Reconstruct Feedback Control of Cell Cycle-Regulated Networks of the Yeast by Neural Network Computing

Shih-Yi Chao and Jung-Hsien Chiang

Department of Computer Science and Information Engineering
National Cheng Kung University, Tainan, Taiwan, R.O.C

Abstract-Cells continuously recycle their gene expressions. In order to understand the expressions of cell cycle-regulated genes, time series expression profiles provide a more complete picture than single time point expression profiles. However, these time series expression profiles raise new challenges for computer scientists and statisticians. One of these challenges is the reconstruction of the regulatory connections between genes, proteins, or other gene products. Recently, some analytic methodology or techniques have been constructed to model such time series data to discover gene regulatory networks. But most of these researches do not take account of the feedback control mechanism within a regulatory network. In our approach, a hybrid method is applied to reconstruction of cell cycle-regulated networks to determine gene interactions in gene expression data, especially to deal with the feedback mechanism of some particular genes. By using Radial Basis Function neural network (RBF) and Recurrent neural network (RNN), experiments conducted on real world Microarray expression data verify that this approach is sufficient for fitting the data set and reconstructing the feedback regulatory networks.

I. INTRODUCTION

As biology enters an area that genomes of several organisms have been completely sequenced, the next great challenge is determining gene regulatory networks. Time course DNA array data is created by measuring gene expression values over a number of time steps, while subjecting the cell or the organism to some stimulants, so that the conduct of genes in response to particular experimental conditions can be measured or observed potential cause-effect relationships. The goal of analyzing such data is to identify possible direct or indirect connections between genes. These connections between genes can be considered as a gene regulatory network \( N_i \) of positive, negative, or none connections by which transcription factors can regulate other genes up or down. There are many types of gene regulatory network models which have been proposed in the past. Their nature and composition are categorized by several factors: contemplating on gene expression (quantized or continuous values [1][2]), contemplating on causal relationship between genes (Bayesian analysis or Dynamic Bayesian [3][4]), and contemplating on time domain (discrete or continuous time[5]). However, the contemplating on Feedback control in genetic networks is also important. That is, some genes have unique characteristics, for instance, they regulate themselves. In regulatory networks, genes may have one or more activators or inhibitors which are regulated by their own gene products. Without the existence of activators or inhibitors, there is no way to complete the progress of gene transcription. Only partial sets of genes function as activators or inhibitors. Furthermore, lots of a cell’s activity is organizes as sets of genes co-regulated by particular proteins (or transcription factors) to respond to different conditions, for example, cell division. Therefore, the present challenge is to understand how cells control global gene expression programs, which means specific gene expression programs involve regulated transcription of many other genes or involve regulated transcription of themselves. Our approach aims to provide a system to construct regulatory networks that illustrate cell cycle related genes and their cause-effect relations to other gene products.

This paper is organized as follows. In Section 2, we describe the materials adopted by this paper; then, the Radial Basis Function neural network (RBF) module is adopted to classify cell cycle-regulated transcription factor proteins and to reconstruct gene transcriptional regulatory network by Recurrent neural network (RNN). Finally, experimental results and discussions are presented.

II. METHODS

A. Materials

1) DNA Microarray experiments

In biological, each cell is the product of specific gene expression programs involving regulated transcription of thousands of genes. These transcriptional programs are modified through the cell cycle progress. Data from Spellman et al. is originally downloaded from the reference web site [6]. The goal of yeast cell cycle analysis from Spellman et al. is to identify all the genes whose mRNA expression levels are regulated by the cell cycle. In our research, we use this data set to reconstruct regulatory networks and target genes which are regulated by some specific transcription factors. Moreover, we aim at transcription factors that are auto-regulated or feedback controlled to build regulatory networks.

2) Transcription factor binding sites

Our computational method attempts to integrate gene expression data and sequence data. As a result, we collect known transcription factor binding sites sequences for yeast from SGD[7], TRANSFAC[8], and YPD of Incyte Proteome BioKnowledge Library[9], which are listed in Table 1.

