GENE EXPRESSION DATA CLUSTERING BASED ON LOCAL SIMILARITY COMBINATION*

DE PAN, FEI WANG†

Department of Computer Science and Engineering
220 Handan Road,
Fudan University, Shanghai 200433, P.R.C.
E-mail: {pande,wangfei}@fudan.edu.cn

Clustering is widely used in gene expression analysis, which helps to group genes with similar biological function together. The traditional clustering techniques are not suitable to be directly applied to gene expression time series data, because of the inhered properties of local regulation and time shift. In order to cope with the existing problems, the local similarity and time shift, we have developed a new similarity measurement technique called *Local Similarity Combination* in this paper. And at last, we'll run our method on the real gene expression data and show that it works well.

1. Introduction

The increasingly used microarray techniques generate more and more biological data veryday to the biologists, who reasonably need special computational tools to help. The data obtained from microarray technique records the expression levels of genes in the form of time series points by measuring the concentration of the corresponding mRNAs.[1,2,3] Thus it provides the possibility and opportunity to insight the genes' behaviors and functions indirectly, and to find gene pairs with different kind of regulation relationships and group genes together with similar biological function, which usually demonstrate similar expression profiles against the time series, and at last to construct a biological network.[4,5]

Clustering genes with highly similar expression profiles, locally or globally and time shifted, is one of the most important steps to analyse the microarray data, which usually applies a kind of similarity measurement first, such as the traditional Pearson correlation or Euclidean distance[6] methods, and then a clustering paradigm follows which classifies genes basing on their pairwise similarities into different clusters. Here the time shift means a time lag between the local parts of the two gene profiles, because the first gene——the regulator, usually effects the downstream gene——-the regulated, with a time delay. The clustering structure will reveal the transcriptional information of the genes in the biological environment to help the understanding of the biological control mechanism. And in a deeper analysis, the clustering results will serve, usually with many other kinds of biological in-

*This work is supported by grants 60303009, 60496325 of Chinese National Natural Science Foundation.
†Correspondence should be addressed to F. wang email:wangfei@fudan.edu.cn

formation and data, such as transcriptional factors and binding sites, or the protein-protein interaction information, to insight the gene functions in the molecular level.

In order to analyse the time series microarray data, many methods have been developed to measure the similarity between genes. Here we'll give a brief review about the related similarity measurement methods. The aforementioned Euclidean method is widely used in other scientific or engineering fields, which usually directly dismisses the time information and only focuses on the global profile distance calculation, and rarely works well on microarray data. The well-known standard Pearson correlation method also computes the global similarity and ignores the time series characteristic. The modified Pearson correlation method[7] based on the standard Pearson Correlation was developed quickly, which takes the time lag into account, but which also considers the global time lag only and there is no good approach to deciding the size of lag yet. And a particular method introduced by Spellman et al in 1998 applied the Fourier transformation on the time series data,[8] which was proved to be effective for the periodic data. Recently, some more sophisticated methods have been presented. The *Edge Detection Method* tries to find the main changes in expression levels $(edge)$ and gives a score by comparing the edges between the two genes,[9] which will lose information when two edges are far apart. Another method is *Dominant Spectral Component Method*,[10] which first decomposes the time series data into frequency components and attains a pair of frequent component with least difference, and then transforms them back to time space and uses the standard Pearson coefficient to calculate the similarity. The *Event method*[11] transforms time series expression level into a string of events—- R(Rising), F(Falling) & C(No changing, and gives each gene pair a score by applying the *Needleman-Wunsch* alignment algorithm, which will aslo lose too much information while only giving a global similarity score.

We propose a new method in our work to measure the similarity between micaoarray gene expression profiles, which takes the local similarity and time shift properties into account, yet simultaneously solved by the previous methods. Our method will discretize the data first and then apply a matrix to find all the local matching information including the time lag. Then a optimal combination of the local matches follows to attain a global similarity.

2. Method

In this section we'll describe the paradigm of our new similarity measurement, in which the original expression data will be discritized by using two equations and, then we'll demonstrate how to use a matrix to discover all the local matching information. And we'll give the definition of the optimal combination of the local match candidates to obtain the optimal alignment between the gene pair, and by defining an equation we'll finally get the similarity.

