symbol Computation and Knowledge Engineering (Ministry of Education, China), and the Key Laboratory for New Technology of Biological Recognition of Jilin Province (No. 20082209).

References

[1] Shen X, Vikalo H. Inferring parameters of gene regulatory networks via particle filtering. *EURASIP Journal on Advances in Signal Processing*, 2010, **2010**, 204612.

[2] Loomis W F, Sternberg P W. Genetic networks. *Science*, 1995, **269**, 649.

[3] Albert R. Boolean modeling of genetic regulatory networks. In: Ben-naim E, Frauenfelder H, Toroczkai Z (eds.). *Complex Networks*, Springer, New York, USA, 2004.

[4] Margolin A A, Nemenman I, Basso K, Wiggins C, Stolovitzky G, Favera R D, Califano A. ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, 2006, **7**(Suppl 1), S7.

[5] Liu G X, Feng W, Wang H, Liu L, Zhou C G. Reconstruction of gene regulatory networks based on two-stage Bayesian network structure learning algorithm. *Journal of Bionic Engineering*, 2009, **6**, 86–92.

[6] Akutsu T, Miyano S, Kuhara S. Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symposium on Biocomputing*, 1999, **4**, 17–28.

[7] Chen K C, Wang T Y, Tseng H H, Huang C Y F, Kao C Y. A stochastic differential equation model for quantifying transcriptional regulatory network in *Saccharomyces cerevisiae*. *Bioinformatics*, 2005, **21**, 2883–2990.

[8] van Someren E P, Wessels L F, Reinders M J. Linear modeling of genetic networks from experimental data. *Proceedings of International Conference on Intelligent System for Molecular Biology*, 2000, **8**, 355–366.

[9] Friedman N, Linial M, Nachman I, Pe'er D. Using Bayesian networks to analyze expression data. *Journal of Computational Biology*, 2000, **7**, 601-620.

[10] Maraziotis I, Dragomir A, Bezerianos A. Recurrent neuro-fuzzy network models for reverse engineering gene regulatory interactions. *Lecture Notes in Computer Science*, 2005, **3695**, 24–34.

[11] Datta D, Choudhuri S S, Konar, A, Nagar A, Das S. A recurrent fuzzy neural model of a gene regulatory network for knowledge extraction using differential evolution. *Proceedings of the Eleventh conference on Congress on Evolutionary Computation*, Trondheim, Norway, 2009, 2900–2906.

[12] Floares A G. A neural networks algorithm for inferring drug gene regulatory networks from microarray time-series with missing transcription factors information. *Proceedings of the International Joint Conference on Neural Networks*, Atlanta, GA, USA, 2009, 848–854.

[13] Knott S, Mostafavi S, Mousavi P. A neural network based modeling and validation approach for identifying gene regulatory networks. *Neurocomputing*, 2010, **73**, 2419–2429.

[14] Ao S I, Palade V. Ensemble of Elman neural networks and support vector machines for reverse engineering of gene regulatory networks. *Applied Soft Computing*, 2011, **11**, 1718–1726.

[15] Echanobe J, del Campo I, Bosque G. An adaptive neuro-fuzzy system for efficient implementations. *Information Sciences*, 2008, **178**, 2150–2162.

[16] Lin C J, Lee C Y. A self-organizing recurrent fuzzy CMAC model for dynamic system identification. *International*

Journal of Intelligent Systems, 2008, **23**, 384–396.

[17] Salgado P. Rule generation for hierarchical collaborative fuzzy system. *Applied Mathematical Modelling*, 2008, **32**, 1159–1178.

[18] Zadeh L A. Fuzzy sets. *Information and Control*, 1965, **8**, 338–353.

[19] Cell Cycle Analysis. Cell Cycle Gene Expression Data in *Saccharomyces cerevisiae*, [2011-2-24], http://genome-www.stanford.edu/cellcycle/

[20] Spellman P T, Sherlock G, Zhang M Q, Iyer V R, Anders K, Eisen M B, Brown P O, Botstein D, Futcher B. Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell*, 1998, **9**, 3273–3297.

[21] Saccharomyces Genome Database, [2011-2-24], http://www.yeastgenome.org

[22] Predicted Regulatory Module. Saccharomyces Genome Database, [2011-2-24], http://db.yeastgenome.org/exp_modules/cellcycle/cluster_293.html