3) The Network motifs

According to Tong Ihn Lee et al. [10], there are six models for transcriptional regulatory networks. Three of these models, which include the feedback mechanism, are described in Figure 1. Circles represent transcription factors, and rectangles represent target genes. Solid arrows represent regulatory relationship between transcription factors and their target genes. Dotted arrows indicate transcription factors and their product genes. Figure 1-(a) Auto-regulation, this network consists of a
regulator that binds to the promoter region of its own gene. Multi-component loop, illustrated in Figure 1-(c), consists of a regulatory circuit whose closure involves two or more factors. Figure 1-(b) is the Feed-forward loop network motif, which contains a regulator that controls a second regulator and has the additional feature that both regulators bind a common target gene. The left two modes are Figure 1-(d) and (e), which include no feedback connections, are single input motif and multi-input motif, respectively. The multi-input network consists of a set of regulators that bind together to a set of genes. Take feed-forward motif as an example, it contains a regulator that controls a second regulator (or controls more other regulators), and have the additional feature that both regulators bind a common target gene.

It seems that transcription factors can be exactly divided into different kinds of network motifs by their characteristics of regulator functions. Nevertheless, we observe that some transcription factors play critical roles in varied network motifs. For instance, the regulator functions of ACE2 are fit for feed-forward loop, single input motif and multi-input motif. The SWI4 transcription factor belongs to single input motif, multi-input motif, feed-forward motif, and auto-regulate motif. For that reason, we re-group transcription factors into several catalogs and list some examples in Table 2. Deserving to be mentioned, the last catalog is non-transcription factor genes, and is encoded as “000” label while training RBF classification. We randomly select some known target genes for training sets of “others” catalog, such as DSE1 gene and so on.

Table 1. Examples of Training set of Yeast transcription factors and their specific binding sites sequences that used by RBF classification.

<table>
<thead>
<tr>
<th>ORFs</th>
<th>TF</th>
<th>Network Motif</th>
<th>Binding sites sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>YKL112W</td>
<td>ABF1</td>
<td>Feed-forward loop</td>
<td>RTCAYTNNNNACGW</td>
</tr>
<tr>
<td>YLR131c</td>
<td>ACE2</td>
<td>Feed-forward loop</td>
<td>TGCTGGT</td>
</tr>
<tr>
<td>YIL131C</td>
<td>FKH2</td>
<td>Feed-forward loop</td>
<td>GTAAACA, RYMAAYA</td>
</tr>
<tr>
<td>YMR043W</td>
<td>MCM1</td>
<td>Multi-component</td>
<td>CCNRATNNNG, WTWCCYAAWNNNGTJAA</td>
</tr>
<tr>
<td>YKL062W</td>
<td>MSN4</td>
<td>Multi-component</td>
<td>AAGGGG</td>
</tr>
<tr>
<td>YMR021C</td>
<td>MAC1</td>
<td>Multi-component</td>
<td>TTTGCTC</td>
</tr>
<tr>
<td>YNL216W</td>
<td>RAP1</td>
<td>Auto-regulation</td>
<td>CAYCCRTCRA, WRMACCCTACAYY</td>
</tr>
<tr>
<td>YHR084w</td>
<td>STE12</td>
<td>Auto-regulation</td>
<td>ATGAAACA</td>
</tr>
<tr>
<td>YER111c</td>
<td>SWI4</td>
<td>Auto-regulation</td>
<td>CACGAAA</td>
</tr>
</tbody>
</table>

B. Discover Candidate Transcription Factors by RBF

According to [10], all 106 Transcription Factors that are segregated into functional categories based on the primary functions of their target genes, for example, cell cycle, DNA/RNA/Protein biosynthesis, metabolism and environmental response. Transcription Factors in different categories are responsible for dissimilar mission, just like their behavior. Not alike [11] and [12], they predict regulatory networks by non-grouping Transcription Factors. We follow naive rules in Saccharomyces cerevisiae species, namely, we group Transcription Factors according to the primary function in advance. After that, we select cell cycle related transcriptional regulators and find out the expression behavior of each regulator by neural network and then perform to predict candidate transcription factors. RBF is first introduced in the solution of the real multivariate interpolation problem [13] and [14]. The RBF is designed for curve-fitting or approximation problem in a high-dimensional space. From this viewpoint, learning is equivalent to finding a surface in a multidimensional space that provides a best fit to the training data. In other words, learning RBF in our approach is in terms of the criterion for “the best fit” being measured in Microarray gene expressions sense. It is not always desirable to exactly interpolate scattered Microarray data when the data is contaminated with noise. A smooth approximation is often more useful to represent a gene’s expression profile. Consequently, the RBF classifier is capable of categorizing whether the genes are cell cycle related transcriptional regulators or not, in accordance with gene’s expression profile.

