Expert Systems with Applications 37 (2010) 2202–2213

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

An expert system to identify co-regulated gene groups from time-lagged gene clusters using cell cycle expression data

Li-Ching Wu [a], Jhih-Long Huang [b], Jorng-Tzong Horng [b,d,*], Hsien-Da Huang [c]

[a] Institute of System Biology and Bioinformatics, National Central University, Taiwan
[b] Department of Computer Science and Information Engineering, National Central University, Taiwan
[c] Institute of Bioinformatics, National Chiao-Tung University, Taiwan
[d] Department of Bioinformatics, Asia University, Taiwan

ARTICLE INFO

Keywords:
Expert system
Data mining
Gene expression
Bioinformatics

ABSTRACT

Motivation: The analysis of time series gene expression data can provide us with the opportunity to find co-regulated genes that show a similar expression patterns under a contiguous subset of experimental conditions. However, these co-regulated genes may behave almost independently under other conditions. Furthermore, the similarity in the expression pattern might be time-shifted. In that case, we need to be concerned with grouping genes that share similar expression patterns under a contiguous subset of conditions and where the similarity in expression pattern might have time lags. In addition, to be considered a time-shifted similar pattern, because co-regulated genes in a biological process may show a periodic pattern in their cell cycle expression, we also should group genes with periodic similar patterns over multiple cell cycles. If this is carried out, we can regard such grouped genes as cell-cycle regulated genes. *Results:* We propose a method that follows the *q*-cluster concept [Ji, L., & Tan, K. L. (2005). Identifying time-lagged gene clusters using gene expression data. *Bioinformatics, 21*(4), 509–516] and further advances this approach towards the identification of cell-cycle regulated genes using cell cycle microarray data. We used our method to cluster a microarray time series of yeast genes to assess the statistically biological significance of the obtained clusters we used the *p*-value obtained from the hypergeometric distribution. We found that several clusters provided findings suggesting a TF–target relationship. In order to test whether our method could group related genes that other methods have found difficult to group, we compared our method with other measures such as Spearman Rank Correlation and Pearson Correlation. The results of the comparison demonstrate that our method indeed could group known related genes that these measures regard as having only a weak association.

1. Introduction

DNA microarray technology enables the simultaneous study of gene expression levels on a large scale. Expression level is the logarithm of the abundance of the mRNA of a gene under a specific condition. The gene expression data of a microarray is arranged as a data matrix. Each gene corresponds to one row and each condition to one column. Each element of this matrix represents the expression level of a gene under a specific condition. The conditions of a microarray may be different time points, different environmental conditions or different organs. The analysis of microarray data has facilitated the study of genetic regulatory networks. The correlation patterns of genes with experimental conditions can be used to identify the networks that are comprised of correlated genes and thus how the correlated genes interact with each other.

Clustering methods can be applied to either the genes or the conditions of the microarray matrix separately. However, some problems occur when applying clustering to the analysis of gene expression data. A set of genes may simultaneously activate a particular biological process over certain contiguous conditions but behave independently under other condition. Therefore, we need to group genes that have similar behavior under a specific subset of the conditions. Clustering can not satisfy this requirement. The biclustering method is a technique that makes this possible and allows the grouping of genes and conditions simultaneously within a data matrix. The goal of biclustering is to find a bicluster that is a subset of genes that show similar behavior under a specific subset of the conditions (Cheng & Church, 2000). Thus, genes in the same bicluster are co-expressed and further are likely to be co-regulated.

In the analysis of time-series expression data, a set of genes may activates a particular biological process over a certain contiguous

[*] Corresponding author. Address: Department of Bioinformatics, Asia University, Taiwan.
E-mail address: horng@db.csie.ncu.edu.tw (J.-T. Horng).

0957-4174/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2009.07.053

set of conditions instead under a discrete condition. In such a case, we should find biclusters for a contiguous subset of conditions. However, in fact, co-expressed genes do not regulate each other simultaneously but only after a certain time lag. That is to say, there is a transcriptional time lag whereby the regulator gene takes time to express its protein product and a further delay occurs as the target gene responds to the regulator protein. Hence, because of the transcriptional time lag of co-regulated genes, we need to take time-lagged co-regulated genes into consideration when forming biclusters. In addition, when considering time-lagged similarity of expression patterns between genes, it is necessary to consider biclusters with coherent values for both the rows and columns of the expression matrix. This is because their expression relies on a promoter that is a structural regulatory sequence recognized by a TF of the RNA polymerase holoenzyme. The reason that co-expressed genes share a common sequence within their promoter will therefore result in shared expression. However, the recognition efficiency of this TF is not the same for every gene having the same promoter. This condition leads to biclusters having a variety of coherence values. There are two kinds of coherence values for a bicluster. The first is a shifted similarity pattern that can be viewed as based on an additive model. The second is a scaled similarity pattern that can be viewed as based on a multiplicative model. In a mathematical sense, scaling and vertical shifting of the expression level can be referred to as linear transformations. Consider two time-series x and y. In this case y is a linear transformation of x if it can be expressed as $y = mx + b$.

Many biological processes show periodic pattern such as the cell cycle process, therefore it is useful to find periodically regulated genes with similar periodic patterns of expression. We can then use these cell-cycle regulated genes to map the transcriptional regulatory network that controls the cell cycle.

A suffix tree is a data structure that contains all suffixes of a string s. It has been widely used for string matching and exact sequence comparison (Ukkonen, 1995). This approach was used to develop an algorithm for building a suffix tree that runs in time $O(n)$. Once a suffix tree is built, most problems can be solved in linear-time using it. The suffix tree built for a set of strings is called a generalized suffix tree (Gusfield, 1997). In order to avoid creating empty suffixes, we usually append to s an extra character $ before the building of the suffix tree. The key feature of the generalized suffix tree is that any leaves in this tree contain two pieces of information: The first is the string number and the second is the starting position of a suffix that makes up this string.

