Cluster Analysis of Gene Expression Data Based on Self-Splitting and Merging Competitive Learning

Shuanhu Wu, Alan Wee-Chung Liew, *Member, IEEE*, Hong Yan, *Senior Member, IEEE*, and Mengsu Yang

***Abstract*—Cluster analysis of gene expression data from a cDNA microarray is useful for identifying biologically relevant groups of genes. However, finding the natural clusters in the data and estimating the correct number of clusters are still two largely unsolved problems. In this paper, we propose a new clustering framework that is able to address both these problems. By using the one-prototype-take-one-cluster (OPTOC) competitive learning paradigm, the proposed algorithm can find natural clusters in the input data, and the clustering solution is not sensitive to initialization. In order to estimate the number of distinct clusters in the data, we propose a cluster splitting and merging strategy. We have applied the new algorithm to simulated gene expression data for which the correct distribution of genes over clusters is known *a priori*. The results show that the proposed algorithm can find natural clusters and give the correct number of clusters. The algorithm has also been tested on real gene expression changes during yeast cell cycle, for which the fundamental patterns of gene expression and assignment of genes to clusters are well understood from numerous previous studies. Comparative studies with several clustering algorithms illustrate the effectiveness of our method.**

***Index Terms*—cDNA microarrays, cluster splitting and merging, gene expression data analysis, overclustering, self-splitting and merging competitive learning (SSMCL).**

I. Introduction

ADVANCES in the cDNA microarray technology have enabled biologists to monitor thousands of genes simultaneously and measure the whole-genome mRNA abundance in the cellular process under various experimental conditions [1]–[3]. A large amount of gene expression profile data has become available in several databases [4]. The challenge now is to make sense of such massive data sets and this requires the development of powerful data analysis tools.

A crucial step in the analysis of gene expression data is the detection of gene groupings that manifest similar expression patterns. Most current methods for gene expression data analysis rely on the use of clustering algorithms [5]–[8]. The fundamental biological premise underlying these approaches is that genes that display similar expression patterns are coregulated and may share a common function. Although this assumption may be overly simplistic and will not always be true, it has proved to be useful for the exploration of gene expression data [9]–[21].

Although many different clustering algorithms have been used for gene expression data analysis, they all suffer from various shortcomings. For example, hierarchical clustering suffers from robustness, uniqueness, and the inversion problems, which complicate interpretation of the resulting hierarchy [22]. Algorithms based on optimization [12], [13] cannot guarantee that the resulting solution corresponds to the global optimum. K-means methods [5], [6] and self-organizing maps (SOM) algorithms [7] produce clustering results that are strongly dependent on initialization, and there is no guarantee that the resulting clusters are natural clusters.

Recently, a new competitive learning paradigm, called the one-prototype-take-one-cluster (OPTOC), has been proposed [23]. In conventional competitive learning, if the number of clusters is less than the natural clusters in the data, at least one of the prototypes would win data from more than one cluster. In contrast, OPTOC would win data from only one cluster, while ignoring the data from other clusters. The OPTOC-based learning strategy has the following two main advantages: 1) it can find natural clusters, and 2) the final partition of the dataset is not sensitive to initialization.

In this paper, we propose a new clustering framework based on the OPTOC learning paradigm for clustering gene expression data. The new algorithm is able to identify natural clusters in the dataset as well as provides a reliable estimate of the number of distinct clusters in the dataset. This paper is organized as follows. In Section II, we describe the structure of the new clustering algorithm in relation to the general clustering task. In Section III, we provide detailed description of the OPTOC competitive learning paradigm. The overclustering and merging strategy for estimating the number of distinct clusters are described in Sections IV and V, respectively. Experimental results and comparative studies on the clustering of simulated and real gene expression data are provided in Section VI. Finally, Section VII presents the conclusions.

II. A New Clustering Framework

Cluster analysis has proved to be a useful tool in discovering structures and patterns in gene expression data. In general, the goal of cluster analysis is to group data with similar patterns to form distinct clusters. Cluster analysis helps to reduce the complexity of the gene expression data since genes with similar

Manuscript received January 24, 2003; revised October 6, 2003. This work was supported under a CityU SRG Grant (Project 7001183) and under an interdisciplinary research grant (Project 9010003).

S. Wu was with the Department of Computer Engineering and Information Technology, City University of Hong Kong, Kowloon, Hong Kong. He is now with the School of Information, Wu Yi University, Guandong, China.

A. W.-C. Liew is with the Department of Computer Engineering and Information Technology, City University of Hong Kong, Kowloon, Hong Kong (e-mail: itwcliew@cityu.edu.hk).

H. Yan is with the Department of Computer Engineering and Information Technology, City University of Hong Kong, Kowloon, Hong Kong and with the School of Electrical and Information Engineering, University of Sydney, Sydney, NSW 2006, Australia.

M. Yang is with the Department of Biology and Chemistry, City University of Hong Kong, Kowloon, Hong Kong.

Digital Object Identifier 10.1109/TITB.2004.824724

patterns are grouped together. It also aids in the discovery of gene function because genes with similar gene expression profiles can serve as an indicator that they participate in the same or related cellular process.

Given a dataset of N dimension, the goal is to identify groups of data points that aggregate together in some manner in an N-dimensional space. We call these groups "natural clusters." In the Euclidean space, these groups form dense clouds, delineated by regions with sparse data points. Thus, an effective clustering algorithm should be able to: a) identify the natural clusters, and b) estimate the correct number of natural clusters that exist in the dataset.

Most conventional clustering algorithms require the specification of the correct number of clusters in the dataset [6]. Moreover, there is no guarantee that the clusters found correspond to natural clusters in the dataset even if the correct number of clusters is given. In many cases, the clusters obtained by a clustering algorithm depend heavily on the formulation of the objective function of the algorithm. In other words, the algorithm itself imposes an artificial structure on the data. An example is the ellipsoidal structure imposed by the K-means algorithm. The implications of not finding natural clusters are: i) a natural cluster might be erroneously divided into two or more classes, or worst still, ii) several natural clusters or part of them are erroneously grouped into one class. Such behaviors obviously lead to wrong inferences about the data.

In view of the above discussions, we propose a new clustering framework called self-splitting and merging competitive learning clustering (SSMCL). The new algorithm is able to identify the natural clusters through the adoption of a new competitive learning paradigm called the one-prototype-take-one-cluster (OPTOC) method introduced in [23]. The OPTOC learning paradigm allows a cluster prototype to focus on just one natural cluster, while minimizing the competitions from other natural clusters. Since it is very difficult to estimate reliably the correct number of natural clusters in a complex high-dimensional dataset, we adopted an overclustering and merging strategy to estimate the number of *distinct clusters* in the dataset. The overclustering and merging strategy can be viewed as a top-down (divisive clustering), followed by a bottom-up (agglomerative clustering) process. In the top-down step, loose clusters (as measured by their variances) are successively split into two clusters until a prespecific number of clusters (set to be larger than the true number of clusters in the data) are obtained. The overclustering minimizes the chance of missing some natural clusters in the data. The merging step then attempts to merge similar clusters together, until finally all remaining clusters are distinct from each other.

