Inferring gene correlation networks from transcription factor binding sites

Ghasem Mahdevar¹, Abbas Nowzari-Dalini^{2*} and Mehdi Sadeghi^{3,4}

¹Department of Bioinformatics, Institute of Biochemistry and Biophysics, University of Tehran, Enghlab Avenue, 1417614411, Tehran, Iran

²Department of Computer Science, School of Mathematics, Statistics, and Computer Science, University of Tehran, Enghlab Avenue, 1417614411, Tehran, Iran

³National Institute of Genetic Engineering and Biotechnology, Shahrak-e Pajooresh, km 15, Tehran - Karaj Highway, P.O. Box:14965/161, Tehran, Iran

⁴School of Biological Sciences, Institute for Research in Fundamental Sciences (IPM), No. 1, Shahid Farbin Alley, between embassy of Italy and Kamranie crossroad, Shahid Lavasani St., Tehran, Iran

(Received 30 December 2013, accepted 21 February 2014)

Gene expression is a highly regulated biological process that is fundamental to the existence of phenotypes of any living organism. The regulatory relations are usually modeled as a network; simply, every gene is modeled as a node and relations are shown as edges between two related genes. This paper presents a novel method for inferring *correlation networks*, networks constructed by connecting co-expressed genes, through predicting co-expression level from genes promoter's sequences. According to the results, this method works well on biological data and its outcome is comparable to the methods that use microarray as input. The method is written in C++ language and is available upon request from the corresponding author.

Key words: gene expression, neighbor joining, neural networks, self – organizing maps

INTRODUCTION

Gene expression is at the basis of molecular biology. Through this procedure, cells use the information stored in DNA, *the genes*, to perform and control the vast number of functions required for them to survive in various conditions (Davidson et al., 2003; Lodish et al., 2007). Gene expression consists of a series of regulated processes, to be precise, transcription, RNA modifications (eukaryotic genes), and translation. By regulating these processes and thereby controlling the amount of produced proteins, cells can respond to any stimulus (Brazhnik et al., 2002; Latchman, 1998; B. Lewin, 2004).

Recently, there have been many efforts for discovering how genes regulate expression of themselves (Hecker et al., 2009; Huynh-Thu et al., 2010; Kabir et al., 2010; Meyer et al., 2007; Yip et al., 2010). *Transcription*, the first step in the gene expression which converts DNA to RNA via a complex machinery is an extremely regulated process (Jacob and Monod, 1961; Lodish et al., 2007). This step is mainly dependent upon the successful binding of *Transcription Factors (TFs)* proteins to explicit

positions in genes upstream or promoters, also known as *TF Binding Sites (TFBSs)* (Chua et al., 2004; B. Lewin, 2004; Mahdevar et al., 2012). In general, 4% of all genes produce TFs and usually multiple TFs act together to regulate the expression of a given gene (Bar-Joseph et al., 2003; Schlitt and Brazma, 2007; Yu et al., 2003). Unfortunately, finding these TFs and their corresponding TFBSs is costly and laborious; therefore, computational discovery of TFs and TFBSs has attracted considerable attention (Bailey and Elkan, 1994; GuhaThakurta, 2006; Hertz and Stormo, 1999; Mahdevar et al., 2012; Thijs et al., 2002; Tompa et al., 2005).

Information flow from DNA to proteins begins with transcription process; besides, it is the origin of response to numerous stimuli (Latchman, 1998; Lodish et al., 2007). As a result, regulation of transcription machinery is more crucial to cells than other aforementioned regulatory steps (Davidson, 2001).

TFs that involve in transcription process are also proteins, i.e., products of some genes can influence the expression of other genes. This means that genes are in a network and interact with each other in order to react to stimuli or to perform vital processes of cells. Of course, there are other biological networks in cells, e.g., protein – protein – interaction networks or metabolic

Edited by Ali Masoudi-Nejad

* Corresponding author. E-mail: nowzari@ut.ac.ir

pathways. These networks are usually depicted as nodes connected by edges. Nodes represent genes, proteins, or metabolites that are cell response to stimuli. Edges often represent relations, such as the binding of a TF to its target TFBS or direct molecular interactions.

Understanding how genes regulation occurs and identifying the interactions between genes in a living system, namely, *network inference*, are important topics in today's biology. In this paper we have focused on the later topic.

In general, there are two strategies for modeling gene networks: physical modeling and influence modeling (Gardner and Faith, 2005). Following two paragraphs describe these strategies in brief.

Physical approach tries to find true molecular interaction between TFs and TFBSs that control transcript synthesis. Specifically, in physical approach the goal is to identify the TFs and the corresponding TFBSs that regulate transcription process. Restricting regulators to TFs made this approach easy to apply but also weak to infer the network and to describe regulatory mechanism.

The goal in the influence approach is to discover – indirect – regulatory influence of each transcript on the synthesis of all transcripts; that is, these models do not utilize or consider proteins, TFBSs, or TFs data explicitly, but, implicitly.

Based on both approaches, various methods with their advantages and pitfalls have been introduced to model gene networks, to name a few, Correlation Networks (Butte and Kohane, 2000; D'Haeseleer et al., 2000; Filkov et al., 2002; Margolin et al., 2006; Stuart et al., 2003), Bayesian Networks (Friedman et al., 2000; Husmeier, 2003; A. Lewin et al., 2006; Perrin et al., 2003), Boolean Networks (Akutsu et al., 1999; Kauffman, 1969; Mehra et al., 2004), Differential Equations (Chen et al., 1999; Gebert et al., 2007), and Neural Networks (Tian and Burrage, 2006; Vohradsky, 2001).