2.1. *Data Discretization*

Here we'll use a 3-value discretization method to preprocess the original gene microarray expression data matrix $G_{n\times m}$ to get the discretized matrix $E_{n\times(m-1)}$, where n means the

gene number and m indicates the time condition points. And before we get the object matrix we'll apply Equ.(1) on G firstly to get a temp matrix $G'_{n\times(m-1)}$. Then the Equ.(2) will be used G' to attain object matrix E.

$$G'_{i,j} = \begin{cases} 1, & \text{if } G_{i,j} = 0 \ \& \ G_{i,j+1} > 0, \\ -1, & \text{if } G_{i,j} = 0 \ \& \ G_{i,j+1} < 0, \\ 0, & \text{if } G_{i,j} = 0 \ \& \ G_{i,j+1} = 0, \\ \frac{G_{i,j+1}-G_{i,j}}{|G_{i,j}|}, & \text{if } G_{i,j} <> 0. \end{cases} \tag{1}$$

$$E_{i,j} = \begin{cases} G'_{i,j}, & \text{if } G'_{i,j} = 0, -1, 1, \\ 1, & \text{if } G'_{i,j} >= t, \\ -1, & \text{if } G'_{i,j} <= -t, \\ 0, & \text{if } -t < G'_{i,j} < t\ . \end{cases} \tag{2}$$

The parameter t is a customized threshold, which we empirically set to be 1.0. This value means that only an apparent change——increasing or decreasing——in expression level can be assigned to the discretized value 1 or -1, and all others should be 0 for the reason to maximally eliminate the potential noise in the original data.

2.2. *Local Matching*

In order to calculate the similarity between two genes, denoted by $S(X, Y)$, we'll use a matching matrix to find all the local matching information. Here X and Y stand for two genes respectively, and represented by sequences $(x_1, x_2, ..., x_{m-1})$ and $(y_1, y_2, ..., y_{m-1})$ derived from the discretized matrix E. And we define $X(i, j)$ to be the subsequence $(x_i, x_{i+1}, .., x_j)$ of X and $X(i)$ is the ith element of X.

In our method, we will calculate similarity by finding all the possible subsequence matching between the X and Y, and then find a optimal combination of the local candidate matches. A local match can be defined to be two subsequences from two genes respectively exactly having the same sequence: $X(i_1, j_1) = Y(i_2, j_2)$ and $X(i_1 + k) = Y(j_1 + k)$ for each $k(0 <= k <= j_1 - i_1)$, where we have $1 <= i_1 < j_1 <= m - 1$, $1 <= i_2 < j_2 <= m - 1$ and $|j_1 - i_1| = |j_2 - i_2|$.

After we have found all the local matches between the genes, we should combine the local matches into an optimal global match with longest length. We should introduce several important parameters. The first is the *minSubLen* which defines the minimum length of a matched subsequence. A local match with too short length means nothing but high random probability in matching. The second parameter, the *maxTimeLag*, is the time shift between the two subsequences which is the difference of i_1 and j_1. A too big time lag is difficult to explain in biology but there always exists time shift when the gene regulation works. So we'll confine the time shift in a limitation. After that, we'll find that some local candidate matches have the problem of overlapping which will not be allowed in the optimal combination.

Here we'll apply a matrix $M_{m\times m}$ to find all the candidate local matches. The first row and first column will be initialed to be 0. And after that we'll fill the matrix as the algorithm in Fig.1 .

1)**Begin**
2) **for** $i = 0$ **to** $m - 1$
3) **for** $j = 0$ **to** $m - 1$
4) **if** $i == 0$ **or** $j == 0$
5) $M(i, j) = 0$**;**
6) **else if** $X'(i) == Y'(j)$
7) $M(i, j) = M(i - 1, j - 1) + 1$**;**
8) $M(i - 1, j - 1) = 0$**;**
9) **else**
10) $M(i, j) = 0$
11) **end if**
12) **end for**
13) **end for**
14)End