III. The OPTOC Paradigm

In conventional clustering algorithms, if the number of prototypes is less than that of the natural clusters in the dataset, there must be at least one prototype that wins patterns from more than two clusters, and this behavior is called one-prototype-take-multiple-clusters (OPTMC). Fig. 1(a) shows an example of learning based on the OPTMC paradigm, where P1 actually wins all three clusters and finally settles at the center

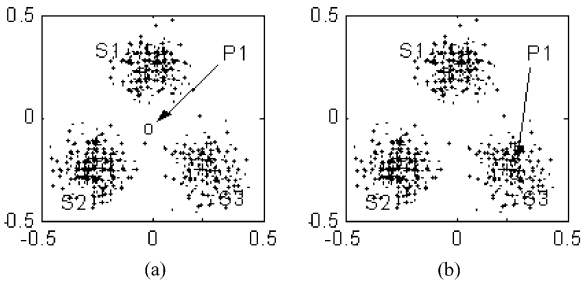

Fig. 1. Two learning methods: OPTMC versus OPTOC. (a) One prototype takes the center of three clusters (OPTMC). (b) One prototype takes one cluster (OPTOC) and ignores the other two clusters.

of clusters S1, S2, and S3. The OPTMC behavior is not desirable in data clustering since we would expect each prototype to characterize only one natural cluster.

In contrast, the OPTOC idea proposed in [23] allows one prototype to characterize only one natural cluster in the dataset, regardless of the number of clusters in the data. This is achieved by constructing a dynamic neighborhood using an online learning vector $\vec{A}_i$, called the asymptotic property vector (APV), for the prototype $\vec{P}_i$, such that patterns inside the neighborhood of $\vec{P}_i$ contribute more to its learning than those outside. Let $|\vec{x}\vec{y}|$ denote the Euclidean distance from $\vec{x}$ to $\vec{y}$, and assume that $\vec{P}_i$ is the winning prototype for the input pattern $\vec{X}$ based on the minimum-distance criterion. The APV $\vec{A}_i$ is updated by

$$\vec{A}_i^* = \vec{A}_i + \left(\vec{X} - \vec{A}_i\right) \bullet \Theta\left(\vec{P}_i, \vec{A}_i, \vec{X}\right) \bullet \frac{\delta_i}{n_{\vec{A}_i}} \tag{1}$$

where Θ is a function given by

$$\Theta\left(\vec{\mu}, \vec{v}, \vec{\omega}\right) = \begin{cases} 1, & \text{if } |\vec{\mu}\vec{v}| \geq |\vec{\mu}\vec{\omega}| \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

and δ_i, within the range $0 < \delta_i \leq 1$, is defined as

$$\delta_i = \left(\frac{|\vec{P}_i\vec{A}_i|}{|\vec{P}_i\vec{A}_i| + |\vec{P}_i\vec{X}|}\right)^2 \tag{3}$$

and $n_{\vec{A}_i}$ is the winning counter which is initialized to zero and is updated as follows:

$$n_{\vec{A}_i} = n_{\vec{A}_i} + \delta_i \bullet \Theta(\vec{P}_i, \vec{A}_i, \vec{X}). \tag{4}$$

The winning prototype $\vec{P}_i$ is then updated by

$$\vec{P}_i^* = \vec{P}_i + \left(\vec{X} - \vec{P}_i\right) \bullet \alpha_i \text{ where } \alpha_i = \left(\frac{|\vec{P}_i\vec{A}_i^*|}{|\vec{P}_i\vec{A}_i^*| + |\vec{P}_i\vec{X}|}\right)^2. \tag{5}$$

If the input pattern $\vec{X}$ is well outside the dynamic neighborhood of $\vec{P}_i$, i.e., $|\vec{P}_i\vec{X}| \gg |\vec{P}_i\vec{A}_i|$, it would have very little influence on the learning of $\vec{P}_i$ since $\alpha_i \to 0$. On the other hand, if $|\vec{P}_i\vec{X}| \ll |\vec{P}_i\vec{A}_i|$, i.e., $\vec{X}$ is well inside the dynamic neighborhood of $\vec{P}_i$, both $\vec{A}_i$ and $\vec{P}_i$ would shift toward $\vec{X}$ according to (1) and (5), and $\vec{P}_i$ would have a large learning rate α_i according to (5). During learning, the neighborhood $|\vec{P}_i\vec{A}_i|$ will decrease monotonically. When $|\vec{P}_i\vec{A}_i|$ is less than a tiny quantity ε, $\vec{P}_i$ would eventually settle at the center of a natural cluster in the input pattern space and the learning stops. Thus, with the help of the APV, each prototype will locate only one natural cluster and ignore other clusters. Fig. 1(b) shows an example of learning

based on the OPTOC paradigm. In this figure, P1 finally settles at the center of S3 and ignores the other two clusters S1 and S2.

IV. Top-Down Step: Cluster Self-Splitting

When the number of clusters in the input space is more than one, additional prototype needs to be generated to search for the remaining clusters. In [23], a procedure is described for the detection of additional natural clusters in the data as follows. Let $\vec{C}_i$ denote the center, i.e., arithmetic mean, of all the patterns that $\vec{P}_i$ wins according to the minimum-distance rule. The distortion $|\vec{P}_i\vec{C}_i|$ measures the discrepancy between the prototype $\vec{P}_i$ found by the OPTOC learning process and the actual cluster structure in the dataset. For example, in Fig. 1(b), $\vec{C}_i$ would be located at the center of the three clusters S1, S2, and S3 (since there is only one prototype, it wins all input patterns), while $\vec{P}_i$ eventually settled at the center of S3. After the prototypes have all settled down, a large $|\vec{P}_i\vec{C}_i|$ indicates the presence of other natural clusters in the data. A new prototype would be generated from the prototype with the largest distortion when this distortion exceeds a certain threshold ξ. Ideally, if a suitable threshold can be given, the cluster splitting process would terminate when all natural clusters in the dataset are found, thus giving the optimum number of clusters. Unfortunately, due to the high dimension and the complex structure exhibited by the gene expression data, the determination of a suitable threshold to find all natural clusters is very difficult, if not impossible, in practice.