Cell cycle related transcription factors are quite special for the existence of lives. There are several reasons why genes might be regulated in a periodic manner coincident with the cell cycle. Such regulation might be required for the proper functioning of mechanisms that maintain order during cell division. The cell cycle is comprised by G1 phase: growth and preparation of the chromosomes for replication; S phase: synthesis of DNA; G2 phase: preparation for M phase: mitosis [15]. The special behavior of the chromosomes in mitosis requires some special controls. We’ve observed some well known cell cycle related Transcription Factors before, and found some expression profiles in particular. Take Figure 2. as an example, those ORFs (YLR131c, YKL112W, YOR028C, YPL177C, YPR104C, YMR021C, and YDL056W) in yeast have been proved with relation to cell cycle. The x-axis represents 17 time points. Observing that from time point 1 to 17, including two mitosis phases, those four ORFs have analogous expression profiles during these time points. To reveal this characteristic, we train RBF by some constraints on pre-processing. We assume that time points during mitosis phases are important expression features for cell cycle related Transcription Factors. That is, the emphasis of time points during mitosis phases would be greater in heft than other time points. Hence, let $\alpha>0$, $\beta>0$, $\gamma>0$, $\delta>0$, and we list some constraint conditions as follows:

\[
\begin{align*}
    h_{14} &= h_{15} + \alpha \\
    h_{16} &= h_{15} + \beta \\
    h_i &= h_{i} + \gamma \\
\end{align*}
\]

(1)  
(2)  
(3)
\[ h_i = h_i + \delta \quad (4) \]
\[ \sum_{i=1}^{n} h_i = 1, \text{and} \ h_i \geq 0 \quad (5) \]

where \( h_i \) (\( i = 1, 2, \ldots, 17 \)) represents heft of each pre-processing node. We re-scale \( h_i \) by the following constraints:

\[ 3 \times h_i + \alpha + \beta \geq 0.25 \quad (6) \]
\[ 3 \times h_i + \gamma + \delta \geq 0.25 \quad (7) \]
\[ \sum_j a_j h_j = 3(h_0 + h_i) - (\alpha + \beta + \gamma + \delta) \quad a_j = 0 \quad j = 7, 8, 9, 14, 15, 16 \quad (8) \]

where \( a_j \) (\( j = 1, 2, \ldots, 17 \)) represent the pre-processing node. Under this approach, the RBF network architecture is one input layer with two kinds of input data, one hidden layer, and one output layer. There are 17 nodes in input layer representing the original 17 time point gene expression values and re-scaled between 0 and 1. The other input nodes indicate the sequences of known transcription factor binding sites, which are illustrated in Figure 3. The known sequences of transcription factor binding sites obtained from [7], [8], and [9], are encoded into numeric between 0 and 1.

**TABLE 2. SOME EXAMPLES OF RE-GROUPING TRANSCRIPTION FACTORS**

<table>
<thead>
<tr>
<th>Gene</th>
<th>Single Input Motif</th>
<th>Multi Input motif</th>
<th>Feed-forward</th>
<th>Auto-regulate</th>
<th>Encoded catalog number of RBF classifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWI4</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td>001</td>
</tr>
<tr>
<td>ACE2</td>
<td>√</td>
<td>√</td>
<td>√</td>
<td></td>
<td>010</td>
</tr>
<tr>
<td>SUM1</td>
<td>√</td>
<td></td>
<td></td>
<td>√</td>
<td>100</td>
</tr>
<tr>
<td>ARO80</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td>011</td>
</tr>
<tr>
<td>TAP6</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td>101</td>
</tr>
<tr>
<td>REB1</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td>110</td>
</tr>
<tr>
<td>FKH1</td>
<td></td>
<td></td>
<td></td>
<td>√</td>
<td>111</td>
</tr>
<tr>
<td>others</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>000</td>
</tr>
</tbody>
</table>

Next, we briefly describe the input layer and hidden layer. Let \( X \) represent input layer vector, and \( G_i \) (where \( i = 1, 2, \ldots, n \)) represent neurons in hidden layer, which is a Gaussian kernel in the form:

\[ G_i(X, \mu_i) = \exp\left( -\frac{1}{2\sigma^2} \| X - \mu_i \|^2 \right) \quad (9) \]

where \( \mu_i \) is a vector representing the center of the \( i \) kernel and \( \sigma^2 \) is the corresponding variance. The output layer implements a weighted sum of hidden-output units:

\[ F(X) = \sum_{i=1}^{m} w_i G(X, \mu_i) - \theta \quad \text{for} \ i = 1, 2, \ldots, m \quad (10) \]

where \( w_i \) are the output weights, each corresponding individually to the connection between a hidden neuron and an output neuron. Later we use gradient descent to determine the weights of the network. Finally, the vector \( \theta \) represents biases.