Figure 1. Algorithm for mining local matches

And now we need to consider the two restrictions: the *minSubLen* and the *maxTimeLag*, which will confine our combination on the cell $M_{i,j}$ satisfying $|i-j| <=$ *maxTimeLag* and $M_{i,j} >=$ *minSubLen*. As a result, some useless candidate subsequence matches will be eliminated, which will greatly reduce the computation complexity to find the optimal combination. Here we give a formal description of the optimal combination problem. Given a set of triples $S = \{s_1, s_2, ..., s_n\}$ with each $s =_{def} (len, rowI, colI)$ in S recoding the information of a cell in M satisfying the aforementioned conditions including the value of the cell and its row and column indexes respectively. Here the value len records the length of matched subsequence, and the match in the two genes ends at positions $rowI$ and index $colI$ respectively. We define the operator *Non-Conflict*(s_i, s_j) to be TRUE if for any $s_i \in S$ and $s_j \in S(i <> j)$ with $s_i.rowI + s_i.len <= s_j.rowI$ and $s_i.colI + s_i.len <= s_j.colI$, or $s_j.rowI + s_j.len <= s_i.rowI$ and $s_j.colI + s_j.len <= s_i.colI$. Now our optimal problem is to find a subset $S' \subseteq S$ that maximizes $\sum_{s'\in S'}(s'.len)$ and any s'_i and s'_j is non-conflict. Then the attained subset S' will serve for the similarity calculation. Here we have proved the general *Optimal Local Combination Problem* to be NPC, which can be reduced from the Weighted Independent Set problem. And we also calculate the size of $|S|$ to obtain 0.6722C as an upper bound, where C is the length of a gene in G when we set*minSubLen* and *maxTimeLag* to be 4 and 3 respectively, which actually makes a brute searching method possible. The solution will list all the legal subset of S with no conflict elements and get the subset with maximized $\sum_{s'\in S'}(s'.len)$ as an optimal combination. We'll sort the elements into a list S in lexicographic order first when implement the brute searching which will reduce the comparison times.

Given gene X and gene Y with the sequences X=(-1,0,1,-1,1,-1,1,-1,1,0,1,-1,1,-1,1,-1,1) and Y=(1,0,0,0,-1,1,-1,1,-1,1,-1,1,-1,1,-1,0,0) respectively, we'll show the matching matrix of gene X and gene Y in Tab.1. It's easy to select the local matches with length greater than the *minSubLen* and we can confine the cells along the diagonal from upper-left to down-right with $|i - j| <= maxTimeLag$. In Tab.1 the candidate cell appears in bold. In the above example, we have

Table 1. Matching matrix

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	1	0	1	0	0	0	1	0	1	0	1	0	1
0	0	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	2
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	**4**	0	0	0	0	0	0	0	4
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	**6**	0	0	0	0	0	0	0	6
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	**7**	0	0	0	0	0	0	0	7
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	**7**
0	0	0	0	2	0	4	0	6	0	0	0	2	0	**4**	0	**6**	0
0	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
0	0	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

S=$\{(4,8,9),(6,10,9),(7,12,9),(4,15,14),(6,15,16),(7,14,17)\}$, where we set *minSubLen* to be 4 and *maxTimeLag* to be 3. Now our job is to find a subset $S' \subseteq S$ satisfying the aforementioned constraints. Here we get $S' = \{(4,8,9),(7,14,17)\}$, which is the optimal local combination. And a triple such as (7,14,17) means a local match with length 7, and ends at index 14 of gene Y and index 17 of gene X.

2.3. *Exact Similarity*

In this section, we'll calculate the similarity of the gene pair after we find the set S'. Assume the size of $|S'|$ to be K, which is the number of elements in the set. Then we'll use the following Equ.(3), where the parameter K is used to adjust the similarity, a higher K meaning more punishment because one long global match obviously has higher similarity than that calculated from the local combination. So a punishment of K with high value will be reflected in the formulation, and 10 in the second multiplication operator under the

radical sign is usually set to be half of the gene length. And the constant C usually is set to be the length of gene.

$$S(X,Y) = \sqrt{\frac{\sum_{s' \in S'} s'.len}{C} \left(1 - \frac{K-1}{10}\right)}. \quad (3)$$

Our similarity method focuses on the local similarity, and the regulation between genes often functions locally as well as a time lag exists. The global similarity or distance measurement, the Pearson correlation or the Euclidean distance, have difficulties to solve such problems. Even only one time slice lag will greatly reduce the similarity between two genes, and a local similarity will often not be found when applying a global similarity calculation method. In our method, it is easy to locate the local similarity and even give the time lag between the local matching. The triple in S' fairly records the time lag information, and the value of $|s'.rowI - s'.colI|$ gives the time lag of the local matching. To demonstrate the difference of our method and the *Pearson Correlation* method, Fig.2