In order not to miss any natural cluster in the data, we proposed to instead overcluster the dataset. In overclustering, a natural cluster might be split into more than one cluster. However, no one cluster may contain data from several natural clusters, since the OPTOC paradigm actually discourages a cluster from winning data from more than one natural cluster. In overclustering, the number of clusters is set to be larger than the true number. Then after each OPTOC learning, the cluster with the largest variance is split, until the required number of clusters is reached. On the other hand, if the average variance of the natural clusters in the dataset is approximately known (i.e., by past experience for certain type of data), then a variance threshold smaller than the average variance can be set such that the cluster with the largest variance exceeding this threshold is split, until no further splitting is possible.

When cluster splitting occurs, the new prototype is initialized at the position specified by a distant property vector (DPV) $\vec{R}_i$ associated with the mother prototype $\vec{P}_i$ [23]. The idea is to initialize the new prototype far away from its mother prototype to avoid unnecessary competition between the two. Initially, the DPV is set to be equal to the prototype to which it is associated with. Then, each time a new pattern $\vec{X}$ is presented, the $\vec{R}_i$ of the winning prototype $\vec{P}_i$ is updated as follows:

$$\vec{R}_i^* = \vec{R}_i + \left(\vec{X} - \vec{R}_i\right) \bullet \Theta\left(\vec{P}_i, \vec{X}, \vec{R}_i\right) \bullet \frac{\rho_i}{n_{\vec{R}_i}} \tag{6}$$

where

$$\rho_i = \left(\frac{|\vec{P}_i\vec{X}|}{|\vec{P}_i\vec{R}_i| + |\vec{P}_i\vec{X}|}\right)^2 \tag{7}$$

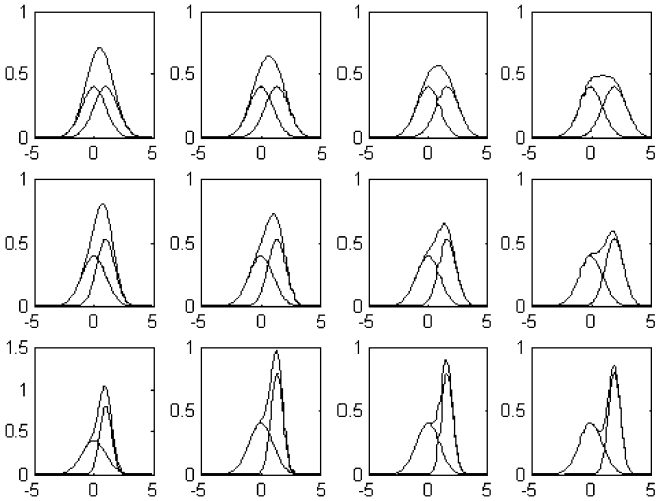

Fig. 2. The joint pdf of two clusters. In the top row, the standard deviations of the two clusters are $\sigma_1 = 1$ and $\sigma_2 = 1$. In the middle row, the standard deviations of the two clusters are $\sigma_1 = 1$ and $\sigma_2 = 0.75$. In the bottom row, the standard deviations of the two clusters are $\sigma_1 = 1$ and $\sigma_2 = 0.5$. Columns 1 to 4 show the change in the joint pdf when the distance between the cluster centers is varied with equal interval from $(\sigma_1 + \sigma_2)/2$ to $2^* \max(\sigma_1, \sigma_2)$.

and $n_{\vec{R}_i}$ is the number of patterns associated with the prototype $\vec{P}_i$.

Note that unlike $\vec{A}_i$, $\vec{R}_i$ always try to move away from $\vec{P}_i$. After a successful split, the property vectors ($\vec{A}_i$, $\vec{R}_i$) of every prototype $\vec{P}_i$ are reset and the OPTOC learning loop is restarted.

V. Bottom-Up Step: Cluster Merging

With overclustering, it is possible that a natural cluster in the dataset would be split into two or more clusters. Thus, some clusters would be visually similar and should be merged together. In this section, we propose a criterion for merging the resulting clusters from the previous overclustering step. The aim of the merging scheme is to produce the final clustering result in which all clusters have distinct patterns. Together with the OPTOC framework, the overclustering and merging framework allow a systematic estimation of the correct number of natural clusters in the dataset.

Our merging scheme is inspired by the observation that a natural cluster should be expected to have a unimodal distribution. Let us assume that the clusters in a dataset have Gaussian distributions, and that the probability density function (pdf) of a distinct cluster is unimodal. If two clusters were well separated, their joint pdf would be bimodal. When two clusters are close to each other to the extent that their joint pdf form a unimodal structure, then it would be reasonable to merge these two clusters into one. Let $\vec{C}_i$ be the center (i.e., mean) of cluster i and σ_i be its standard deviation. We propose that if two clusters satisfy the following condition, they should be merged into one:

$$\left\|\vec{C}_i - \vec{C}_j\right\| \leq \frac{1}{2}(\sigma_i + \sigma_j). \tag{8}$$

In fact, the above merging criterion is somewhat rigorous in our clustering problem. It is reasonable to assume that the ratio of the maximum and minimum standard deviation of two clusters in our clustering application is twofold. In Fig. 2, we give

Over-clustering:

Initialization:

Set number of cluster $K = 1$;
Set $\vec{P}_1 = \vec{R}_1$ at a random location in the input feature space;
Set $\vec{A}_1$ at a random location far from $\vec{P}_1$;
Set the winning counters $n_{\vec{A}_1}$ and $n_{\vec{R}_1}$ to zero;

Learning loop:

Set FINISH = False;
While FINISH = False
OPTOC Learning:
Repeat
1. Randomly read a pattern $\vec{x}$ from the dataset;
2. Find the winner $\vec{P}_i$ where $|\vec{P}_i\vec{x}| = \min_l |\vec{P}_l\vec{x}|$, $l = 1,\ldots,k$. Label $\vec{x}$ with i;
3. Update the Asymptotic Property Vector $\vec{A}_i$ using (1);
4. Update the Distant Property Vector $\vec{R}_i$ using (6);
5. Update the Prototype $\vec{P}_i$ using (5);

Until $\max_l |\vec{P}_l\vec{A}_l| < \varepsilon$ or number if OPTOC iteration exceeds 10.
Split Stage:
If $K <$ maximum number of clusters
1. Find cluster with largest variance, say j;
2. Increment K;
3. Set $\vec{P}_K = \vec{R}_j$;

Reset Stage:
4. For $i = 1 : K$
Set $\vec{R}_i = \vec{P}_i$;
Set $\vec{A}_i$ at a random location far from $\vec{P}_i$;
Set the winning counters $n_{\vec{A}_i}$ and $n_{\vec{R}_i}$ to zero
End For
Else
Set FINISH = True;
End If
End While

Cluster Merging:

Repeat
Find cluster i and cluster j that minimize $\| C_i - C_j \| - 0.5 * (\sigma_i + \sigma_j)$ (see (8));
If $\| C_i - C_j \| \leq 0.5 * (\sigma_i + \sigma_j)$
Merge cluster i and cluster j;
Decrease the number of clusters K by 1;
End If
Until no more clusters can be merged.