![FIGURE 1. TRANSCRIPTIONAL REGULATORY NETWORK MOTIFS: THIS DIAGRAM IS MODIFIED FROM [10]](image1)

![FIGURE 2. FLUCTUATION OF TRANSCRIPT LEVELS DURING CELL CYCLE. PATTERNS OF NORMALIZED TRANSCRIPT LEVELS DIVIDED BY MEAN VALUE AGAINST TIME FOR YLR131C, YKL112W, YOR028C, YPL177C, YPR104C, YMR021C, AND YDL056W.](image2)

![FIGURE 3. THE ARCHITECTURE OF RBF CLASSIFIER. THE SHADEd CIRCLES INDICATE THE SEQUENCES OF TRANSCRIPTION FACTOR BINDING SITES.](image3)

**C. Reconstruction of Regulatory Networks**

Recurrent networks are neural networks with one or more feedback loops. Given a multilayer perceptron as the basic building block, we may have feedback from the output neurons of the multilayer perceptron to the input layer. Another possible form of feedback is from the hidden neurons of the network to the input layer [16]. Since the mechanism of gene transcriptional regulation is a sort of chain reaction over time, we can treat each set of gene expressions as a set of random variables over time. The disadvantage of DBN is that it can not scale well to large-scale datasets. The expression level of a gene at a particular time point can then be estimated by the weighted sum of the expression levels of all genes in the network at a previous time point for the linear additive regulation.
models [3][4]. This kind of linear additive regulatory relationships can reveal some linear relations in a regulatory network, but it lacks the capability to grab the nonlinear dynamics regulatory relationships between genes. Accordingly, we reconstruct gene transcriptional regulation networks by RNN. By using RNN to infer regulatory networks, we are interested in the capability of RNN to derive the complex behaviors of genes.

As described in section 2, we have various cell cycle-regulated transcription factors classified by RBF, and the next process is to find out the target genes. However, it is another demand to determine the target genes for certain transcription regulators. To resolve this problem, it is required that performing global search for optimal target genes which genetic algorithms produce. Hence, the neural computing behaviors of genes.

In the capability of RNN to derive the complex regulatory networks, we are interested in the nonlinear dynamics regulatory relationships can reveal some linear relations in a regulatory network, but it lacks the capability to grab the complex behaviors of genes.

As described in section 2, we have various cell cycle-regulated transcription factors classified by RBF, and the next process is to find out the target genes. However, it is another demand to determine the target genes for certain transcription regulators. To resolve this problem, it is required that performing global search for optimal target genes which genetic algorithms produce. Hence, the neural computing method adopted in this paper combines the GA with the RNN architecture to form a hybrid system. The combined algorithm works as follows:

Input: various cell cycle-regulated transcription factors
Output: various cell cycle-regulated transcription factors
Procedure:
1. Random choose one cell cycle-regulated transcription factor A as the “input” gene, geneA.
2. If geneA is labeled as “010” described in Table2, then run STEP 3 ~ 9 for three times, one for single-input motif RNN architecture, another one for multi-input motif RNN architecture, and the other one for feed-forward motif RNN architecture. It infers that, we’ve trained several RNN architectures for different network motifs that are described in section 2.1.
3. Use the genetic algorithm (GA) [17] to generate combinations of M genes (gene1, gene2, …, geneM) to represent the target genes that are regulated by geneA. Each combination is a chromosome. The initial set of combinations comprises of the initial population of chromosomes.
4. The training set, including initial population of chromosomes and geneA, will consist of the Microarray expression values from time point 1 to 17, and the initial population of chromosomes will be the target output for the RNN.
5. Execute the gradient descent algorithm on this training data via the RNN to determine the weights between the input genes and the output genes until stopping criterion is met.
6. Return the RMSE of RNN to GA. This is the fitted value for particular chromosome.
7. Repeat 3, 4, and 5 for each chromosome.
8. Repeat STEP 3-6 as a GA run, using crossover and mutation operators on all chromosomes to alter the choice of output gene combinations.
9. When some stopping criterion is met, the GA stops. Record the best chromosome and the weights of the RNN.