2. Related work

A great many approaches have been developed for the identification of co-regulated genes from microarrays. The correlation method is one that determines whether two variables have a strong global association, but this approach does not take time lag issue into consideration. However, another correlation method, the Cross-Correlation Method (Kato et al., 2001), is different from the traditional Pearson Correlation approach. It takes into account time-lagged co-regulations when testing the expression levels of two genes and identifies if there is a significant linear relationship. However, this approach still only determines whether two genes have strong global similarity but does not determine local similarity. Yet another approach is the edge detection method (Filkov et al., 2001), which scans through each gene's expression curve to find where big changes in expression level (edges) occur, and then sums up the number of edges making up the expression curves of the two genes that have the same direction in order to generate a similarity score. Gene pairs are likely to have a positive-regulated relationship and if this is true, they will be given a high score. On the other hand, edges that are farther apart are given a lower score. Although the edge method strongly focuses on local similarity between two gene expression curves, there are always too few edges that match between gene pairs and this gives rise to relatively low similarity scores. Yet another approach is the Event Method (Kwon et al., 2003). This first transforms the directional changes in expression level into a directional event (Rising, Constant or Falling) by calculating the slope values at each time point; these are then converted into events. When the transformation process is complete, global sequence alignment by the Needleman–Wunsch algorithm is used to find the best match between the two event strings taking noise and time lag into account. Gene pairs that have an activation relationship are likely to have a high similarity score for the event strings. Although the event method is efficient at finding gene TF–target relationships, it is unable to produce a high score for related genes when the event strings are similar for periodical short matches because the method uses global alignment. Moreover, it is computationally inefficient because it needs n^2 pairwise global alignments if there are n genes. There is one further limitation: this is that it provides putative TF–target relationships for users by using gene pairs instead of gene clusters. It is clear that clusters are more efficient than pairs when finding TF–target relationships. The next available approach is CLARITY (Balasubramaniyan et al., 2005). This can find locally similar regions in gene expression profiles by measuring similarity based on Spearman rank correlation. In order to discovering local time-shifted relationships between two profiles, the program enumerates all possible alignments in a systematic way. In order to reduce the complexity of computation for $O(n^2)$, it use an approximate algorithm. Although CLARITY tests similarity between genes by measuring shape (the qualitative behavior) of the expression profiles, which is very useful, there are a few limitations to this program. One exception is where cell cycle co-expressed genes are highly related to the cell cycle but their qualitative behavior different over multiple cycles. The result is a horizontal shift and scale problem with the expression profiles and CLARITY finds it difficult to identify such genes as having a strong association. A further approach is q-cluster (Ji & Tan, 2005). Here, the profile is transformed into three type of change denoted by 1, -1 and 0. In this method, the pattern of a q-cluster is indicated as an event string of length $(q-1)$ to show the changes that occur and this reflects how expression level changes from condition i to condition $i+1$ under the q conditions. As the data is transformed into three distinct classes, there are 3^{q-1} q-clusters in total and each q-cluster has a unique q-value, where $0 \leqslant q\text{-value} \leqslant 3^{q-1}$ (Ji & Tan, 2005). Although q-cluster provides users with detailed information that allows the detection of periodically co-regulated genes, users must search for these genes themselves. That is to say, q-cluster does not further group genes together according the periodic similar patterns relative to the cell cycle expression data.

Spellman et al. (1998) first performed a genome-wide transcriptional analysis of the mitotic cell cycle of *Saccharomyces cerevisiae* using microarrays and found about 800 periodically expressed genes that peak each cycle. We can regard these genes as cell-cycle regulated genes. In another similar investigation (Whitfield et al., 2002) it proved possible to identify genes periodically expressed over the human cell cycle using microarray analysis. Although a large number of studies (Cho et al., 1998; Spellman et al., 1998) have revealed over 800 genes that are cell-cycle regulated in yeast, there has been a lacking of a good method to further group these periodically expressed genes into clusters by biclustering in order to get an understanding of transcription regulatory networks within cell cycle.

When biclustering (Cheng & Church, 2000) was first applied to gene expression data, it used the measure, mean squared residue, to find the biclusters. In later years, many approaches to biclustering of expression data have been proposed. However, if these approaches focus on finding exclusive biclusters in a time series of expression data, there are a number of problems that are of concern. A major problem is that they ignore many additional relationships across the time series such as time-lagged relationships between transcription factors and thus genes that are activated by this transcription factor. Another problem is that the process groups genes that show similar activity patterns under a subset of discrete conditions instead of under contiguous conditions. In the analysis of time series dataset, it is reasonable to restrict the analysis to biclusters with contiguous columns to reduce the time complexity of computation.

3. Methods

In order to take into account the bicluster's coherent values when biclustering, we firstly transform the expression data into event strings. We then use the set of event strings to construct a generalized tree. When the generalized tree has been built, we can use it to form biclusters quickly while taking time lags into consideration. Furthermore, during the analysis of cell cycle expression data, we need to focus on finding cell-cycle regulated genes. Therefore, we further transform the event strings into transactions representing what event substrings (items) are included in genes' event strings (transactions). Next, we use Apriori's concept to obtain the set of similar patterns (items) that occur simultaneously with the same genes (transactions). Finally, we use the positions at which the similar patterns occur for the same genes to further group the genes taking periodically similar patterns and time lags into consideration.

3.1. Phase 1: Transforming expression data into event strings

When we have n genes and m conditions in the form of time-series expression data, the time-series expression data can be represented as a $A = n \times m$ matrix, where $A_{i,j}$ represents the expression level of gene i under condition j. First, Matrix A is converted into a $A' = n \times (m-1)$ matrix such that

$$A'_{i,j} = \begin{cases} \frac{A_{i,j+1} - A_{i,j}}{|A_{i,j}|} & \text{if } A_{i,j} \neq 0, \\ \mathit{Infinity} & \text{if } A_{i,j} = 0 \text{ and } A_{i,j+1} > 0, \\ -\mathit{Infinity} & \text{if } A_{i,j} = 0 \text{ and } A_{i,j+1} < 0, \\ 0 & \text{if } A_{i,j} = 0 \text{ and } A_{i,j+1} = 0. \end{cases}$$

to reflect the change of each gene expression level between two neighboring time points. The transformation function we used comes from q-cluster. Each entry $A'_{i,j}$ reveals the directional change (the rate of change across time) from the expression level of gene i at time point j to the expression level of gene i at time point $j+1$. After A has transformed into A' matrix, we are interested in how the change in the gene expression level can be converted into a set of symbols, Σ. Σ contains five symbols, $\{\mathrm{D}, \mathrm{U}, \mathrm{N}, \mathrm{H}, \mathrm{L}\}$ meaning down-regulated, up-regulated, unchanged, unchanged with high expression, and unchanged low expression, respectively. We use a threshold t (same with q-cluster) to transform the changes into event symbols such that

$$A''_{i,j} = \begin{cases} \mathrm{U} & \text{if } A'_{i,j} \geqslant t, \\ \mathrm{D} & \text{if } A'_{i,j} \leqslant -t, \\ \mathrm{N} & \text{otherwise}. \end{cases}$$

Further, we also use a threshold s to transform the symbol, N into H or L to describe what happens with the expression profiles in more depth. For every sub-matrix $a'' = 1 \times k$ of A'' whose starting row is i and starting column is j, where $i \in 1 \ldots n$, $j \in 1 \ldots m-1$, $j + k \leqslant m-1$, and

$$\begin{cases} \text{if } a''_{i,j-1} = \mathrm{U}, a''_{i,j} = \mathrm{N}, A_{i,j} > s, a''_{i,j+1} = \mathrm{N}, A_{i,j+1} > s, \ldots, a''_{i,j+k} = \mathrm{N}, A_{i,j+k} > s, A_{i,j+k+1} > s, a''_{i,j+k+1} = \mathrm{D}, \\ \text{then } a''_{i,j} = \mathrm{H}, a''_{i,j+1} = \mathrm{H}, \ldots, a''_{i,j+k} = \mathrm{H}, \\ \text{if } a''_{i,j-1} = \mathrm{D}, a''_{i,j} = \mathrm{N}, A_{i,j} < -s, a''_{i,j+1} = \mathrm{N}, A_{i,j+1} < -s, \ldots, a''_{i,j+k} = \mathrm{N}, A_{i,j+k} < -s, A_{i,j+k+1} < -s, a''_{i,j+k+1} = \mathrm{U}, \\ \text{then } a''_{i,j} = \mathrm{L}, a''_{i,j+1} = \mathrm{L}, \ldots, a''_{i,j+k} = \mathrm{L}. \end{cases}$$

After the conversion phase, matrix A' is transformed into matrix A'' and $A''_{i,j}$ represents the converted symbol of the change from the expression level of gene $i$at time point j to the expression level of gene i at time point $j+1$. As an example, Tables 1–3 show the process of converting the gene expression levels. The original matrix A is shown in Table 1. In Table 2, the matrix $A\prime$ reflects the changes in

Table 1
The original expression matrix.