After choosing the network model, network parameters, specifically, scores of edges between all pairs of genes or nodes in the network must be inferred through available data. As aforesaid, this process is known as *network inference* or *reverse engineering*.

High-throughput technologies such as Microarray have led to an exceptional growth in data related to sequence features and gene expression. Essentially, four kinds of data are available to employ in gene networks inference process, namely, 1 – amount of existing transcripts, 2 – proteins, or 3 – metabolites and 4 – genome data.

Measuring the concentrations of RNA transcripts, produced proteins, or metabolites give an insight into the gene expression and are the fundamental resources for several studies. Comprehensive overviews of these methods studies can be found in (Barabasi and Oltvai, 2004; DeJong, 2002; Filkov, 2005; Hecker et al., 2009; Sirbu et al., 2010).

Genomics data, e.g., genes and TFBSs, are supportive

to the networks inference: relations between TFBSs and gene expression can be revealed by this kind of data. Methods that use genome sequence data fall into two categories: those that rely on prior knowledge of TFBSs, and those that discover TFBSs by computational effort and do not take prior information about TFs and TFBSs into consideration. The research work done by Beer and Tavazoie (2004) is a good example for the first category; they employed both known instances of TFBSs and novel TFBSs extracted from promoter sequences to predict the expression level of *Saccharomyces cerevisiae* (yeast) genes by training a Bayesian Network on microarray data of environmental stress and cell cycle (Beer and Tavazoie, 2004). Accuracy of their method on yeast genome was around 0.70. In a more recent work that belongs to the latter category, Pavesi and Valentini (2009) applied a machine learning approach to classify sets of co-expressed genes using information extracted from their regulatory regions.

In this paper, we propose a new method that infers genes correlation network from their promoter sequences by using a neural network without exploiting any a priori knowledge about input promoters, their potential TFBSs, or need to clustered input. A collection of genes promoter sequences is the only input of the proposed method and the result is correlation network of genes corresponding to these promoters. In brief, the method begins by extraction of potential TFBSs from all input sequences through computation; then, it tries to find expression similarity between all pairs of inputs with respect to the discovered TFBSs by a specific type of neural networks; at last, it uses calculated similarities to find groups of inputs and to build correlation network of given genes. Performance and limitations of the proposed method on inferring correlation networks from both simulated and biological data are presented in the result section.

METHOD

In this section, we explain the proposed method in detail. Simply, the input of the method is a collection of promoter sequences, $\mathbf{P} = \{p_1, p_2, \dots, p_n\}$, with n sequences of arbitrary lengths; and the output is correlation network of input, inferred from properties of computationally discovered TFBSs that exist among input promoter sequences. The predicted network is shown by graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ with n nodes.

In order to assign weight to edge set $\mathbf{E}$ from discovered collection of TFBSs, a neural network and a neighbor joining method are utilized, as follows: TFBSs occurring in $\mathbf{P}$ are used as inputs to the neural network, its outcome will be the amount of correlation between expression patterns of two $\mathbf{P}$ members. At last, an application of neighbor joining method to these correlations yields elements of $\mathbf{E}$.

The general steps of our method, including Extracting

Promoters Information, Finding Expression Correlation, and Building Correlation Network are described in the following subsections.

Extracting promoters information Aim of this step is to extract information of all input sequences. TFBSs, short subsequences in promoter sequences with high frequency in comparison to random sequences (Das and Dai, 2007; GuhaThakurta, 2006), are very important to transcription machinery (Jacob and Monod, 1961). Because of their higher frequency, it is possible to discover known and novel TFBSs by searching for overrepresented subsequences. Certainly, many computationally TFBSs discovery methods are available; for an overview of these methods see (MacIsaac and Fraenkel, 2006; Tompa et al., 2005). Among them, AmotiF tool (Mahdevar et al., 2012) has good sensitivity and specificity, runs fast, and is planned to find overrepresented pattern with respect to the biological facts about TFBSs (See Mahdevar et al. (2012) for more details about this method). Therefore, we have chosen AmotiF tool to find TFBSs. Experiments with other TFBSs discovery methods confirm the validity of this choice: AmotiF achieved the best accuracy in the discovery of random TFBSs planted in a set of sequences. Precisely, the accuracy of AlignACE (Roth et al., 1998), AmotiF (Mahdevar et al., 2012), BioProspector (Liu et al., 2001), and MEME (Bailey and Elkan, 1995) in finding 10 TFBSs planted in 100 random sequences of 500 nucleotides was 39%, 62%, 56%, and 53%. Where, TFBSs range from 6 to 20 base pairs in length and have 5 to 10 instances with 60% to 90% identity. Furthermore, AmotiF gained the highest score in the famous assessment provided by Tompa et al. (2005), as stated by Mahdevar et al. (2012).

By investigating TFBSs of *S. cerevisiae* Promoter Database (SCPD) (Zhu and Zhang, 1999) and TFBSs of *Escherichia coli* Database (RegulonDB) (Salgado et al., 2004), we have found that TFBSs positions are far from being random, to be exact, TFBSs positions have little standard deviation: 28% of maximum value, which is consistent with the results of Beer and Tavazoie (2004). Moreover, we have found that in these two species 70% of all TFBSs are placed in the first 250 nucleotides of the promoter sequence. We have employed these properties to adjust scores of TFBSs returned by AmotiF; simply, by reducing score of TFBSs that violates them.