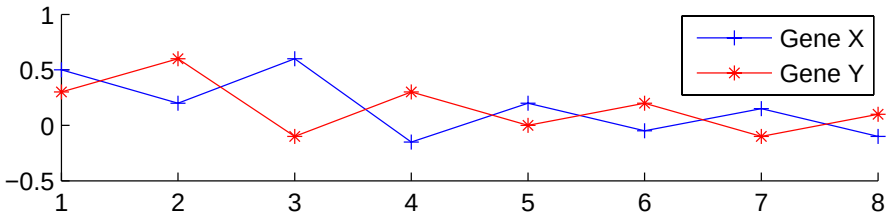

Figure 2. Our similarity method for the profiles is 0.93 whereas the Pearson correlation is -0.38.

shows two profiles with length 8, which have a highly correlated relationship according to our similarity method but almost unrelated when applied the Pearson correlation, and where the -0.38 means a negative regulation——one gene depresses another geneś expression. The reason is that the profiles have a high similar profiles except one slice time offset which the Pearson correlation has difficulty to cope with but our method can easily reflect.

3. Experimental Results

In this section we'll report the experimental results by applying our method on the real gene microarray data.

3.1. *Data & Clustering*

In order to demonstrate the performance and correctness of our new method, we'll run our method on the real time series gene data. We use the earliest gene expression data which is accessible at *Paul T. Spellman's* website [8] and also widely used in academic research. The data is mainly attained by four independent experiments for synchronized reasons: factor arrest, elutriation, arrest of a cdc15 temperature-sensitive mutant and cdc28, which

consists of all the 6178 Yeast ORFs. But we'll not directly use this data, and there is too much of them. In the research of Steven Skiena and Vlfilkov(http://www.cs.sunysb .edu/ skiena/gene/jizu/), they searched the Yeast database and got 1007 genes from that in Feb. 2000. And by reviewing the published literatures on these genes, they collected 888 gene regulations, positive or negative. On this basis, we find 288 genes in alpha data set with their known regulation relationship among the 888 gene regulation relationship. Here the alpha data set has 18 time condition points with 7 minutes interval.

We'll run our method on the alpha data set and construct a similarity matrix for the genes in the data set to record the pairwise gene similarity, which is symmetrical and used for later clustering.

Clusters will de attained from the similarity matrix by using the GCLUTO,[12] a clustering tool consisting several clustering analysis methods. Here we use the clustering option based on the graph partition method.[13] GCLUTO will first construct a graph where each gene is represented by a node and edges between nodes are assigned with the corresponding similarities. GCLUTO will cut the graph with an optimal approach recursively until a pre-specified number of clusters been attained.

3.2. *Results Evaluation*

The number of clusters is a critical parameter in the clustering which will greatly effect the robustness of the clustering structure. Fortunately, when the parameter is over 20, the clustering structure is quite robust to the variation of the parameters.

Fig.3 and Fig.4 show cluster 7 in 30-ways clustering and cluster 8 in 25-ways clustering. They have a great high similarity, except in Fig.4 there are a little more genes for there are 5 less clusters. But the genes in Fig.3 can be found in cluster 7 in 25-ways clustering. And Fig.5 and Fig.6 both demonstrate cluster 2 in 30-ways clustering and in 25-ways clustering respectively. They also have a high similarity. And it is the same with the previous cluster pair, that the genes in cluster 2 in 30-ways can all be found in cluster2 in 25-ways. At last, in our experimental results, all the other clusters in the 25-ways clustering except no 2 and 8, they all can find a corresponding cluster in the 30-ways clustering with high similarity except several genes more or less. As a fact, when the gene clusters number is over 20, the structure is rather robust and changes little with the pre-specified cluster number increasing.

In Fig.3 and Fig.4, we obviously find two main profiles of the genes, and there is a time lag between them. This is very difficult for the traditional similarity or distance measurement, Pearson correlation or Euclidean distance, to find such clusters with time lag existing. But our method can give a high similarity between genes with similar profiles even there is a time lag, where the inter similarity in the cluster is 0.787.