Fig. 3. Pseudocode for the proposed SSMCL algorithm.

an illustration of the joint pdf of two clusters. In the top row, the standard deviations of the two clusters are $\sigma_1 = 1$ and $\sigma_2 = 1$. In the middle row, the standard deviations of the two clusters are $\sigma_1 = 1$ and $\sigma_2 = 0.75$. In the bottom row, the standard deviations of two clusters are $\sigma_1 = 1$ and $\sigma_2 = 0.5$. Columns 1 to 4 show the joint pdf when the distance between the centers of the two clusters is varied with equal step from $(\sigma_1 + \sigma_2)/2$ to $2^* \max(\sigma_1, \sigma_2)$. It is apparent that the three joint pdfs in column 1 all appear to be unimodal when the standard deviation ratio changes from $\sigma_i{:}\,\sigma_2 = 1{:}\,1$, to $1 : 0.75$, and finally $1 : 0.5$.

In fact, two clusters having a joint pdf like those in columns 2 and 3 of Fig. 2 also seem to be merged into one cluster. When two clusters are merged into one, the mean and standard deviation of the merged cluster is calculated. Then, the merging process is repeated, until no more clusters can be merged together. The pseudocode for the proposed SSMCL algorithm is shown in Fig. 3.

VI. Results and Discussions

In this section, we verify the performance of the proposed SSMCL algorithm using both simulated and real expression data. We first use simulated gene expression profiles, where the correct solution was known *a priori*, to validate the effectiveness of our algorithm in finding natural clusters and the correct number of clusters. Then we validate the algorithm by clustering the yeast cell cycle data set provided by Cho *et al.* [24] and examine the biological relevance of the clustering results.

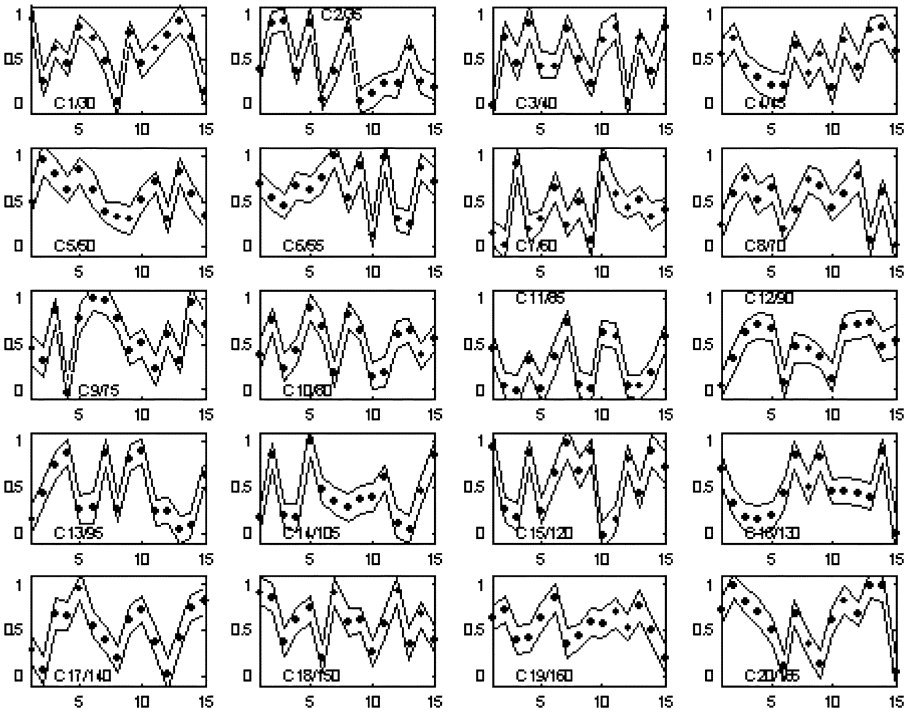

Fig. 4. 1785 randomly generated temporal patterns of gene expression grouped in 20 clusters. Each cluster is represented by the average profile pattern in the cluster (dot line). Solid lines indicate the one standard deviation level of each expression about the mean. Cm/n denotes cluster number m containing n individual profiles. The standard deviation of each dimension in each cluster is equal to 0.15.

A. Clustering Validation: Simulated Gene Expression Data

We randomly generated 20 seed patterns of gene expressions with 15 time points each. Then each pattern was transformed into a cluster by generating many profiles from the pattern. Each cluster contains 30 to 165 profiles, with the total number of profiles in the dataset equal to 1785. Fig. 4 shows the number of profiles in each cluster. For each cluster, the data along each time point k were set to have a standard deviation of σ_k. In our data, we set all σ_k to be equal to 0.15, where the value 0.15 reflects the typical variation along each time point observed in the published expression profiling experiments in [20].

It is well known that many partition-based clustering algorithms are sensitive to initialization, even if the exact number of clusters is known. Poor initialization often results in incorrect partitions, where a natural cluster in the dataset is divided into several clusters, or a cluster in the final clustering result can contain data from several natural clusters. We want to verify that the OPTOC clustering framework can find all the natural clusters in the simulated dataset, independent of initialization. For simplicity, we set the number of iterations of the OPTOC learning to 10, which are generally enough for $|\vec{P}_i\vec{A}_i|$ to converge sufficiently in our applications. Alternatively, since the typical variation along each time point is $\sigma_k = 0.15$ for expression profile data and the pooled variance is given by

$$\left(\sum_{k=1}^{M}\sigma_k^2\right)^{1/2} = \sigma_k\sqrt{M}$$

a reasonable value to use is $\varepsilon = 0.1\sqrt{M}$, where M is the number of time point in the profile. The splitting is stopped when 20 clusters have been generated. Fig. 5 shows the clustering results. We found that the proposed OPTOC-based algorithm was successful in finding all the natural clusters. Moreover, almost all genes were placed into the correct groups, with the exception that one profile in cluster #7 is wrongly grouped into cluster #18, three profiles in cluster #9 are wrongly grouped into cluster #14 and cluster #18, and two profiles in cluster #13 are wrongly grouped into cluster #20.

In the next experiment, we find out whether our overclustering and merging strategy can merge similar clusters and stop at the exact number of clusters automatically, when the exact number of clusters in the data is not known. We set the number of clusters to 28. Fig. 6 shows the resulting 28 clusters before the merging process. After 28 clusters are obtained, cluster merging is performed using the criterion (8). The cluster merging process stopped automatically when exactly 20 clusters were found and the results are shown in Fig. 7. A careful examination of the results showed that only four genes are wrongly clustered: two profiles from cluster #18 are wrongly placed into cluster #1, one profile from cluster #18 is wrongly placed into cluster #20, and one profile from cluster #18 is wrongly placed into cluster #3. Interestingly, the number of misclassified genes is actually less than that in direct clustering of the data into 20 clusters. Further experiments on other randomly generated gene expression profile data also indicated that our algorithm is robust with respect to finding natural clusters and estimating the correct number of clusters.