	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}
G1	−0.42	−0.01	0.19	−0.09	−0.10	−0.68	−0.16	0.33	0.29	0.11	0.35	0.21
G2	−0.43	0.15	0.3	0.05	0.07	−0.02	0.33	0.26	−0.41	0	−0.4	0.03
G3	−0.11	0.27	0.38	0.46	−0.06	−0.29	−0.49	−0.11	0.03	−0.16	0.43	−0.03
G4	−0.27	−0.13	0.2	0.17	0.09	0.14	−0.34	0.1	0.23	−0.34	0.13	0.19
C5	−0.25	−0.28	−0.35	0.82	0.58	0.02	−0.51	−0.22	−0.4	−0.22	1.07	0.11

Table 2
The matrix showing the changes from the original matrix.

	$C_{1,2}$	$C_{2,3}$	$C_{3,4}$	$C_{4,5}$	$C_{5,6}$	$C_{6,7}$	$C_{7,8}$	$C_{8,9}$	$C_{9,10}$	$C_{10,11}$	$C_{11,12}$
G1	0.98	20	−1.47	−0.11	−5.8	0.76	3.06	−0.12	−0.62	2.18	−0.4
G2	1.35	1	−0.83	0.4	−1.29	17.5	−0.21	−2.58	1	$-\infty$	1.08
G3	3.45	0.41	0.21	−1.13	−3.83	−0.69	0.78	1.27	−6.33	3.69	−1.07
G4	0.52	2.54	−0.15	−0.47	0.56	−3.43	1.29	1.3	−2.48	1.38	0.46
G5	−0.12	−0.25	3.34	−0.29	−0.96	−26.5	0.57	−0.82	0.45	5.86	−0.9

Table 3
The result of conversion of the original matrix. Some event symbols N are have transformed into H to describe what is happening within the gene profiles in more depth.

	$C_{1,2}$	$C_{2,3}$	$C_{3,4}$	$C_{4,5}$	$C_{5,6}$	$C_{6,7}$	$C_{7,8}$	$C_{8,9}$	$C_{9,10}$	$C_{10,11}$	$C_{11,12}$
G1	U	U	D	N	D	U	U	N → H	D	U	D
G2	U	U	D	U	D	U	N → H	D	U	D	U
G3	U	U	N → H	D	D	D	U	U	D	U	D
G4	U	U	N → H	D	U	D	U	U	D	U	U
G5	N	N	U	N → H	D	D	U	D	U	U	D

gene expression levels on conversion from matrixA. Finally, the result of conversion process ($t = 0.3, s = 0.1$) forms matrix A'', which is shown in Table 3. In the latter case, some symbols N have now been transformed into H, because, despite the expression not varying greatly, the expression is still high and therefore important in itself.

3.2. Phase 2: Using converted matrix to construct the generalized tree

Let the set of event strings $S = \{S_1, S_2, \ldots, S_n\}$ be obtained from each row in converted matrix A'', which comes from the first phase. Each of these strings has $m - 1$ event symbols and corresponds to the symbols in a row of the converted matrix. To build a suffix tree, add a special end of string character,, to each event string. Then, use the set of event strings S to construct the generalized tree T. After the generalized suffix tree T has been built, we can use it to find the common subsequences shared among more than two sequences. Based on such common subsequences, the similar sequences (event strings) are grouped together. Use of a generalized suffix tree allows a pattern of length n to be found in $k$strings in $O(n + k)$ time. This is known as Exact String Matching. We can use Exact String Matching to find co-regulated genes in a similar way to q-cluster. As shown in Table 4, genes which have an arbitrarily similar pattern of length 3 can be grouped together. Like q-cluster, users could then choose an interesting pattern that is repeated to pinpoint TF–target relationships. For example, G3 and G5 have a similar pattern, UUD, and G5 activates this pattern after G3 by two time lags. A TF–target relationship is likely between G3 and G5. So far, our method is able to achieve the same objectives as q-cluster, but our approach gives more. If users want to see other lengths among similar patters, users just query through the generalized tree T that has been built by transforming the event strings. This is not true for q-cluster and the q-clusters need to be calculated again. Furthermore, although q-cluster provides users with detailed information that helps the detection of periodically co-regulated genes, users still have to search manually. That is to say, q-cluster does not further group genes together according to the periodic similar patterns for cell cycle expression data. In our method, we take into account periodically similar patterns to group periodically co-regulated genes as can be seen in the next phase.

3.3. Phase 3: Identifying similar patterns (items) that occur simultaneously among genes (transactions)

Let $T = \{T_1, T_2 \ldots T_n\}$ be the set of transaction, where n is the number of genes in expression data. A transaction can be thought of as a basket (gene) that contains a set of items (event substrings of a fixed length), which are bought together. A generalized suffix tree allows us to quickly find common subsequences that are shared among gene event strings; therefore we can use it to built transactions representing the event substrings that are included in gene event strings. When we have a set of transactions, T, we can use these to find the frequent h-itemsets (h similar patterns), and count the number of times they appear simultaneously in transactions (genes). The number of times that a transaction is found is defined by us as the support S, which is different to the Apriori approach (Agrawal and Srikant, 1994). There are several steps involved in finding frequent h-itemsets that are similar to the Apriori algorithm, but there are also some differences. First, find all frequent 1-itemsets. Then, do the next steps for each iteration until the frequent h-itemsets are found. The candidate k-itemsets are next generated by merging frequent $(k - 1)$-itemsets. As the next step, prune the candidate k-itemsets whose subsets are infrequent. Take the candidate k-itemsets that survive the pruning process and count the number of times they appear simultaneously among transactions. If the k items (similar patterns) among every k-itemsets do not overlap (similar pattern positions should not overlap each other), then we count this as one. Furthermore, we can limit the distance of items' starting positions to be larger than $d$and then count this as one. Finally, extract the frequent k-itemsets, a subset of candidate k-itemsets, with counts that are no less than the support S. For example, if we use the transactions shown in Table 3, they can be transformed into frequent 2-itemsets and the result is shown in Table 5. It should be noted that in our method, the items in every itemsets are not limited to being different from each other. The reason that this is done is so we can take into account instances when the periodic similar patterns among genes are the same. If G2 and G4 are examined, it will be found that they have the periodic similar pattern, UDU, which occur at position [3,8] in G2 and at position [4,7] in G4. Although we have collected the transactions (genes) that have period similar patterns, we still need to further group these genes according to the periodically similar patterns' positions to find related transactions; this is carried out in the next phase. If the number of grouped genes is less than threshold S, it is clear that the number of grouped genes can never greater than the threshold S. Thus, we can regard the threshold S as

Table 4
Grouping genes that have similar activation patterns.