END
Change another cell cycle-regulated transcription factor (i.e. geneA), until no transcription factors are left. When all the steps described above are completed, regulatory network structures can be derived to comprise the connections of the transcription factors and their target regulated genes. The parameters for the GA operators and RNN parameters are shown in Table 3.

<table>
<thead>
<tr>
<th>GA parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossover</td>
<td>One Point, crossover rate (0.8)</td>
</tr>
<tr>
<td>Mutation</td>
<td>Random, mutation rate (0.1)</td>
</tr>
<tr>
<td>Selector</td>
<td>Roulette Wheel</td>
</tr>
<tr>
<td>Population Size</td>
<td>50 – 250</td>
</tr>
<tr>
<td>Generations</td>
<td>100 – 1000</td>
</tr>
<tr>
<td>RNN parameters</td>
<td>Values</td>
</tr>
<tr>
<td>Epochs</td>
<td>50 – 100</td>
</tr>
<tr>
<td>Gradient descent</td>
<td>Standard</td>
</tr>
<tr>
<td>Weight Update</td>
<td>Online</td>
</tr>
</tbody>
</table>

D. System Procedures
The whole system is implemented by matlab 6.5, and we briefly introduce the overall system procedures by Figure 4. The GA uses a standard random mutation and a standard binary representation with one point crossover. To select the “good” combinations of target genes, we select appropriate GA mutation and crossover operators to alter the chromosomes. One chromosome of the GA represents a number of genes taken from the full set of genes and is used by RNN to check how well the expression values of this number of genes affected by particular transcription regulators. On the other hand, the GA consists of populations of such chromosomes, and each chromosome is evaluated by the RNN for its fitness value to the given transcription regulators. The choice of RNN architecture is according to the labels which RBF classifier assigns. Furthermore, the final returned RNN output error (RMSE) is treated as a fitness value for some particular combinations of target genes.

The stopping criterion includes not only the fitness value fits some criteria but also the determination of the RNN choosing steps. In other words, the GA never stops until all appropriate RNN architectures are executed. Take gene SWI4 listed in Table2 as an example, it is labeled “001”; which means the regulatory functions of SWI4 are fit for single-input motif, multi-input motif, feed-forward and auto-regulate motif, and SWI4 is trained for four different RNN architectures for each network motif. In that case, each transcription factor can chose suitable RNN architecture more than one time, and find out dissimilar set of target regulatory genes. After all candidate transcription factors are run by this system procedures and output regulatory networks, the GA is then complete.
III. EXPERIMENTAL RESULTS

The yeast cell cycle data set consists of almost 6300 genes and 17 time points of Microarray value. Consequently, the number of gene combinations increases significantly. We illustrate some experimental results with Table 4. In these experiments, the number of epochs of RNN is 100, and the number of generations of the algorithms is varied while recording the error rates for the training data. The operators, crossover rate and mutation rate, run on the data are 0.8 and 0.1, respectively. The population size is 250 for all generations listed in Table 4. Notice that, the average RMSE is decreasing while the number of generations is increasing. The last column in Table 4 indicates the minimum RMSE within the populations. For example, throughout the 800 generations, the RMSE value of the best fitted target sets of genes that RNN trained is about 0.18.

<table>
<thead>
<tr>
<th>GA generations</th>
<th>Average RMSE</th>
<th>The minimum RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>3.43621939</td>
<td>0.241770628</td>
</tr>
<tr>
<td>200</td>
<td>1.167214745</td>
<td>0.216831078</td>
</tr>
<tr>
<td>400</td>
<td>1.579679118</td>
<td>0.218843935</td>
</tr>
<tr>
<td>600</td>
<td>0.849939818</td>
<td>0.209291992</td>
</tr>
<tr>
<td>800</td>
<td>0.80232844</td>
<td>0.1897575</td>
</tr>
<tr>
<td>1000</td>
<td>0.301333701</td>
<td>0.174043958</td>
</tr>
</tbody>
</table>