We also compared our clustering results with the results obtained by using the k-means algorithm. Fig. 8 shows the best clustering results (in terms of the lowest within-class sum-of-square error) by running k-mean 15 times using different initialization. Three and two expression profiles in cluster #18 are wrongly grouped into cluster #1 and cluster #13, respectively.

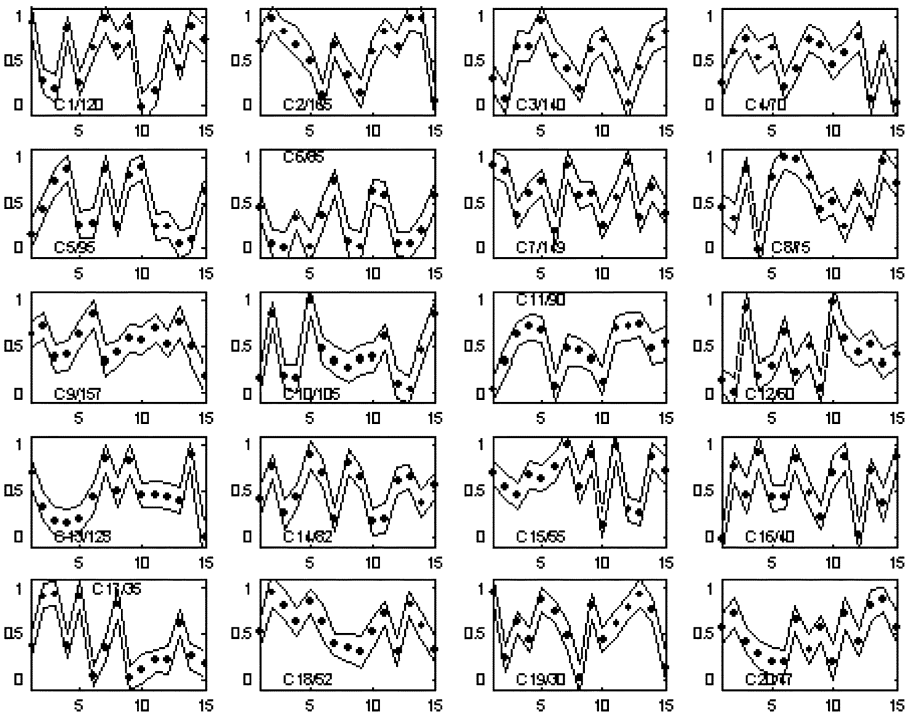

Fig. 5. Clustering results for the 1785 randomly generated temporal patterns of gene expression. The corresponding clusters between the input (in Fig. 4) and output are #1∼ #15, #2 ∼ #20, #3 ∼ #17, #4 ∼ #8, #5 ∼ #13, #6 ∼ #11, #7 ∼ #18, #8 ∼ #9, #9 ∼ #19, #10 ∼ #14, #11 ∼ #12, #12 ∼ #7, #13 ∼ #16, #14 ∼ #10, #15 ∼ #6, #16 ∼ #3, # 17 ∼ # 2, #18 ∼ #5, #19 ∼ #1, and #20 ∼ #4.

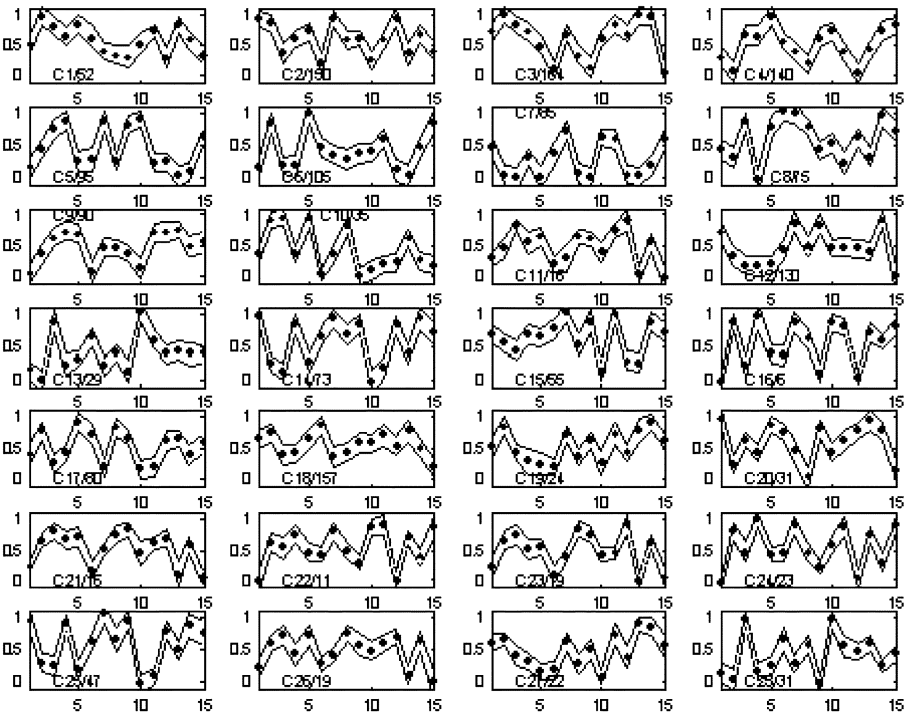

Fig. 6. The clustering results by setting the number of clusters to 28.

Cluster #19 is the combination of cluster #4 and cluster #6 in Fig. 4, while cluster # 17 in Fig. 4 is divided into two clusters, i.e., cluster #7 and cluster #12.

B. Biological Validation: Yeast Cell Cycle

The yeast cell cycle data set has established itself as a standard for the assessment of newly developed clustering algorithm. This data set contains 6601 genes, with 17 time points for each gene taken at 10-min intervals covering nearly two yeast cell cycles (160 min). This data set is very attractive because a large number of genes contained in it are biologically characterized and have been assigned to different phase of the cell cycle.