Existing TFBSs discovery methods may split a TFBS into two or more TFBSs. For instance, they may report first few nucleotides of a TFBS as a single TFBS and the rest of it as another one. Thus, the proposed method merges TFBSs whose occurrences are very similar in all input sequences into one TFBS. Furthermore, the proposed method removes TFBSs that exist in almost all inputs, since these types of TFBSs have no discriminative information.

At last, score of every TFBS in each input promoter will

be saved in matrix $\mathbf{T} = [t_{i,j} | 1 \leq i \leq n, 1 \leq j \leq m]$, where m is the number of remaining TFBSs and $t_{i,j}$ is the score of j th TFBS in the i th sequence, normalized by dividing it to the square root of number of nonzero $t_{k,j}$, where, $1 \leq k \leq n$. Explicitly,

$$t_{i,j} \leftarrow \frac{t_{i,j}}{\sqrt{|\{t_{k,j} | t_{k,j} \neq 0, 1 \leq k \leq n\}|}}, \quad (1)$$

for all $1 \leq i \leq n$ and $1 \leq j \leq m$.

This normalization reduces scores of weak TFBSs interspersed throughout the input promoters.

Finding expression correlation Here, the goal is to achieve expression correlation between all two members p_i and p_j of $\mathbf{P}$ from matrix $\mathbf{T}$ and store them in matrix $\mathbf{C} = [c_{i,j} | 1 \leq i, j \leq n]$. We have developed a *Self-Organizing Map (SOM)*, a special type of *neural networks*, for this purpose.

We choose SOM because it is an unsupervised clustering method with no needs to specify the exact number of clusters or even estimating this number by fine-tuning a parameter. SOM is relatively stable, e.g., its outcome is not dependent to the initial weights, in contrast to methods such as K-means clustering (Martin et al., 2007). SOMs are not prone to over-fitting or overlearning (Meissner et al., 2009) and their training procedure is just a simple iteration (Kohonen, 1989). Furthermore, replacing SOM by other clustering methods (e.g. K-means or hierarchical clustering) has resulted in worse performance.

Fundamentally, neural networks are particular type of machine learning algorithms which consist of a set of neurons and applicable for pattern classification, identifying correlations in the data, and many more problems. There exist two kinds of learning for a neural network: *supervised* and *unsupervised*.

In supervised learning, training patterns with known classification repeatedly are presented to the network and network parameters or weights are adjusted such that the number of misclassified inputs decreases. These two procedures, i.e. presenting patterns to the network and adjusting the weights of the network, are iterated until some convergence criterion is satisfied.

Unsupervised learning is appropriate when there is no prior information about classification of inputs. In unsupervised learning, the task is to classify or to cluster inputs such that close members, or members who belong to single class or cluster, share common properties or show higher similarity according to a predefined measure.

SOMs are unsupervised neural networks suitable to both classifying and clustering data according to their similarities, consisted of two layers of neurons (Kohonen, 1989): input layer and output layer, as illustrated in Fig. 1.

Input of the developed SOM is information collected from every input promoter sequence, stored as rows of $\mathbf{T}$; therefore, the network has m neurons in its input layer.

Fig. 1. SOM architecture with 3 input neurons and 4 output neurons.

Each input neuron connects to every output neurons via variable connection weights. Weights are random at start and adjusted during training process, such that close nodes become more sensitive to similar rows.

In order to get to stable weights, *training* or presenting whole input to the network and adjusting its weights, should be iterated multiple times. By examination, we have found that the number of training (λ) should at least be proportional to n and $50 \times n$ is an appropriate value. Networks weights would eventually become fixed, because the training procedure has a decreasing variable called the learning rate; in almost all cases that we have tested, $50 \times n$ is enough to reach this point and further iterations are unnecessary.

Output neurons are in connection with their adjacent or neighboring neurons and compete to have minimum distance to an input pattern. There are ways to define neighborhood relationship and topology of the output layer, e.g., 2D, 3D, honeycomb, or grid neighboring. Output layer of SOM in this paper has grid structure as depicted in Fig. 1. In order to capture all distinct patterns exist in inputs, number of the output neurons that are not very close to each other should be more than the number of possible input patterns. Again, by examination we have found that $121 \times \min(n, 2^m)$, which depends on both m and n , is appropriate for the number of neurons in output layer, σ . Since we have n sequences, and each of which is composed of m two valued variables, number of distinct patterns is less than or equal to the minimum of n and 2^m . Thus, to have distance of 10 neurons between winning neuron of each pattern on the output layer, number of output neurons should be $(10 + 1)^2 \times \min(n, 2^m)$ or more. (Calculating network specifications from the number of input sequences and number of discovered TFBSs has made our method more practical on inputs

with various sizes.)

After training procedure, Pearson correlation coefficient is calculated between activities of output neurons when i th or j th rows of input are presented to input neurons and results reported as correlation between i th and j th input promoters. Algorithm 1 shows procedure of finding expression correlation through the scores of TFBSs found in the input promoters with more computational details.