Pattern	Gene [start position]
UUD	G1[0], G2[0], G3[6], G4[6], G5[8]
UHD	G1[6], G2[5], G3[1], G4[1], G5[2]
UDU	G2[1], G2[3], G2[8], G3[7], G4[4], G4[7], G5[6]
…	…
DDD	G3[3]
DDU	G3[4], G5[4]
DND	G1[2]

Table 5
The candidate 2-itemsets after pruning. G2[5,3] and G5[2,4] can not be counted as one because their positions overlap. The items (similar patterns) in the itemset are not limited to be different from each other, see itemset, UDU, UDU. The itemsets {UUD,UUD}, {UHD,UHD} and {UHD,DDU} have a support of less than 2 and therefore the final frequent 2-itemsets are {UHD,DDD}, {UHD,UDU}, {UDU,UDU}, etc.

Patterns	Gene [start positions]	Support
UUD, UUD		0
UUD, UHD	G1[0,6], G2[0,5], G3[6,1], G4[6,1], G5[8,2]	5
…	…,	…
UHD, UHD		0
UHD, UDU	G2[5,1], G2[5,3], G2[5,8], G3[1,7], G4[1,4], G5[2,6]	5
…	…	…
UHD, DDU	G3[1,4] G5[2,4]	1
UDU, UDU	G2[3,8], G4[4,7]	2
…	…	…

a first cutoff that is used to limit the size of the grouped genes and avoid unnecessary computations. The detailed pseudo code is below:

```
Input: The number of similar patterns,$h$.
Output: frequent $h$-itemsets.
Scan the transactions to find frequent $h$-itemsets
Scan the transactions $t \in T$to find frequent 1-itemsets;
for $(k = 2; k <= h; k++)${
    Generate candidate $k$-itemsets; // By merging a frequent
    $(k-1)$-itemsets.
    Prune candidate $k$-itemsets whose subsets are infrequent;
    Scan the transactions $t \in T$ to count the occurrences of
    itemsets in candidate $k$-itemsets; // $k$items' starting position
    do not overlap.
    Extract the frequent $k$-itemsets, a subset of candidate $k$-
    itemsets with counts no less than support $S$;
}
Return $h$-itemsets;
```

3.4. Phase 4: Further grouping of genes according to their similar pattern positions

When we have obtained frequent h-itemsets from the above phases, we need to further group the genes according the positions at which these similar patterns (items) occur simultaneously for the genes (transactions). Using Table 5 as an example, it is possible to see how further clustering is carried out. We already know that the itemset, UUD, UHD, in Table 5 occurs at positions, [0,6] in G1, [0,5] in G2, [6,1] in G3, [6,1] in G4 and [8,2] in G5. Suppose that a node is a set of positions for itemset, UUD, UHD, among the same gene (transaction). Hence, this set of nodes is {G1[0,6], G2[0,5], G3[6,1],G4[6,1],G5[8,2]}. The edge is the Euclidian Distance between two nodes. For example, the distance between node, G1[0,6], and node, G2[0,5] is $\sqrt{(0-0)^2+(6-5)^2} = 1$. We create a graph that contains these nodes and computed the edges as shown in Fig. 1a. When the graph is complete, the edge containing the minimum distance is first selected. If the selected edge's distance is less than the cutoff threshold d we set it to3 and the two nodes that are connected by this edge are merged (see Fig. 1b). The merged node's position is re-defined as the average of the original two nodes' positions (see Fig. 1b) and the distances along the edges that the merged node is connected with are re-computed for next merging. Next, we similarly select another edge that has the minimal distance for the whole graph and then merge two nodes if the minimal distance between them is less than the cutoff threshold d. Of course, we then should update the merged nodes and the edges that connect the new node with the other nodes. This merging process is continued until we can no longer find a minimal distance that is less than the cutoff threshold d. The final result is shown in Fig. 1d. We find that we have indeed grouped these genes into two groups, [G1,G2] and [G3,G4,G5] according to the similar pattern positions. The pseudo code is presented below:

```
Input: The frequent $h$-itemset and these transactions (genes)
that items (similar patterns) among this frequent $h$-itemset
occur simultaneously at.
Output: The result of grouping according the positions.

Build the initial graph;
Compute Euclid Distance between two nodes;
repeat
    minDistance = Select the minimal distance;
    if(minDistance < cutoff threshold $d$) {
        Merge two nodes and then update merged node and dis-
        tances between merged node and other nodes;
    }else{
        Break;
    }
end
Return grouping result;
```

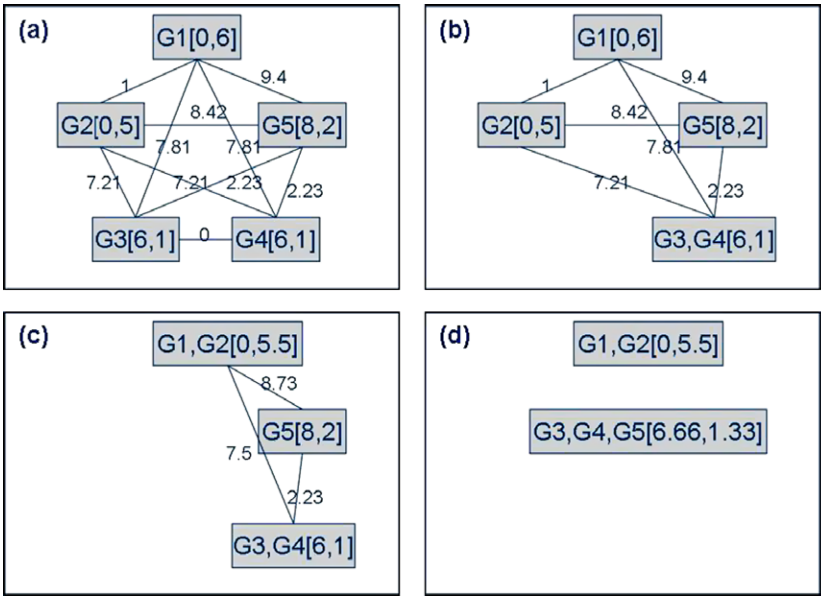

Fig. 1. The process of grouping. (a) The original graph has five node and 10 edges. The distances are already computed by Euclidian Distance. (b) The edge between G3 and G4 has the minimal distance 0 within whole graph. We merge node G3 and G4, and then update the merged node's positions. We also re-compute the distances between the merged node and the other nodes. (c) We merge nodes, G1 and G2, and update this merged node's positions by averaging the merging nodes' positions and then also re-compute the distances between the merged node and the other nodes. (d) The final result of grouping according to the original positions is two clusters.

3.5. Phase 5: Functional enrichment

Based on the hypothesis that co-expressed genes are likely to have a similar function, we can use this to help analyze the large number of clusters generated by our procedure. In order to show that the clusters we generated have biological significance, we use p-values that are obtained from the hypergeometric distribution. The probability (p-value) of observing at least m genes from a specific class within a study cluster of size n is given by

$$p\text{-value} = 1 - \sum_{i=0}^{m-1} \frac{\binom{f}{i} \times \binom{N-f}{n-i}}{\binom{N}{n}}$$

where f is the total number of genes from the population within a specific class and N is the total number of genes within the total population (defined as all genes represented on the microarray). Thus, the p-value corresponds to the probability of obtaining at least m gene of the cluster in a random set of size n. A low p-value indicates that the genes annotated belong to an enriched class that is statistically biologically significant. Our system has been coupled to Ontologizer (Robinson et al., 2004) to calculate p-values to detect statistically significant enrichment within one or more GO categories. Furthermore, we also calculate p-value for the statistically significant enrichment for some TFs' known target-gene groups. Moreover, we have modified some open source code from Ontologizer so that these subroutines are completely integrated into our system.