The raw expression profiles are downloaded from http://genomics.stanford.edu. First, we eliminate those genes whose ex-

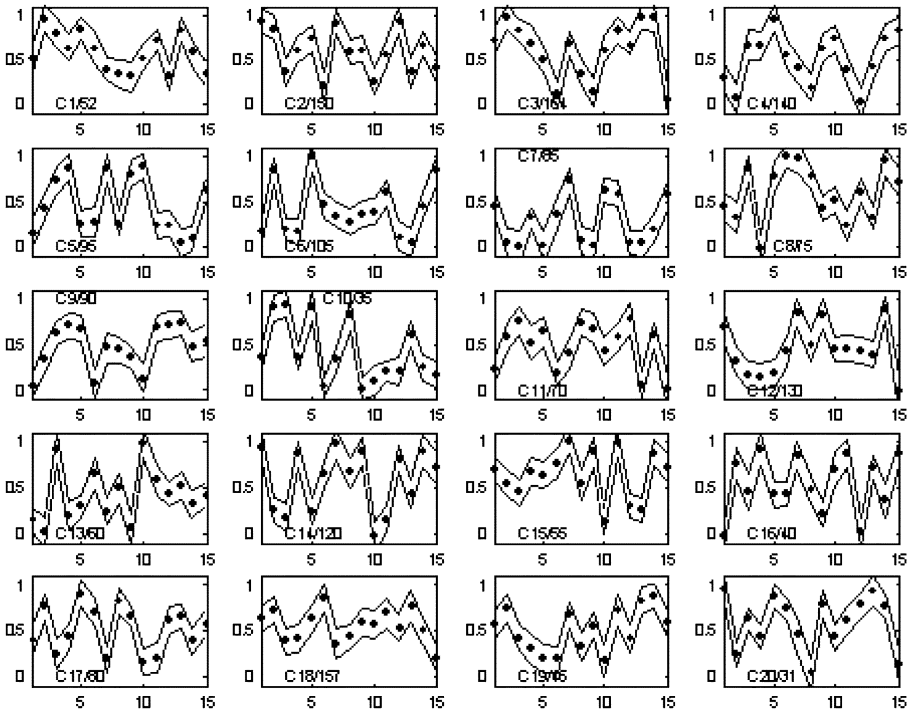

Fig. 7. The final clustering results from the results in Fig. 6 after cluster merging. The corresponding clusters between input (in Fig. 4) and outputs are #1 $\sim$ #5, #2 $\sim$ #18, #3 $\sim$ #20, #4 $\sim$ #17, #5 $\sim$ #13, #6 $\sim$ #14, #7 $\sim$ #11, #8 $\sim$ #9, #9 $\sim$ #12, #10 $\sim$ #2, #11 $\sim$ #8, #12 $\sim$ #16, #13 $\sim$ #7, #14 $\sim$ #15, #15 $\sim$ #6, #16 $\sim$ #3, #17 $\sim$ #10, #18 $\sim$ #19, #19 $\sim$ #4, and #20$\sim$ #1.

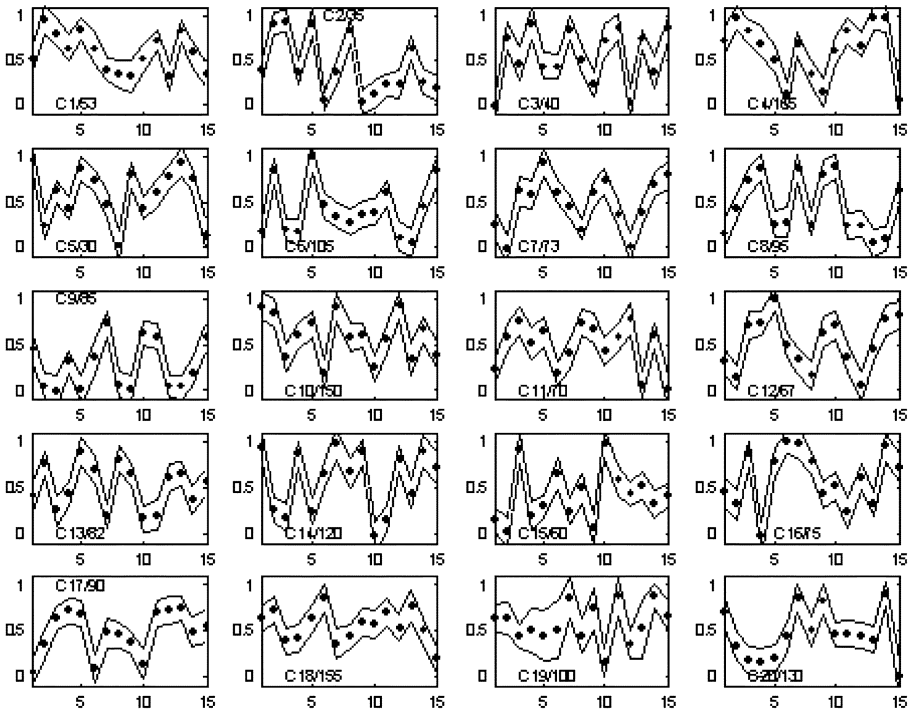

Fig. 8. The best clustering results obtained by running the k-means clustering algorithm 15 times with different initialization. The corresponding clusters between the input (in Fig. 4) and output are #1 $\sim$ #5, #2 $\sim$ #2, #3 $\sim$ #3, #4$\cup$ #6 $\sim$ #19, #5 $\sim$ #1, #7 $\sim$ #15, #8 $\sim$ #11, #9 $\sim$ #16, #10 $\sim$ #13, #11 $\sim$ #9, #12 $\sim$ #17, #13 $\sim$ #8, #14 $\sim$ #6, #15 $\sim$ #14, #16$\sim$ #20, #17 $\sim$ #7 $\cup$ #12, #18 $\sim$ #10, #19 $\sim$ #18, and #20 $\sim$ #4.

pression levels are relatively low and do not show significant changes during the entire time course by a variation filter with the following criteria: a) the value of expression profile at all 17 time points is equal to or greater than 100 (raw data units); b) the ratio of the maximum and the minimum of each time-course expression profiles is at least equal to or greater than 2.5. A total of 1368 gene expression profiles passed the variation filter and were normalized to be between 0 and 1.