Table 4: The experimental results of GA with RNN

The overall transcriptional regulatory network can be constructed based on the results that come out from RNN and GA. Figure 5 shows some examples of the estimated regulatory network connections with either auto-regulation or other feed-forward interactions. The analyses demonstrate that our proposed approach is capable of identification of significantly regulatory network structures and corresponding regulatory interactions. Among the reconstruction regulatory networks, some have been confirmed by biological experiments. For instance, the Gene Ontology (GO) annotations [18] of ACE2 are “transcriptional activator activity”, and “G1-specific transcription in mitotic cell cycle” [19]. It has been proved that SWI5 and ACE2 are cell cycle-regulated transcription factors that activate expression of early G1-specific genes in Saccharomyces cerevisiae. SWI5 and ACE2 also activate expression of a number of other genes expressed in G phase of the cell cycle, including ASH1, EGT2, PMA1, PCL9, and SIC1. The GO annotations of SWI4 are “G1/S transition of mitotic cell cycle” [20] and “cell cycle” [21]. They also confirm the efficiency of the reconstructions of regulatory networks. According to [22], SWI4 and SWI6 play a crucial role in START-specific transcription in Saccharomyces cerevisiae. They analyze the protein complex of SWI4 and SWI6, which exhibit these two genes direct G1/S-specific transcription in yeast. Figure 5 also shows the RAPI and SWI4 are auto-regulate transcription factors, which are already improved biologically: FKH2 regulates cell cycle dependent expression of the CLB2 cluster of genes, which include SWI5 and ACE2[23].

IV. DISCUSSION

The reconstruction of regulatory networks from temporal expression data is one of the most instant problems in computational biology. It is acknowledged that, the causes of heterogeneity genetic-related circumstances, for examples, the cell cycle, or diseases such as cancers, are products of complex interactions between genes over time. The analysis of gene expression data will become increasingly widespread. When appraising approaches for discovery of gene regulatory networks, the number and type of sources of data must be taken into account. Besides, the approach must be capable of handling noisy and high dimensional gene expression data. The approach described here, has been shown to be effective on real-world expression data. The stochastic nature of GA means that there are not the same results can be expected from each run of the algorithm, and the GA is run for a fixed number of generations for each output of regulatory networks. However, to increase the number of genes that the GA can select from, which in turn could require more GA generations.

As a result, increasing the GA generations also increases the computational time. But it does show that results on Microarray data can be discovered correctly by GA used in our approach. In addition, this approach builds regulatory networks piece by piece.
piece. Imagine that one network motif is one regulatory mechanism unit with specific set of genes, including the influences and the targets. We discover all the formed units one by one, and eventually join these units by their simultaneously occur transcription regulators. The above-mentioned shows the advantages of generating smaller but more precise regulatory networks, in that each of the path or the unit in the networks can be seen without being masked by other connects. It is not the same as traditional complicated regulatory relationships that too many to visualize as a network to yield information in an indigestible format for biologists.

Also, the order of regulatory processing can be displayed faithfully by this approach, especially the feed-forward network motif, which contains a regulator controls a second regulator and has the additional feature that both regulators bind one or more common target genes. The phenomenon of this biology mechanism is often seemed, but ignored while reconstruct regulatory networks. It is a major contribution provide by this approach that considering not only the cause-effect relationships between genes but also the chain influences upon a set of genes sequentially.

V. CONCLUSIONS

We’ve developed a neural computing approach to reconstruct the gene regulatory network in silico. Upon the Microarray data and the sequences of transcription factor binding sites, the approach has been shown to be able to accurately fit the data on which it is trained. We also observe that some transcription factors play critical roles in varied network motifs. In other words, some functions of transcription regulators are fit for several kinds of regulatory network motifs. We adopt these characteristics by training the RBF classifier for categorizing cell cycle-regulated transcription factors. Additionally, it has been shown that the RNN and GA to fit the feedback and feed-forward network motif. The experimental results have been shown that our approach by using GA and RNN has the capability of finding regulatory networks. RNNS with diversified architectures indicate varied network motifs to reconstruct complete regulatory networks with loop mechanism. Combined RNN with GA, provide the global searching capacities of finding proper target regulated genes for some particular transcription regulators. The chromosomes that GA used are combinations of target genes and by using crossover and mutation operators on all chromosomes to alter the choice of output gene combinations. This approach is on the basis of both gene expression data and sequences data, so it is time significant and binding region significant data analysis. Finally, since this method has been previously shown to also classify cell cycle-regulated transcription factors as well and then reconstruct regulatory networks, it can be considered a candidate multipurpose tool for Microarray expression data analysis.

REFERENCES