4. Experimental results

4.1. Clustering with the aim of finding time-lagged co-regulated genes

In this experimental system, we applied Spellman's yeast cell cycle dataset that includes 6331 open reading frames. The full dataset contains all the expression data for the alpha factor, cdc 15 and elutriation time course experiments. We used only the alpha factor dataset to validate our approach. We examined the results obtained with this dataset by our method for the detection of time-lag co-regulated genes over the cell cycle. We applied the approach to the cell-cycle regulated genes that were identified by Spellman's group (Spellman et al., 1998) in our experiments by using the identified cell-cycle regulated genes as the population of genes tested using our method. These identified cell-cycle regulated genes involves 800 genes and include 14 well-known cell-cycle TFs, SWI4, YOX1, SWI5, FKH1, RAP1, YHP1, HCM1, ACE2, STP2, GAT3, PHD1, GCR1, TEC1 and MET28. Furthermore, we add some other known cell-cycle TFs to the original dataset including ARR1, MBP1, RPN4, YAP1, FKH2, MCM1, STE12, YAP5, STP1 making a total of 809 genes. The Spellman's yeast cell cycle dataset covers two cell cycles and for this reason we formed clusters based on whether the gene profiles had two similar patterns of length 3 units over two cell cycles. The threshold s was set to 0.1 and the threshold t was set to 0.3. The distance d between two similar patterns was set to 2. The threshold D was 5 to take into account time lag. The support S for cluster size was set at 2 to avoid missing any significant clusters.

Our method is designed to fit the special needs of users; hence, users are able to pick out clusters of interest according to what they think are significant similar patterns within these clusters. For example, someone may believe that the patterns, UUH, UUD, are significant whereas the patterns, DDD, UUU, are insignificant. In order to filter some clusters that are insignificant, we can pick up clusters whose similar patterns both start with the character, U, and contain at least one TF. To realize the biological relevance of these chosen clusters, the p-value is calculated from the hypergeometric distribution in order to model the probability of observing at least $m$genes, from a cluster with n genes that may by chance contain a TF that regulates f known (documented) genes from the 809 genes in the dataset. We use YEASTRACT (Teixeira et al., 2006) to search for documented TF–target relationships for every cluster and then calculate the p-value for each TF that is grouped with its known regulated genes. We present only the top few clusters (p-value < 0.01), which are summarized in Table 9 in the Appendix. Although most of the clusters we have picked up do not show a statistically significant p-value, the result is really quite good. If Cluster 1 is examined, it is found to contain the TF, Fkh2, and this TF would seem to regulate the genes found in Cluster 1. In order to validate whether Fkh2 really regulates these Cluster 1 genes, we plot Fkh2 and the genes that the result is coordinated regulation (see Fig. 2). It can be seen that Fkh2 over-expresses simultaneously with or before the genes in this cluster and this supports the idea Fkh2 regulates these genes. Similarly, in chosen Cluster 2, Swi4 would seem to regulate the genes that form this cluster. In order to confirm regulation they were also plotted in a similar way to above (see Fig. 3). The Edge Detection and Event Method is one method of finding a regulated relationship. However, there are some drawbacks. Firstly, the result is regulator–target pairs instead of clusters, but a TF does not regulate only one gene but a group of genes with similar function; thus cluster results are more useful than pairs. Secondly, the regulator–target pairs that are identified by the Event Method only say there is a have high correlation and do not provide more detailed information such as the exact lag time.

We can also pick up clusters whose similar patterns both start with the character U but do not contain any known TFs. Then, we used YEASTRACT to obtain TFs that regulate some of the genes from the chosen clusters. We calculated the p-value obtained from the hypergeometric distribution to model the probability of

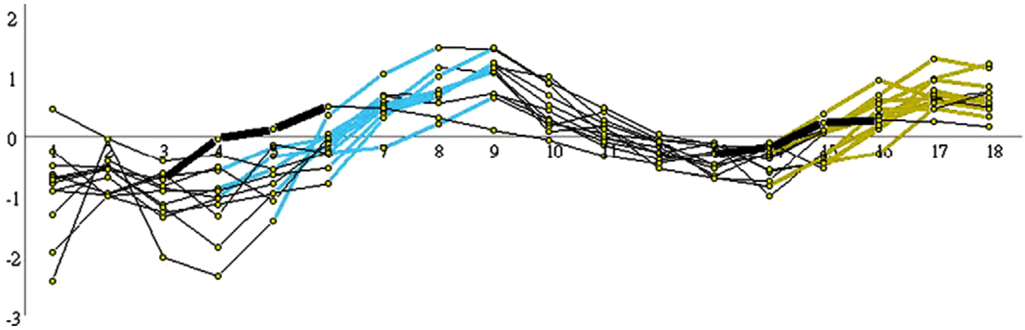

Fig. 2. Fkh2 (black bold line) really regulates the genes displayed here from Cluster 1 because it over-expresses simultaneously or before than known regulated genes.

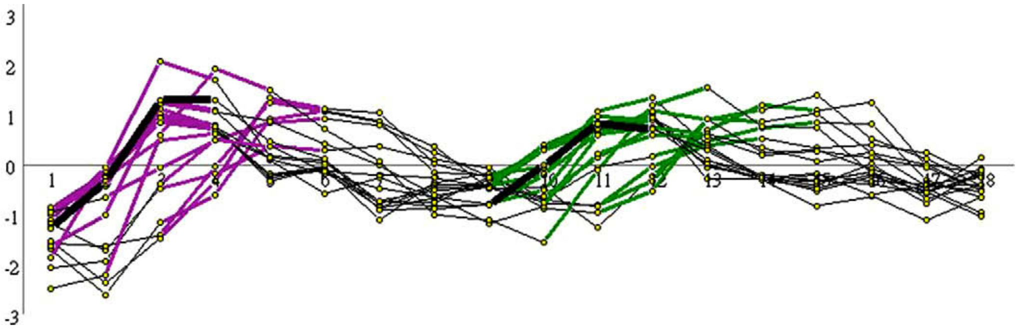

Fig. 3. Swi4 (black bold line) really regulates the genes displayed here from Cluster 2 because it over-expresses simultaneously or before than known regulated genes.

observing at least *m* genes from a chosen cluster with *n* genes by chance if a TF (not be included in this chosen cluster) regulates *f* known genes from the 808 genes in the dataset. We again present only the top clusters (p-value < 0.01), which are summarized in Table 9. The statistically most significant clusters (p-value < 0.01) that contain a TF and also contain this TF's regulated genes. It is clear that if the percentage of a cluster's genes that are regulated by a TF is much more larger than the percentage of the population genes that are regulated by the same TF, this cluster is more statistically significant.