The number of OPTOC iterations is set to 10 and the maximum number of clusters is set to 30. Fig. 9 shows the resulting

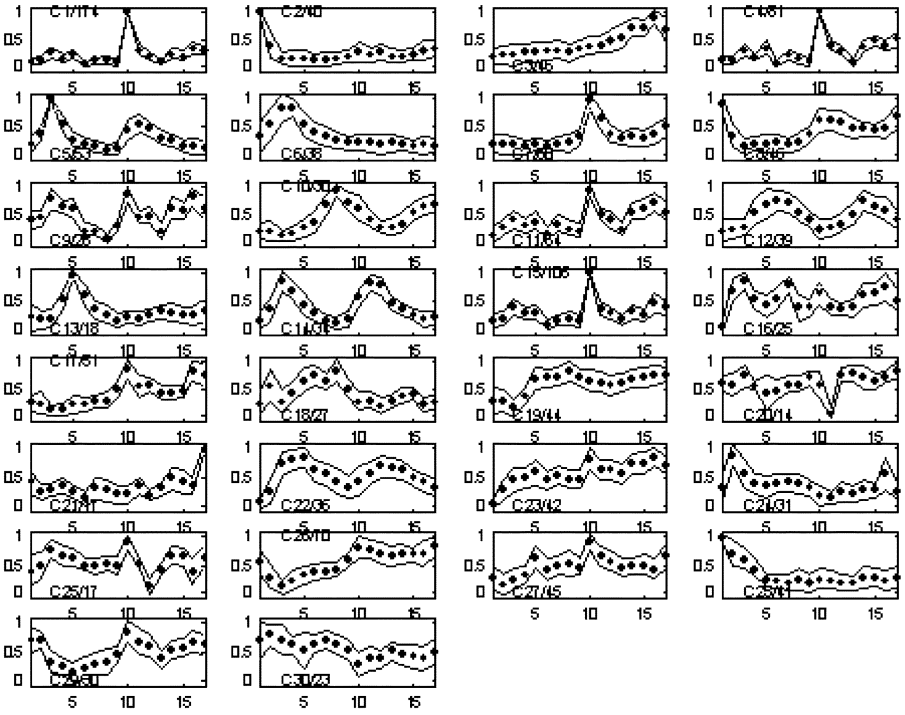

Fig. 9. The clustering results for the yeast cell cycle data. The number of clusters is set to 30.

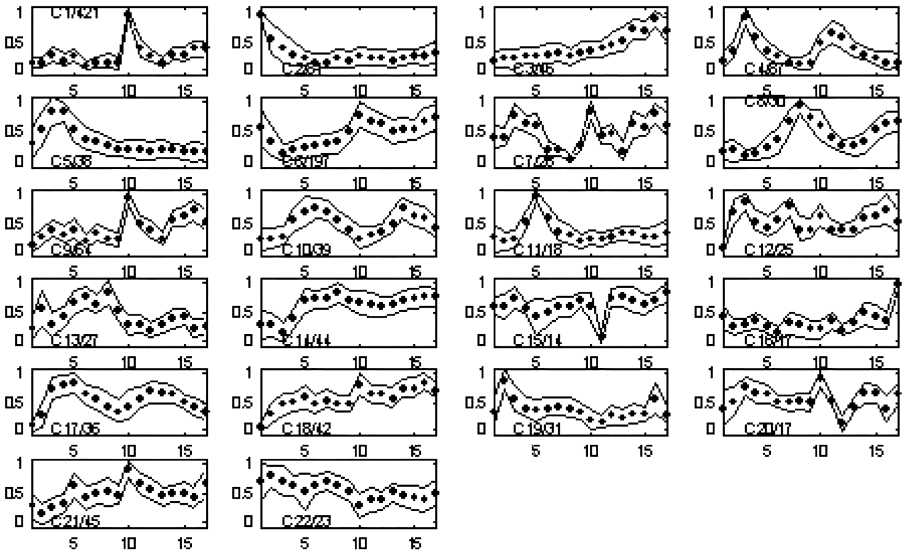

Fig. 10. The final clustering results for the yeast cell cycle data after cluster merging. 22 distinct clusters are obtained.

30 clusters before merging. The process of cluster merging stopped finally at 22 clusters. Fig. 10 shows the merging results. From Fig. 10, we observe that the resulting 22 merged clusters have no apparent visual similarity.

We also checked the resulting 22 clusters to determine whether it could automatically expose known patterns without using prior knowledge. For this purpose, we used gene expression data from the previous study of Cho *et al.* [24], where 416 genes have been interpreted biologically and 110 genes passed our filter. Those gene expression profiles include five fundamental patterns that correspond to five cell cycles phases: early G1, late G1, S, G2, and M phase. In Fig. 11, we show the five clusters that contain most of the genes belonging to these five different patterns. It is obvious that these five clusters correspond to the five cell cycle phases.

For a comparative study, we compared our results with the results obtained by the simulated annealing (SA) based clustering algorithm proposed by Alexander and Rainer [20] and the k-means algorithm. For the SA algorithm, the 1306 genes that passed a variation filter similar to ours were grouped into 20 clusters in which many patterns in the SA clustering output are consistent with ours. For example, clusters #1, #2, #3, #4, #6, #7, #8, #11, #12, #13, #15, #16, #17, #18, #19, and #20 ob-

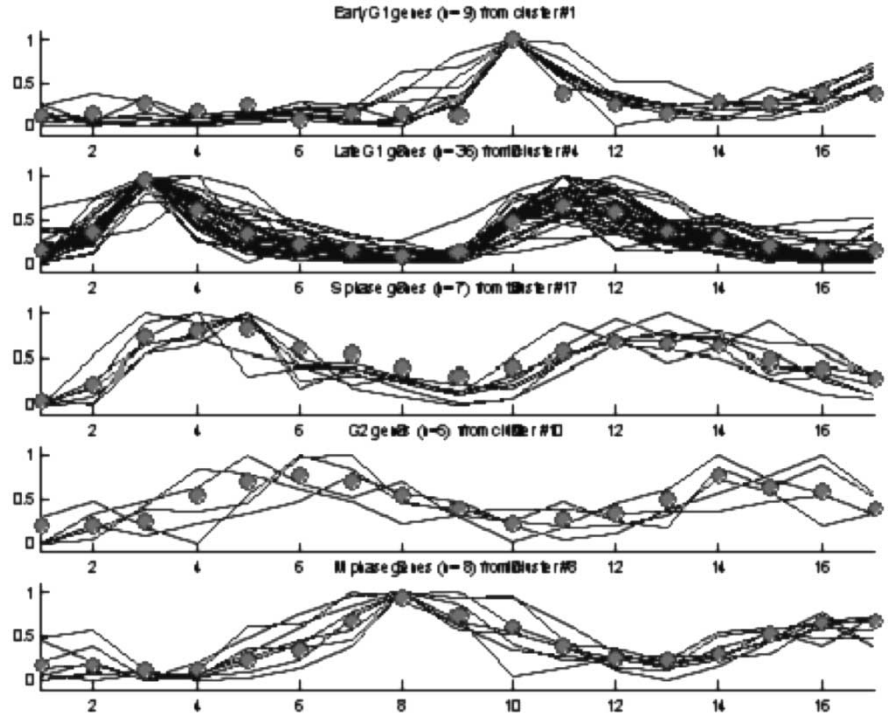

Fig. 11. Five fundamental patterns taken from Fig. 10 that correspond to the five cycle phases. On each subplot, filled cycles represent the average pattern for all profiles in the cluster. The genes presented are only those that belong to this cluster and are biologically characterized and assigned to a specific cell cycle phase.