Algorithm 1: Training SOM

INPUT: TFBSs matrix $\mathbf{T} = [t_{i,j} | 1 \leq i \leq n, 1 \leq j \leq m]$

OUTPUT: correlation matrix $\mathbf{C} = [c_{i,j} | 1 \leq i, j \leq n]$

Initialization: let the weight between input neuron $1 \leq i \leq m$ and output neuron $1 \leq j \leq \sigma$, $w_{i,j}$, be a random value in range $[-1, +1]$;

Training:

for $p \leftarrow 1$ to λ

for $r \leftarrow 1$ to n

Find Winner: Find $1 \leq v \leq \sigma$ such that

$$\sum_{i=1}^m (t_{r,i} - w_{i,v})^2 \quad (2)$$

be minimum possible value;

Update Weights: Increase $w_{i,j}$ by $\alpha \times \delta(v, j) \times (t_{r,i} - w_{i,j})$ for all $1 \leq i \leq m$ and $1 \leq j \leq \sigma$; where α is decreasing variable called the learning rate, defined as

$$\alpha \leftarrow \frac{1}{2} \left(\frac{1}{100} \right)^{\frac{p}{\lambda}}, \quad (3)$$

and function $\delta(\cdot)$, neighborhood function, is defined as

$$\delta(v, j) \leftarrow \exp \left(- \frac{(\text{row}(v) - \text{row}(j))^2 + (\text{col}(v) - \text{col}(j))^2}{2\mu^2} \right), \quad (4)$$

where functions $\text{row}(\cdot)$ and $\text{col}(\cdot)$ respectively return row and column of input index in the output layer. μ is time decreasing variable that affects neighboring magnitude

$$\mu \leftarrow 5 \left(\frac{2}{10} \right)^{\frac{p}{\lambda}}. \quad (5)$$

end for

end for

Compute Correlations: Let $c_{i,j}$ be the Pearson correlation coefficient of the output layer activity when information of p_i or p_j are presented to the input layer; specifically, let

$$c_{i,j} \leftarrow \text{Pearson} \left(\left[\sum_{k=1}^m t_{i,k} w_{k,\ell} | 1 \leq \ell \leq \sigma \right], \left[\sum_{k=1}^m t_{j,k} w_{k,\ell} | 1 \leq \ell \leq \sigma \right] \right) \quad (6)$$

for all $1 \leq i, j \leq n$.

Building correlation network In this step, the goal is to construct graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ from matrix $\mathbf{C}$. Nevertheless, in the biological networks genes are in groups with many interactions (Barabasi and Oltvai, 2004), these groups connect to other groups via special member genes with large number of connections. We are interested to find these groups.

Graph $\mathbf{G}$ has n nodes correspond to n genes, that is, $\mathbf{V} = \{v_1, v_2, \dots, v_n\}$. The edge between gene i and j , $e_{i,j}$, has initial weight of $c_{i,j}$. In the beginning, genes are placed randomly on a n by n 2d plain; then, each one slightly moves toward genes with higher mutual correlation by a small amount. To be exact, position of gene $1 \leq i \leq n$ at time $t + 1$, $v_i(t + 1)$, is computed from $v_i(t)$ by the following formula:

$$\overline{v_i(t+1)} \leftarrow \overline{v_i(t)} + \alpha(t) \left(\sum_{j=1}^n e_{i,j} \left\| \overline{v_i(t)} - \overline{v_j(t)} \right\| \overline{v_j(t)} \right) \quad (7)$$

where function $\alpha(t)$ is time – decreasing temperature function, calculated according to the rule

$$\alpha(t) \leftarrow \exp\left(-\frac{1}{t+1}\right), \quad (8)$$

and $\overline{v_i(0)}$, $1 \leq i \leq n$, are randomly scattered on a n by n 2d plain.

This procedure repeats until genes positions become fixed or temperature reaches specific value of $\alpha(t) < 0.001$. Now, genes with similar expressions are closer and it is easy to identify gene groups. We have used a method called neighbor joining to find these groups as follows.

Initially, each group consists of only one unique gene; then, two groups with minimum distance, i.e., neighbors, join to form a new group. Distance of newly formed group to others is average of distances of two joined neighbors to them. Groups join if the distance between them is smaller than a certain threshold value. Experiments revealed that this threshold value should increase to some extent by growth of n , and $\ln(n)$ works well. (The threshold value for joining groups should depend to some extent on the number of genes (n), because in the beginning genes are randomly scattered on a n by n plain and initial distances between them depends on n . We have examined several functions of n , among them $\ln(n)$ is good for this purpose.)

After finding groups, edges weights will be adjusted: weight of edges whose both vertices are in the same group increases by 25%; and, finally, edges with low weights, less than 0.25, will be removed.

RESULTS

This section presents the results of running our method on both simulated and biological datasets. Nonetheless, we need some measures to evaluate them; thus, we discuss employed measures at the outset.

Measures used to evaluate the results In order to judge the accuracy of proposed method on inferring correlation networks, a number of well – known standard statistical measurements, shown in Table 1, are employed. Measures introduced in Table 1 rely on four statistics: *true positive (TP)*, *true negative (TN)*, *false positive (FP)*, and *false negative (FN)*. TP is the number of edges in real network that predicted to be edges by the method; TN is the number of edges that does not exist in both real and constructed networks; FN is the number of edges ignored by the method; and, FP is the number of edges that the method draws incorrectly.

Simply, *sensitivity (SN)* and *specificity (SP)* quantify prediction accuracy, respectively, via true positive and true negative values. *Correlation coefficient (CC)* calculates same value by considering all four statistics. By dividing the number of true prediction by all predictions, *accuracy (ACC)*, probability of obtaining true predictions, can be calculated.