In Tab.2, we shows the statistical results of the first 10 clusters in the 30-ways clustering for space reason. The column labeled Size displays the number of objects that belongs to each cluster. The column labeled ISim displays the average similarity between the objects of each cluster. The column labeled ISdev displays the standard deviation of these average internal similarities. The column labeled ESim displays the average similarity of the objects

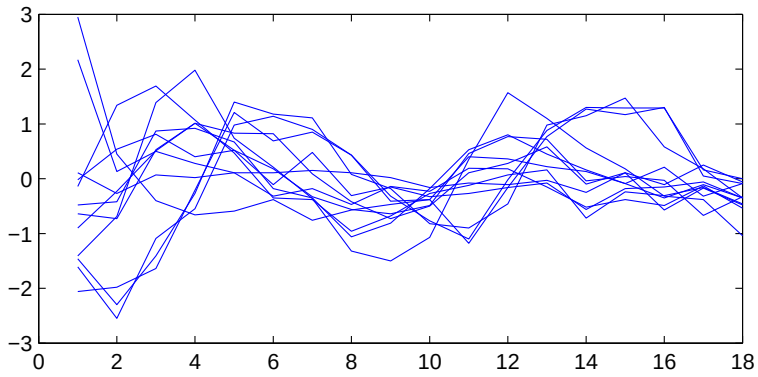

Figure 3. Cluster 8 in 30-ways clustering

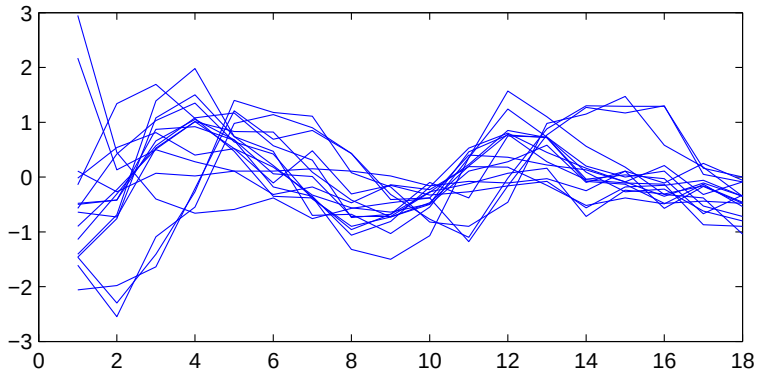

Figure 4. Cluster 7 in 25-ways clustering

of each cluster and the rest of the objects . Finally, the column labeled ESdev display the standard deviation of the external similarities. The ISim is much higher than the ESim and the ISdev and ESdev are much lower compared with the previous values. Now we can conclude that our method has successfully find the clusters with high similarity.

4. Discussion

We have proposed a new gene expression similarity measurement, which has successfully solved the local regulation and time lag problems in microarray data. By discretizing the original data and finding their local match, an optimal combination can be attained from the local matches and get the global match. And we also check our method on the real gene expression data and find that the clustering structure is rather robust. The genes in the same cluster also demonstrate high similarity. In the future work, we can compare our results with the gene database, for instance the MIPS, to check the genes clustered in the