Table 10 in Appendix the results show that although the chosen clusters do not contain any TFs, most of the genes among them seem to be regulated by a TF. Moreover, based on the fact that genes with a similar function are always regulated by same TF, the results reveal that the clusters generated by our method have some biological significance. We also plotted the TF for the known regulated genes and the gene expression profiles of the cluster to validate whether the group created by our method occurred by chance or had real biological significance. If we plot the TF Swi5 and genes' expression profiles from Cluster 21 (see Fig. 4), we find that Swi5 is over-expressed simultaneously or before the regulated genes and this supports the idea that Swi5 regulates these genes. Similarly, the TF Ace2 seems to regulate the genes from Cluster 16 (see Fig. 5).

5. Comparison of other approaches

In order to test whether our method could group co-regulated genes that other method find hard to group, we compared our method with some correlation measures such as Spearman Rank Correlation and Pearson Correlation. Initially, we picked two genes, MCM7 and MCM4, which are components of the MCM complex (Davey et al., 2003). Fig. 6 shows the expression levels of these

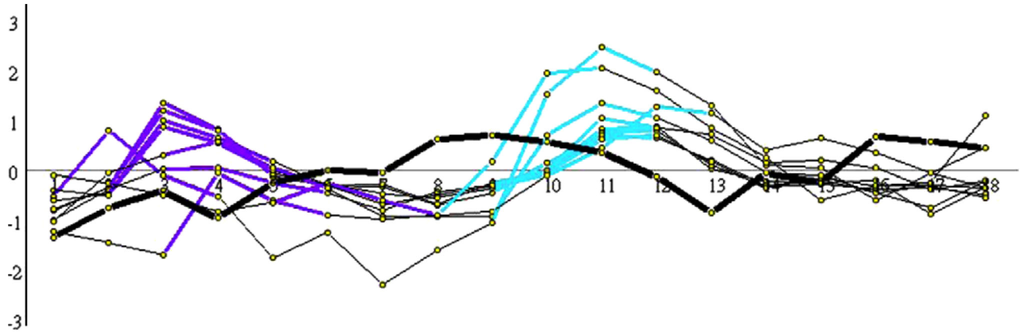

Fig. 4. Although Swi5 (black bold line) does not group with its known regulated genes, we could show that it really regulated the genes displayed here because it over-expresses simultaneously or before than known regulated genes.

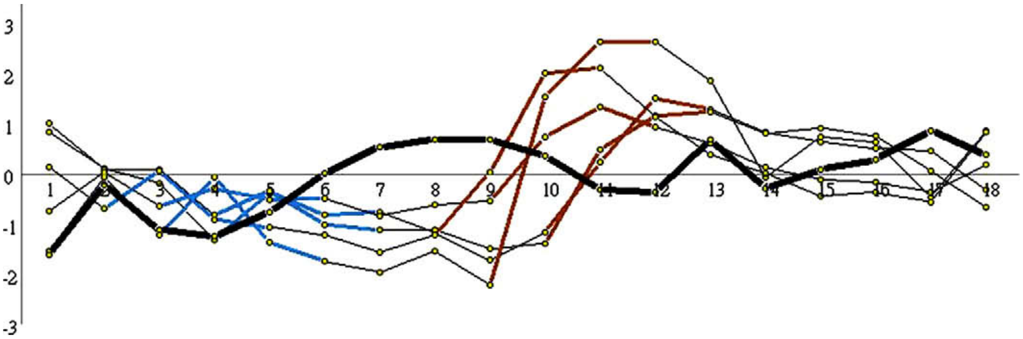

Fig. 5. Although Ace2 (black bold line) does not group with its known regulated genes, we could show that it really regulated the genes displayed here because it over-expresses simultaneously or before than known regulated genes.

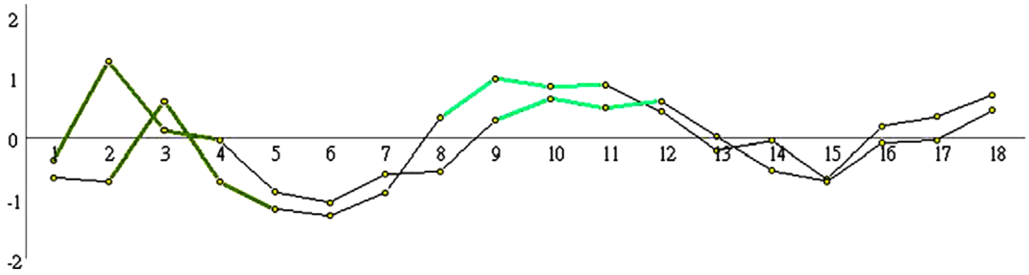

Fig. 6. The profiles of two genes, MCM7 and MCM4, which are components of the MCM complex.

Table 6
The pairwise similarity between MCM7 and MCM4 as calculated by Pearson Correlation and Spearman Rank Correlation.

Pearson Correlation	Spearman Rank Correlation
0.6	0.66

two genes and similar expression patterns that were detected by our method. Then, we calculated the pairwise similarity between the expression profiles of the individual genes according to Spearman Rank Correlation and Pearson Correlation (see Table 6). We find that the pairwise similar scores for MCM7 and MCM4 are actually quite low (⩽0.7). The reason for this is presence of scaled and shifted similarity patterns between these co-regulated genes and this causes these co-regulated genes to be difficult to group together by numerical measures. In contrast, our method transforms the expression profiles into event strings to take into account scaling and shift factors in the expression level between related genes.

We picked out three genes, CDC68, SWI4 and CLN1 that have a regulated relationship between each other. CDC68 is a required activator of SWI4 and SWI4 is a required activator of the G1 cyclin genes CLN1 and CLN2. CDC68's role at the CLN promoters may be indirect (Lycan et al., 1994).

Fig. 7 presents the expression levels of these three genes and we also calculated the pairwise similarity matrix between the expression profiles of these three genes according to Spearman Rank Correlation and Pearson Correlation (see Tables 7 and 8, respectively). In a similar way to the previous result, we found that the pairwise similar scores are not high with half of scores being relatively low (⩽0.7). Again, Spearman Rank Correlation and Pearson Correlation find it hard to group these genes using similarity matrixes. In reality, there are some drawbacks to these two numerical measures. Firstly, they do not provide detailed information on co-regulated gene pairs including time lag between them and the pattern similarity between two genes. The information on similar patterns that we are interested in includes the starting time point and what happens over time to the patterns such as the change between two neighborly time points.

Table 7
The pairwise similarity matrix for CDC68, SWI4 and CLN1 as calculated by Pearson Correlation.

	CDC68	SWI4	CLN1
CDC68	0	0.68	0.69
SWI4	0	0	0.83
CLN1	0	0	0

Table 8
The pairwise similarity matrix for CDC68, SWI4 and CLN1 as calculated by Spearman Rank Correlation.

	CDC68	SWI4	CLN1
CDC68	0	0.70	0.70
SWI4	0	0	0.91
CLN1	0	0	0

6. Conclusion

By converting the gene expression values, we present a local method to find potential TF–target relationships allowing the detection of time-lagged gene clusters based on periodically similar patterns over multiple cycles. Genes that have the same periodic patterns are grouped together and these genes are likely to be cell-cycle regulated genes that are controlled simultaneously by a TF. Generally, these TFs are included in same group as the regulated genes. The similarity of the two expression subprofiles is according to the specific changes in expression level by the genes. Our approach provides users with more detailed information about the detected similar patterns and users can use this information to choose the more significance clusters themselves and thus decrease the number of candidate clusters.