Simulated datasets We have used rMotifGen tool (Rouchka and Hardin, 2007) to generate simulated promoters datasets with respect to properties of biological promoters. Each dataset has 5 to 50 random promoters, each of which is a sequence of length 200 to 600 base pairs. Promoters are in separate groups and members of each group have common TFBSs. A TFBS is 7 to 19 base pairs long, has zero to at most one-third of its length random mutations, and standard deviation of its position in promoters is 28% compared to random positioning.

Figure 2 presents the average of SN, SP, CC, and ACC over 100 datasets: all four measures are very high when sequences have at least one TFBS. Moreover, Fig. 2 reveals that accuracy of predictions can be improved by increasing the number of common TFBSs; also it shows that if sequences have no TFBSs at all, then these four measures will dramatically reduce to about 0.20. This means that sequences are clustered according to their TFBSs and networks inferred by our method are more similar to real networks in comparison to those generated

Table 1. Formulas used to evaluate the results

Measure	Formula
SN	$\frac{TP}{TP + FN}$
SP	$\frac{TN}{TN + FP}$
CC	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP + FN)(TN + FP)(TP + FP)(TN + FN)}}$
ACC	$\frac{TP + TN}{TP + TN + FN + FP}$

Fig. 2. Average SN, SP, CC, and ACC of our method on 100 simulated datasets of sequences with 0 to TFBSs.

Fig. 3. A sample network generate by our method. Each five genes, starting from gene number 1, have their unique TFBS; therefore the perfect result is a tripartite graph. Nodes with higher correlation are closer and edge thickness indicates correlation magnitude between its two heads.

at random.

Figure 3 shows the network generated for a dataset with three sets of sequences; members of each set share a single common TFBS with one mutation, so, the result must be a tripartite graph. There are only five wrong edges in displayed graph and SN, SP, CC, and ACC respectively are 1.00, 0.94, 0.90, and 0.95.

Comparison with gene ontology resource *The Gene Ontology*TM (GO) project provides another resource to look for functional, locational, or procedural similarities within a list of genes. In brief, GO describes genes characteristics by an organized vocabulary of three categories or ontologies: molecular function, cellular component, and biological process. Ontologies are hierarchies of defined terms and each gene might be labeled with one or more terms of ontologies.

Several studies have provided us with evidences that genes that are biologically related would maintain this relation both in their expression levels as well as in their GO data (Sevilla et al., 2005). For instance, Sevilla et al. (2005) have concluded that correlation coefficient between

gene expression and GO data is about 0.70 when Resnik measure of ontology similarity were exploited to estimate genes relationship (Sevilla et al., 2005). Consequently, it is reasonable to compare networks generated by GO data with networks generated by promoter data which our method provides. To perform this comparison, we built 100 datasets, each with 20 to 40 promoter sequences that were selected randomly from yeast genome (promoter sequences were downloaded from Saccharomyces Genome Database Project (Cherry et al., 1998)). We run our method on these datasets and compare resulted network with networks generated by GO data. SN, SP and ACC of this comparison are 0.60, 0.64 and 0.61, respectively.

In another experiment, we build several datasets from yeast genes that contribute in disparate biological processes, e.g. *translation process* and *cytokinesis process*. Since genes of a dataset belong to very distinct processes, their expression should be very dissimilar (Wang et al., 2004); and, networks of these datasets should tend to be multi-partite. Running our method on these datasets resulted in networks that are very similar to multi-partite graphs, as expected; SN, SP, and ACC all are about 0.60.

Biological dataset Our biological dataset contains 500 base pairs of upstream sequences of *S. cerevisiae* (yeast) genes downloaded from YEASTRACT (Teixeira et al., 2006). Among all yeast promoters, we have chosen 254 genes which known to be regulated by 29 TFs.

Here our goal is to calculate the accuracy of the proposed method in predicting expression correlation from downloaded upstream sequences; thus, we have compared the output of our method for each pair of genes in the dataset with correlation reported by microarray technology (Lee et al., 2007). Mean square deviation of this comparison was 0.27. Comparison derived a SN of 0.51, SP of 0.56, and CC of 0.41.

It is common to evaluate the accuracy of methods that use microarray data to infer regulatory interactions by their ability to reproduce the experimental data or by their ability to identify verified interactions. For example, Sirbu et al. (2010) have compared evolutionary algorithms presented for regulatory network inference; they concluded that five studied methods, namely DE+AIC (Noman and Iba, 2006), GA+ANN (Keedwell and Narayanan, 2005), GLSDC (Kimura et al., 2003), PEACE1 (Kikuchi et al., 2003), and GA+ES (Spieth et al., 2004) have overall accuracy of 0.44 in predicting identified interactions, which is comparable with accuracy of our method: correlation coefficient value of our method is 0.41. Also, they have found that average of mean square deviation of five studied methods over five sample genes is about 0.21; as noted above, mean square deviation of our method is 0.27, which is slightly higher than this value. These results suggest that capability of our method for approximating the microarray data of yeast is

Fig. 4. Networks generated by two data sources: left by microarray data, right by data that our method provides. Expression patterns of genes DAL2, DAL4, and DAL7 are very different from others, so they are positioned at distant relative to others in both networks. Genes with high expression correlation are close to each other and weak correlation coefficients ($< 50\%$) are depicted by dotted lines.

as good as other five methods.