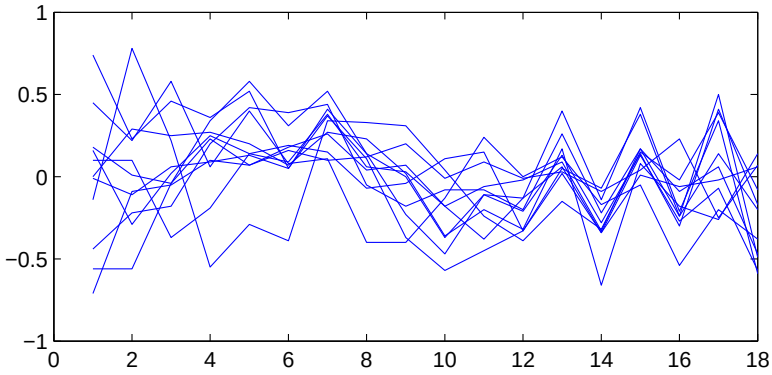

Figure 5. Cluster 2 in 30-ways clustering

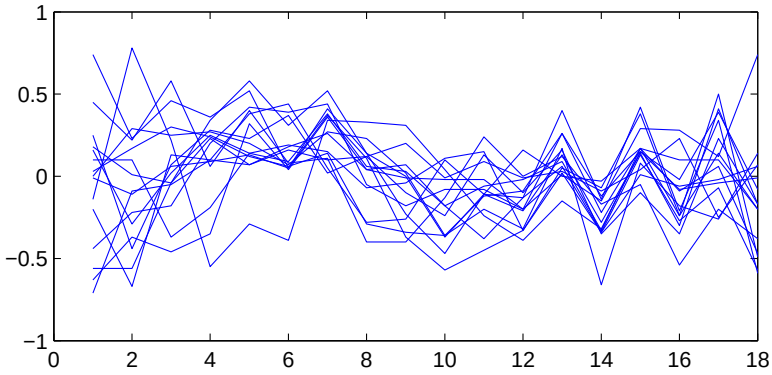

Figure 6. Cluster 2 in 25-ways clustering

same cluster whether have a similar biological function. The data and software is available upon request.

References

1. Geogre C. Tseng, Min-Kyu Oh, Lars Rohlin, James C. Liao & Wing Hung Wong. Issues in cDNA microarray analysis: qulity filtering, channal normalization, models of variations and assements of gene effects. *Nucleic Acid Research*, 2001, Vol.29, No.12, 2549-2557.
2. Brazma A., Vilo J.. Gene expression data analysis. *Federation of European Biochemical Societies: Letters*, 2000, Vol.480(1),17-24.
3. T. Forster, D. Roy & P. Ghazal. Experiments using microarray technology: limitations and standard operating procedures. *Journal of Endocrinology*, 2003, 178, 195C204.
4. Min Zou & Suzanne D. Conzen.A new Dynamic Bayesian Network (DBN) approach for identifying gene regulatory networks from time course microarray data. *Bioinformatics*, 2005, Vol.21(1) , 71C79.

Table 2. Statistic Analysis of the 30-ways clustering.

$Cluster$	$Size$	$ISim$	$ISdev$	$ESim$	$ESdev$
0	4	0.640	0.069	0.426	0.060
1	12	0.796	0.009	0.410	0.055
2	11	0.758	0.036	0.421	0.020
3	6	0.758	0.064	0.386	0.054
4	4	0.602	0.029	0.402	0.050
5	6	0.671	0.036	0.423	0.033
6	11	0.717	0.042	0.396	0.027
7	15	0.704	0.039	0.372	0.033
8	16	0.711	0.044	0.328	0.033
9	9	0.606	0.022	0.373	0.043

5. Shoudan Liang. A genereal reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing*, 1998, Vol.3, 18-29.
6. Gerstein & R. Jansen. The current excitement in bioinformatics-analysis of whole-genome expression data: how does it relate to protein structure and function? *Curr. Opin. Struct. Biol*, 2000, Vol.10, 574C584.
7. Mamoru Kato, Tatsuhiko Tsunoda, Toshihisa Takagi. Lag Analysis of Genetic Networks in the Cell Cycle of Budding Yeast. *Genome Informatics*, 2001, Vo.12, 266C267.
8. Spellman,P., Sherlock,G., Zhang,M., Iyer,V., Anders,K., Eisen,M., Brown,P., Botstein,D. and Futcher,B. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 1998, Vol.9, 3273C3297.
9. Chen, T., Filkov, V. & Skiena, S. Identifying gene regulatory networks from experimental data.*In Proceedings of the Third Annual International Conference on Research in Computational Molecular Biology* , 1999, 94–103.
10. Yeung,L.K, Szeto, L.K.,Liew,A.W.,& Yan, H. Dominant spectral component analysis for transcriptional regulations using microarray time-series data. *Bioinformatics* , 2004, Vol.20, 742–749.
11. Andrew T. Kwon, Holger H. Hoos abd Raymond Ng, Inference of transcriptional regulation relationships from gene expression data. *Bioinformatics* ,2002, Vol.19(8), 905–912.
12. Karypis, G. GCLUTO - a clustering toolkit, 2002, Available at http://www.cs.umn.edu/gcluto.
13. Hideya Kawaji, Yosuke Yamaguchi, Hideo Matsuda, Akihiro Hashimoto. A Graph-Based Clustering Method for a Large Set of Sequences Using a Graph Partitioning Algorithm. *Genome Informatics*, 2001, Vol.12, 93C102.