We have experimented with our method on a time-series expression dataset (Spellman et al., 1998). Genes with similar

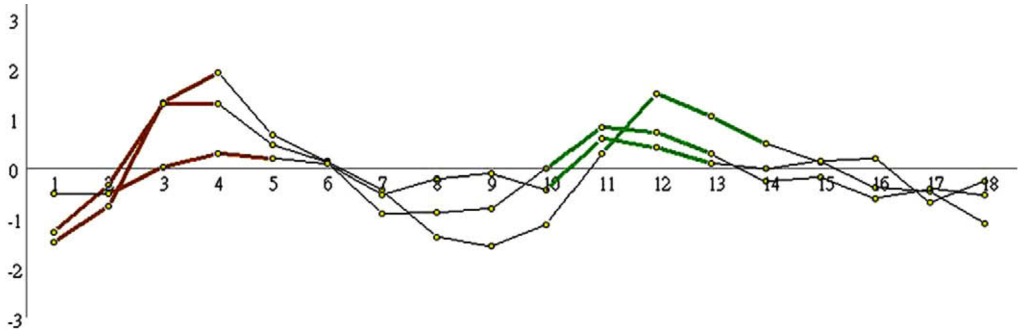

Fig. 7. The profiles of three genes, CDC68, SWI4 and CLN1, which have regulated relationships between each other.

patterns are found to be co-expressed genes and further, they are likely to have a TF–target relationship. In order to validate our method, we used p-value to show that the clusters we generated were able to help find TF–target relationships.

Our method was compared with some other measures such as Pearson correlation and Spearman rank correlation and we found that our method is able to group highly related genes together that other measures found hard to do. Furthermore, unlike the other methods, our approach can provide more detailed information on the TF–target relationships. Finally, the results from the Edge Detection and Event Method are pairs of genes and the clusters obtained by our method are clearly more useful than pairs for finding a TF–target relationship.

Appendix A

See Tables 9 and 10.

Table 9
The statistically most significant clusters (p-value < 0.01) that contain a TF and also contain this TF's regulated genes. It is clear that if the percentage of a cluster's genes that are regulated by a TF is much more larger than the percentage of the population genes that are regulated by the same TF, this cluster is more statistically significant.

Cluster number	Similar patterns	Number of genes (n)	Cluster with this TF	Number of genes be regulated by this TF (m)	% of this cluster's genes be regulated by this TF	% of population genes be regulated by this TF (f)	p-Value	Genes are included in this cluster are this TF's known regulated genes
1	UUH, UUH	25	Swi4	18	72	15.9	2.74E−10	YPL127C, YJL187C YMR199W, YPL267W YPR202W, YER189W YHL049C, YEL075C YDR224C, GR109C YBL003C, YHR218W YPL163C, YER111C YCR065W, YMR307W YDR225W, YFL064C
2	UUN, UUU	25	Fkh2	12	60	10.1	3.51E−8	YNL068C, YJR092W YPL141C, YNL058C YJL051W,YPR119W YML119W, YML034W YMR032W, YLR190W YGR108W,YHR152W
3	UUU, UUU	77	Fkh2	16	33.3	10.1	4.81E−6	YMR001C, YPL141C YJL051W, YHR023W YPR119W,YIL106W YML034W, YNL068C YJR092W, YKL096W YPL242C, YNL058C YPL155C, YLR131C YPR156C, YHR152W
4	UHD, UUH	26	Swi4	14	53.8	15.9	6.25E−6	YJL187C, YGR014W YPL267W, YPR202W YER189W ,YHL049C YEL075C, YGR109C YHR218W, YPL163C YER111C, YCL025C YCR065W, YFL064C
5	UHD, UUH	31	Swi4	15	50	15.9	9.42E−6	YJL187C, YMR199W YPL267W, YPR202W YER189W, YHL049C YEL075C, YOR074C YGR109C, YHR218W YPL163C, YPR120C YER111C, YCR065W YFL064C
6	UUH, UUN	10	Fkh2	6	60	10.1	1.37E−4	YNL068C, YJR092W YPR119W, YML034W YLR190W, YGR108W
7	UHD, UHD	33	Swi4	15	46.8	15.9	2.57E−4	YJL187C, YPL267W YPR202W, YER189W YHL049C, YEL075C YOR074C, YGR109C YHR218W, YPL163C YPR120C, YER111C YCL025C, YCR065W YFL064C
8	UDD, UUU	18	Yhp1	5	31.2	5.3	9.48E−4	YJL194W, YPR019W YLR274W, YIL106W YBR202W

Table 9 (*continued*)

Cluster number	Similar patterns	Number of genes (n)	Cluster with this TF	Number of genes be regulated by this TF (m)	% of this cluster's genes be regulated by this TF	% of population genes be regulated by this TF (f)	p-Value	Genes are included in this cluster are this TF's known regulated genes
9	UDD, UDD	129	Met28	7	6.1	1.7	1.28E−3	YIR017C, YPR167C YER091C, YJL078C YGR055W, YFR030W YGL184C
10	UDD, UUD	177	Tec1	7	4.6	1.7	7.64E−3	YNL283C, YDR055W YNL166C, YER070W YCR018C, YML100W YBR083W
11	UHD, UUU	34	Fkh1	6	22.2	6.9	7.97E−3	YNL068C, YPR119W YPL155C, YCL063W YAR071W, YOR315W

Table 10
The statistically most significant clusters (p-value < 0.01) do not contain any TFs but in which most of the genes are regulated by a TF. It is clear that if the percentage of a cluster's genes that are regulated by a TF is much more larger than the percentage of the population genes that are regulated by the same TF, this cluster is more statistically significant.

Cluster number	Similar patterns	Number of genes (n)	Cluster without this TF	Number of genes be regulated by this TF (m)	% of this cluster's genes be regulated by this TF	% of population genes be regulated by this TF (f)	p-Value	Genes are included in this cluster are this TF's known regulated genes
1	UHD, UUU	62	Mbp1	18	38.2	12.1	1.71E−6	YDR309C, YOR230W YGR014W, YNL339C YIL177C, YHR219W YHL049C, YDR113C YDL018C, YKL096W YMR179W, YJL074C YHR218W, YHR153C YLR462W, YML027W YGR296W, YJL225C
2	UHH, UHH	7	Swi4	7	100	15.9	2.28E−6	YPL127C, YDR224C YLR183C, YBL003C YKR013W, YMR307W YDR225W
3	UHH, UUH	13	Swi4	9	69.2	15.9	2.1E−5	YPL127C, YMR199W YDR224C, YLR183C YBL003C, YKR013W YGR189C, YMR307W YDR225W
4	UUH, UUU	53	Swi4	17	41.4	15.9	5.18E−5	YMR199W, YNL339C YKL008C, YPR202W YHR219W, YHL049C YEL075C, YNL300W YDL018C, YBL003C YMR179W, YHR218W YGL038C, YLR462W YML027W, YGR296W YMR307W
5	UDD, UHD	87	Mbp1	22	27.5	12.1	5.85E−5	YDR309C, YJL073W YJL187C, YBL111C YAR008W, YJR030C YNL339C, YIL177C YNL030W, YHR219W YDL018C, YKL096W YMR179W, YDR545W YDL003W, YHR153C YLL067C, YLR103C YLR462W, YML027W YGR296W, YJL225C
6	UUN, UUU	9	Fkh1	5	62.5	6.9	6.37E−5	YMR215W, YMR001C YJL051W, YCL063W YPR156C
7	UDD, UUH	51	Mbp1	16	32.6	12.1	7.47E−5	YJL187C, YAR008W YNL339C, YNL030W YHR219W, YOR075W YDL018C, YMR179W YDR545W, YDL003W YHR153C, YLL067C YLR103C, YLR462W