Figure 4 shows two networks that are generated from microarray data (left) and from the data that are provided by our method (right) for eight yeast genes: HSC82, HSP26, SIS1, SSA1, and SSA4 that are regulated by the heat shock factor protein (HSF), and DAL2, DAL4, and DAL7, which have a TFBS known as the upstream induction sequence (UIS). Furthermore, according to the GOTM database the first five genes are involved in the *protein folding process* and the rest contribute to the *heterocycle metabolic process* (Ashburner et al., 2000); further exploration using the AmiGO[®] browser (“AmiGO: official online tool-set for searching GO at <http://amigo.geneontology.org>,”) has revealed that these two processes are separated just in the second level from the root of the GO biological processes hierarchy. Thus, expression pattern of genes that have *protein folding process* label should be different from those that share *heterocycle metabolic process* label (Wang et al., 2004). Genes DAL2, DAL4, and DAL7 have strong expression correlation with each other and correctly are positioned far from other genes in both networks, whereas according to microarray data there is no similarity between expression patterns of DAL4 and HSP26 (Lee et al., 2007) and our method incorrectly identified slightly strong correlation between them (Fig. 4, right). Also predicted similarities between DAL2 and SIS1 or SSA4 are incorrect. Sensitivity, specificity, correlation coefficient, and accuracy of this example are 0.89, 0.94, 0.53, and 0.75, respectively.

CONCLUSION

This paper presented a new method for inferring gene correlation networks. The input of this method is a set of genes upstream sequences or promoters, each with

arbitrary length, and the output is correlation network of given genes.

Briefly, the method begins by extracting TFBSs of input promoters, altering their scores according to the standard deviation of positions where they occurred; and, merging similar ones into single one. Finding expression correlation between each pair of genes from these scores by a self – organizing neural network is the next step. In the final step, it generates the network graph by moving genes with similar expression patterns towards each other and joining them into genes groups.

This method has been tested on simulated and biological data. Results show that the accuracy and efficiency of this method in inferring correlation networks, merely, from promoters is almost equivalent to the accuracy of methods that use microarray data as input, assessments of some of which have been presented by Sirbu et al. (2010) and Allen et al. (2012); and to the accuracy of methods that start with clustered promoters, as presented by Beer and Tavazoie (2004).

Accuracy of the proposed method is a function of the quality of its input, which is the result of a TFBSs discovery method; therefore, improving those methods will improve the proposed method as well.

Collecting the results of two or more TFBSs discovery methods on the given upstream sequences and presenting those TFBSs that are reported by many of them to the network, seems to be another promising way to improve accuracy. Also studies have found associations between histone acetylation patterns and gene activity (Kurdistani et al., 2004), which in turn could prove helpful to our method.

When biological data are used instead of simulated data, the accuracy of proposed method drops by 0.30 in terms of CC; the reason for this significant drop is that there are more determining factors in regulation of genes

than presence or absence of TFBSs; factors like those stated in the above paragraph or epigenetic regulation (van Driel et al., 2003).

Combining data provided by this method with other existing data, e.g. GO data and microarray data and testing above mentioned possibilities is our planning for future works.

REFERENCES

- Akutsu, T., Miyano, S., and Kuhara, S. (1999) Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symp. Biocomputing* **4**, 17–28.
- Allen, J. D., Xie, Y., Chen, M., Girard, L., and Xiao, G. (2012) Comparing statistical methods for constructing large scale gene networks. *PLoS ONE* **7**, e29348.
- Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000) Gene ontology: tool for the unification of biology. *Nat. Genet.* **25**, 25–29.
- Bailey, T. L., and Elkan, C. (1994) Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proc. 2nd Int. Conf. Intell. Syst. Mol. Biol.* **2**, 127–138.
- Bailey, T. L., and Elkan, C. (1995) Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Mach. Learn.* **21**, 51–80.
- Bar-Joseph, Z., Gerber, G. K., Lee, T. I., Rinaldi, N. J., Yoo, J. Y., Robert, F., Gordon, D. B., Fraenkel, E., Jaakkola, T. S., Young, R. A., and Gifford, D. K. (2003) Computational discovery of gene modules and regulatory networks. *Nat. Biotechnol.* **21**, 1337–1342.
- Barabasi, A. L., and Oltvai, Z. N. (2004) Network biology: Understanding the cell's functional organization. *Nat. Rev. Genet.* **5**, 101–113.
- Beer, M., and Tavazoie, S. (2004) Predicting gene expression from sequence. *Cell* **117**, 185–198.
- Brazhnik, P., De la Fuente, A., and Mendes, P. (2002) Gene networks: how to put the function in genomics. *Trends Biotechnol.* **20**, 467–472.
- Butte, A. J., and Kohane, I. S. (2000) Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pacific Symp. Biocomputing* **5**, 418–429.
- Chen, T., He, H. L., and Church, G. M. (1999) Modeling gene expression with differential equations. *Pacific Symp. Biocomputing* **4**, 29–40.
- Cherry, J. M., Adler, C., Ball, C. A., Chervitz, S. A., Dwight, S. S., Hester, E. T., Jia, Y., Juvik, G., Roe, T., Schroeder, M., Weng, S., and Botstein, D. (1998) SGD: *Saccharomyces* Genome Database. *Nucleic Acids Res.* **26**, 73–79.
- Chua, G., Robinson, M. D., Morris, Q., and Hughes, T. R. (2004) Transcriptional networks: reverse-engineering gene regulation on a global scale. *Curr. opin. microbiol.* **7**, 638–646.
- D'Haeseleer, P., Liang, S., and Somogyi, R. (2000) Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics* **16**, 707–726.
- Das, M., and Dai, H. K. (2007) A survey of DNA motif finding algorithms. *BMC Bioinformatics* **8**(Suppl. 7), S21.
- Davidson, E. H. (2001) *Genomic Regulatory Systems: Development and Evolution*. Academic Press, San Diego.
- Davidson, E. H., McClay, D. R., and Hood, L. (2003) Regulatory gene networks and the properties of the developmental process. *Proc. Natl. Acad. Sci. USA* **100**, 1475–1480.
- DeJong, H. (2002) Modeling and simulation of genetic regulatory systems: a literature review. *Comput. Biol.* **9**, 67–103.
- Filkov, V. (2005) Identifying gene regulatory networks from gene expression data. In: *Handbook of Computational Molecular Biology* (ed.: S. Aluru), pp. 27.21–27.29. Chapman and Hall/CRC Press Boca Raton.
- Filkov, V., Skiena, S., and Zhi, J. (2002) Analysis techniques for microarray time-series data. *Comput. Biol.* **9**, 317–330.
- Friedman, N., Linial, M., Nachman, I., and Pe'er, D. (2000) Using Bayesian networks to analyze expression data. *Comput. Biol.* **7**, 601–620.
- Gardner, T. S., and Faith, J. J. (2005) Reverse-engineering transcription control networks. *Phys. Life Rev.* **2**, 65–88.
- Gebert, J., Radde, N., and Weber, W. G. (2007) Modeling gene regulatory networks with piecewise linear differential equations. *Eur. Oper. Res.* **181**, 1148–1165.
- GuhaThakurta, E. (2006) Computational identification of transcriptional regulatory elements in DNA sequence. *Nucleic Acids Res.* **34**, 3585–3598.
- Hecker, M., Lambeck, S., Toepfer, S., van Someren, E., and Guthke, R. (2009) Gene regulatory network inference: Data integration in dynamic models - A review. *BioSystems* **96**, 86–103.
- Hertz, G. Z., and Stormo, G. D. (1999) Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics* **15**, 563–577.
- Husmeier, D. (2003) Reverse engineering of genetic networks with Bayesian networks. *Biochemical Society Transactions* **31**, 1516–1518.
- Huynh-Thu, V. A., Irrthum, A., Wehenkel, L., and Geurts, P. (2010) Inferring regulatory networks from expression data using tree-based methods. *PLoS ONE* **5**, e12776.
- Jacob, F., and Monod, J. (1961) Genetic regulatory mechanisms in the synthesis of proteins. *Mol. Biol.* **3**, 318–356.
- Kabir, M., Noman, N., and Iba, H. (2010) Reverse engineering gene regulatory network from microarray data using linear time-variant model. *BMC Bioinformatics* **11**(Suppl. 1), S56.
- Kauffman, S. (1969) Homeostasis and differentiation in random genetic control networks. *Nature* **224**, 177–178.
- Keedwell, E., and Narayanan, A. (2005) Discovering gene networks with a neural-genetic hybrid. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2**, 231–242.
- Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K., and Tomita, M. (2003) Dynamic modeling of genetic networks using genetic algorithm and S-system. *Bioinformatics* **19**, 643–650.
- Kimura, S., Hatakeyama, M., and Konagaya, A. (2003) Inference of S-system models of genetic networks using a genetic local search. *Evolutionary Computation*, 2003 **1**, 631–638.
- Kohonen, T. (1989) *Self-organization and Associative Memory*. Springer-Verlag, New York.
- Kurdistan, S. K., Tavazoie, S., and Grunstein, M. (2004) Mapping global histone acetylation patterns to gene expression. *Cell* **117**, 721–733.
- Latchman, D. S. (1998) *Gene Regulation: A Eukaryotic Perspective* (3rd Ed.). Nelson Thornes Publishers, Ltd, Cheltenham, UK, pp 352.
- Lee, I., Li, Z., and Marcotte, E. M. (2007) An improved, bias-reduced probabilistic functional gene network of baker's yeast, *Saccharomyces cerevisiae*. *PLoS ONE* **2**, e988.
- Lewin, A., Richardson, S., Marshall, C., Glazier, A., and Aitman, T. (2006) Bayesian modeling of differential gene expression. *Biometrics* **62**, 1–9.