(*continued on next page*)

Table 10 (*continued*)

Cluster number	Similar patterns	Number of genes (n)	Cluster without this TF	Number of genes be regulated by this TF (m)	% of this cluster's genes be regulated by this TF	% of population genes be regulated by this TF (f)	p-Value	Genes are included in this cluster are this TF's known regulated genes
								YML027W, YGR296W
8	UDU, UHD	36	Gat3	7	19.4	3.3	8.86E−5	YCR041W, YEL076C YHL049C, YEL075C YLR463C, YFL065C YFL064C
9	UDU, UUH	31	Gat3	6	20	3.3	2.18E−4	YLR467W, YEL076C YHL049C, YEL075C YLR463C, YFL064C
10	UUH, UUU	30	Swi4	13	43.3	15.9	2.55E−4	YPL127C, YJL187C YDR507C, YGR086C YIL141W, YKL113C YPL256C, YBL003C YDL055C, YMR307W YER001W, YDR225W YNL031C
11	UDD, UUN	11	Fkh2	6	54.5	10.1	2.77E−4	YPL141C, YNL058C YJL051W, YML119W YMR032W, YHR152W
12	UDL, UUU	9	Mcm1	4	66.6	8.8	7.57E−4	YJR092W, YPR119W YGL021W, YLR131C
13	UUH, UUU	28	Fkh1	7	31.8	6.9	3.95E−4	YJR092W, YPR119W YGL021W, YPL155C YLR131C, YML034W YAR071W
14	UDD, UUU	60	Swi5	11	21.1	7.1	6.01E−4	YDR055W, YKL164C YPL283C, YGR086C YBR083W, YLR464W YLR467W, YGR041W YKL163W, YNL328C YLR463C
15	UDD, UUD	20	STP1	5	26.3	4	6.4E−4	YER070W, YJR154W YDR501W, YML100W YLR121C
16	UDN, UUH	22	Ace2	5	22.7	4	1.33E−3	YER124C, YNL327W YKL185W, YHR143W YLR079W
17	UUD, UUN	15	Yap5	4	28.5	5.9	7.04E−3	YIL129C, YPL141C YNL058C, YIL158W
18	UDU, UUU	76	Gat3	7	12.5	3.3	1.56E−3	YLR467W, YHL049C YEL075C, YLR462W YLR463C, YBL112C YPR203W
19	UDN, UHD	24	Ace2	5	20.8	4	2.02E−3	YER124C, YNL327W YKL185W, YHR143W YLR079W
20	UDN, UUD	13	Yhp1	4	30.7	5.3	3.49E−3	YJL115W, YNL339C YLR413W, YLL067C
21	UDD, UUH	48	Swi5	9	19.1	7.1	4.22E−3	YLR464W, YLR467W YKL164C, YLR049C YPL283C, YLR463C YJL078C, YGR086C YPL158C
22	UDD, UDN	41	Gcr1	4	10.8	1.9	4.55E−3	YGR240C, YMR055C YGR143W, YIL162W
23	UDD, UUU	39	Mcm1	8	22.8	8.8	8.77E−3	YMR001C, YLR040C YJL157C, YJL194W YNL058C, YLR274W YGR143W, YHR152W
24	UDD, UDN	17	Rpn4	6	35.2	10.7	6.08E−3	YOL016C, YDR055W YBL023C, YKR012C YLR013W, YOR273C
25	UDD, UdN	39	Yap5	6	18.7	5.9	8.83E−3	YLL066C, YPL208W YLL067C, YML050W YBR007C, YPL283C
26	UHH, UUU	25	Mcm1	6	27.2	8.8	9.53E−3	YJR092W, YGL021W YLR131C, YIL158W YBR202W, YDR451C

References

Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules, Proc. of the 20th VLDB Conf., Santiago, Chile.

Balasubramaniyan, R., Hullermeier, E., et al. (2005). Clustering of gene expression data using a local shape-based similarity measure. *Bioinformatics, 21*(7), 1069–1077.

Cheng, Y., & Church, G. M. (2000). Biclustering of expression data. In *Proceedings of the International Conference on Intelligent Systems of Molecular Biology* (Vol. 8, pp. 93–103).

Cho, R. J., Campbell, M. J., et al. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell, 2*(1), 65–73.

Davey, M., Indiani, C., et al. (2003). Reconstitution of the Mcm2-7p heterohexamer, subunit arrangement, and ATP site architecture. *Journal of Biological Chemistry, 278*(7), 4491–4499.

Filkov, V., Skiena, S., et al. (2001). Identifying gene regulatory networks from experimental data. In *Proceedings of RECOMB*.

Gusfield, D. (1997). *Algorithms on strings, trees and sequences: Computer science and computational biology*. London: Cambridge University Press.

Ji, L., & Tan, K. L. (2005). Identifying time-lagged gene clusters using gene expression data. *Bioinformatics, 21*(4), 509–516.

Kato, M., Tsunoda, T., et al. (2001). Lag analysis of genetic networks in the cell cycle of budding yeast. *Genome Informatics, 12*, 266–267.

Kwon, A. T., Hoos, H. H., et al. (2003). Inference of transcriptional regulation relationships from gene expression data. *Bioinformatics, 19*(8), 905–912.

Lycan, D., Mikesell, G., et al. (1994). Differential effects of Cdc68 on cell cycle-regulated promoters in *Saccharomyces cerevisiae*. *Molecular and Cellular Biology, 14*(11), 7455–7465.

Robinson, P. N., Wollstein, A., et al. (2004). Ontologizing gene-expression microarray data: Characterizing clusters with Gene Ontology. *Bioinformatics, 20*(6), 979–981.

Spellman, P. T., Sherlock, G., et al. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast *Saccharomyces cerevisiae* by microarray hybridization. *Molecular Biology of the Cell, 9*(12), 3273–3297.

Teixeira, M. C., Monteiro, P., et al. (2006). The YEASTRACT database: A tool for the analysis of transcription regulatory associations in *Saccharomyces cerevisiae*. *Nucleic Acids Research, 34*(Database issue), D446–D451.

Ukkonen, E. (1995). On-line construction of suffix trees. *Algorithmica, 14*, 249–260.

Whitfield, M. L., Sherlock, G., et al. (2002). Identification of genes periodically expressed in the human cell cycle and their expression in tumors. *Molecular Biology of the Cell, 13*(6), 1977–2000.