- Lewin, B. (2004) Genes VIII. Pearson Prentice Hall, NJ, USA.
- Liu, X. S., Brutlag, D. L., and Liu, J. S. (2001) BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. *Pacific Symp. Biocomputing* **6**, 127–138.
- Lodish, H., Berk, A., Kaiser, C. A., Krieger, M., Scott, M. P., Bretscher, A., Ploegh, H., and Matsudaira, P. (2007) Molecular Cell Biology. W. H. Freeman, New York, USA.
- MacIsaac, K. D., and Fraenkel, E. (2006) Practical strategies for discovering regulatory DNA sequence motifs. *PLoS Comp. Biol.* **2**, 201–210.
- Mahdevar, G., Sadeghi, M., and Nowzari-Dalini, A. (2012) Transcription factor binding sites detection by using alignment-based approach. *Theor. Biol.* **304**, 96–102.
- Margolin, A., Nemenman, I., Basso, K., Wiggins, C., Stolovitzky, G., Favera, R., and Califano, A. (2006) ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* **7**(Suppl. 1), S7.
- Martin, S., Zhang, Z., Martino, A., and Faulon, J. L. (2007) Boolean dynamics of genetic regulatory networks inferred from microarray time series data. *Bioinformatics* **23**, 866–874.
- Mehra, S., Hu, W., and Karypis, G. (2004) A Boolean algorithm for reconstructing the structure of regulatory networks. *Metab. Eng.* **6**, 326–339.
- Meissner, M., Koch, O., Klebe, G., and Schneider, G. (2009) Prediction of turn types in protein structure by machine-learning classifiers. *Proteins* **74**, 344–352.
- Meyer, P. E., Kontos, K., Lafitte, F., and Bontempi, G. (2007) Information-theoretic inference of large transcriptional regulatory networks. *EURASIP Journal on Bioinformatics & Systems Biology* 2007, DOI: 10.1158/2007/79879.
- Noman, N., and Iba, H. (2006) Inference of genetic networks using S-system: information criteria for model selection. *Proc. 8th Genet. Evol. Comput. Conf.* New York, NY, USA, 263–270.
- Pavesi, G., and Valentini, G. (2009) Classification of co-expressed genes from DNA regulatory regions. *Inf. Fusion* **10**, 233–241.
- Perrin, B., Ralaivola, L., Mazurie, A., Bottani, S., Mallet, J., and d’Alche’Buc, F. (2003) Gene networks inference using dynamic Bayesian networks. *Bioinformatics* **19**(Suppl. 2), ii138–ii148.
- Roth, F. P., Hughes, J. D., Estep, P. W., and Church, G. M. (1998) Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. *Nat. Biotechnol.* **16**, 939–945.
- Rouchka, E. C., and Hardin, C. T. (2007) rMotifGen: random motif generator for DNA and protein sequences. *BMC Bioinformatics* **8**, 292.
- Salgado, H., Gama-Castro, S., Martinez-Antonio, A., Diaz-Peredo, E., Sanchez-Solano, F., Peralta-Gil, M., Garcia-Alonso, D., Jimenez-Jacinto, V., Santos-Zavaleta, A., Bonavides-Martinez, C., and Collado-Vides, J. (2004) RegulonDB (version 4.0): transcriptional regulation, operon organization and growth conditions in *Escherichia coli* K-12. *Nucleic Acids Res.* **32**(Suppl. 1), D303–D306.
- Schlitt, T., and Brazma, A. (2007) Current approaches to gene regulatory network modelling. *BMC Bioinformatics* **8**(Suppl. 6), S9.
- Sevilla, J. L., Segura, V., Podhorski, A., Guruceaga, E., Mato, J. M., Martinez-Cruz, L. A., Corrales, F. J., and Rubio, A. (2005) Correlation between gene expression and GO semantic similarity. *IEEE/ACM Trans. Comput. Biol. Bioinform* **2**, 330–338.
- Sirbu, A., Ruskin, H. J., and Crane, M. (2010) Comparison of evolutionary algorithms in gene regulatory network model inference. *BMC Bioinformatics* **11**, 59.
- Spieth, C., Streichert, F., and Zell, N. S. A. (2004) Optimizing topology and parameters of gene regulatory network models from time-series experiments. *Genetic and Evolutionary Computation - GECCO 2004*, 461–470.
- Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. (2003) A gene-coexpression network for global discovery of conserved genetic modules. *Science* **302**, 249–255.
- Teixeira, M. C., Monteiro, P., and Jain, P. (2006) The YEASTRACT database: a tool for the analysis of transcription regulatory associations in *Saccharomyces cerevisiae*. *Nucleic Acids Res.* **34**, D446–D451.
- Thijs, G., Marchal, K., Lescot, M., Rombauts, S., De Moor, B., Rouzé, P., and Moreau, Y. (2002) A Gibbs sampling method to detect overrepresented motifs in the upstream regions of coexpressed genes. *Comput. Biol.* **9**, 447–464.
- Tian, T., and Burrage, K. (2006) Stochastic models for regulatory networks of the genetic toggle switch. *Proc. Natl. Acad. Sci. USA* **103**, 8372–8377.
- Tomba, M., Li, N., Bailey, T. L., Church, G. M., De Moor, B., Esmin, E., Favorov, A. V., Frith, M. C., Fu, Y., Kent, J. W., et al. (2005) Assessing computational tools for the discovery of transcription factor binding sites. *Nat. Biotechnol.* **23**, 137–144.
- van Driel, R., Fransz, P. F., and Verschure, P. J. (2003) The eukaryotic genome: a system regulated at different hierarchical levels. *Cell Sci.* **116**, 4067–4075.
- Vohradsky, J. (2001) Neural network model of gene expression. *FASEB J.* **15**, 846–854.
- Wang, H., Azuaje, F., Member, Bodenreider, O., and Dopazo, J. (2004) Gene expression correlation and gene ontology-based similarity: An assessment of quantitative relationships. *Proc. 2004 IEEE Symp. Comput. Intell. Bioinform. Comput. Biol.* 25–31.
- Yip, K. Y., Alexander, R. P., Yan, K.-K., and Gerstein, M. (2010) Improved reconstruction of *In Silico* gene regulatory networks by integrating knockout and perturbation data. *PLoS ONE* **5**, e8121.
- Yu, H., Luscombe, N., Qian, J., and Gerstein, M. (2003) Genomic analysis of gene expression relationships in transcriptional regulatory networks. *Trends Genet.* **19**, 422–427.
- Zhu, J., and Zhang, M. Q. (1999) SCPD: a promoter database of the yeast *Saccharomyces cerevisiae*. *Bioinformatics* **15**, 607–611.