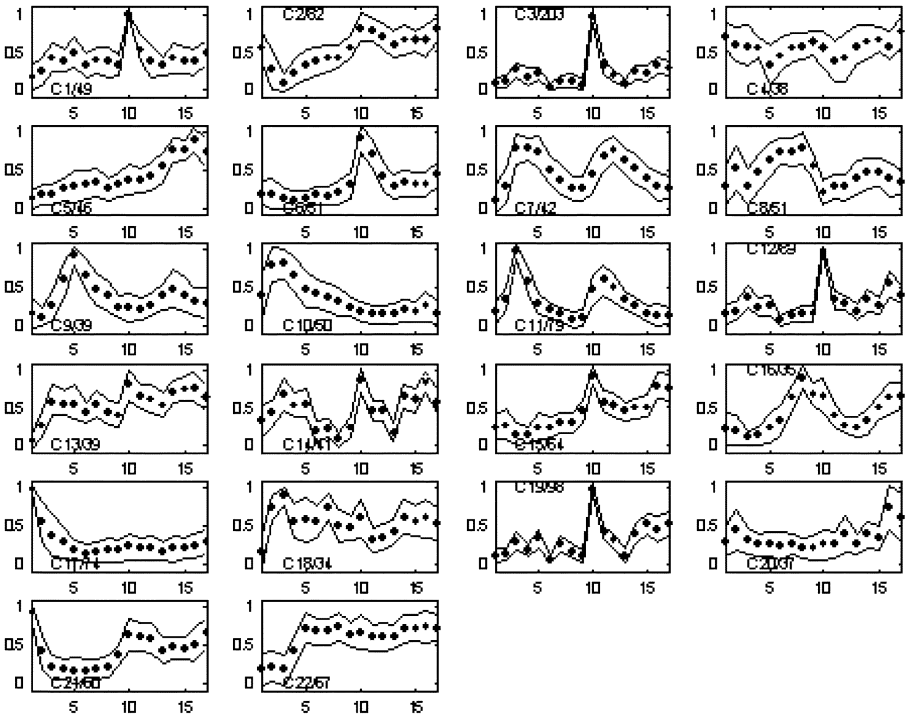

Fig. 12. The best clustering results obtained by running the K-means algorithm 20 times with different initialization for grouping yeast cell cycle expression profile data into 22 clusters. Clusters #3, #12, and #19 are similar visually.

tained by SA in [20, Fig. 5] correspond to clusters #1, #14, #8, #17, #9, #18, #3, #2, #10, #15, #11, #19, #6, #4, #12, and #5 obtained by our algorithm in Fig. 10, respectively. For clusters showing different patterns between the SA clustering method and our method, it is worth mentioning that cluster #9 in [20] has very large variance, and is therefore unlikely to be a natural cluster. For the K-means algorithm, we set the number of clusters to 22 and ran the algorithm with different initialization for 20 times. Fig. 12 shows the best clustering results in terms of the lowest within-class sum-of-square error. It is obvious that clusters #3, #12, and #19 are very similar in appearance and are not distinct clusters.

VII. Conclusion

Cluster analysis is an important tool in gene expression data analysis. An effective clustering algorithm should be able to identify the natural clusters, and to estimate the correct number of clusters in a dataset. In this paper, we have described a new clustering algorithm that can meet those requirements. The ability to find natural clusters in a dataset is based on the OPTOC competitive learning paradigm. The OPTOC paradigm allows one prototype to characterize only one natural cluster in the dataset, regardless of the number of clusters in the data. The OPTOC behavior of a cluster prototype is achieved through the use of a dynamic neighborhood, which causes the prototype to eventually settle at the center of a natural cluster, while ignoring competitions from other clusters. In order to correctly estimate the number of natural clusters in a dataset, we have proposed an overclustering and merging strategy. The overclustering step minimizes the chance of missing any natural clusters in the data, while the merging step ensures that the final clusters are all visually distinct from each other. We have verified the effectiveness of the modified schemes and merging criterion by clustering simulated gene expressions data and real gene expressions profile data for which the biological relevance of the results is known. The results show that the proposed clustering algorithm is an effective tool for gene expression data analysis.

References

[1] D. J. Lockhart and E. A. Winzeler, "Genomics, gene expression and DNA arrays," *Nature*, vol. 405, pp. 827–836, June 2000.

[2] D. Shalon, S. J. Smith, and P. O. Brown, "A DNA microarray system for analyzing complex DNA samples using two-color fluorescent probe hybridization," *Genome Res.*, vol. 6, pp. 639–645, 1996.

[3] R. A. Young, "Biomedical discovery with DNA arrays," *Cell*, vol. 102, pp. 9–15, Jan. 2000.

[4] Some Major Microarray Databases are as follows. [Online]. Available: http://www.ncbi.nlm.nih.gov/geo/ http://genome-www5.stanford.edu/

[5] J. Hartigan, *Clustering Algorithms*. New York: Wiley, 1975.

[6] A. Jain and R. Dubes, *Algorithms for Data Clustering*. Englewood Cliffs, N.J: Prentice-Hall, 1988.

[7] T. Kohonen, *Self-Organizing Maps*. New York\Berlin, Germany: Springer-Verlag, 2001.

[8] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[9] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," *Proc. Nat. Acad. Sci. USA*, vol. 95, pp. 14 863–14 868, Dec. 1998.

[10] X. Wen, S. Fuhrman, G. S. Michaels, D. B. Carr, S. Smith, J. L. Barker, and R. Somogayi, "Large-scale temporal gene expression mapping of central nervous system development," *Proc. Nat. Acad. Sci. USA*, vol. 95, pp. 334–339, 1998.

[11] P. T. Spellman, G. Sherlock, M. O. Zhang, V. R. Iyer, K. Anders, M. B. Eisen, P. O. Brown, D. Botstein, and B. Futcher, "Comprehensive identification of cell cycle-regulated genes of the yeast saccharomyces cerevisiae by microarray hybridization," *Mol. Biol. Cell*, vol. 9, pp. 3273–3297, 1998.

[12] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub, "Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation," *Proc. Nat. Acad. Sci. USA*, vol. 96, pp. 2907–2912, Mar. 1999..

[13] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," *Proc. Nat. Acad. Sci. USA*, vol. 96, pp. 6745–6750, 1999.

[14] C. M. Perou, S. S. Jeffrey, M. Rijn, C. A. Rees, M. B. Eisen, D. T. Ross, A. Pergamenschikov, C. F. Williams, S. X. Zhu, J. C. F. Lee, D. Lashkari, D. Shalon, P. O. Brown, and D. Botstein, "Distinctive gene expression patterns in human mammary epithelial cells and breast cancers," *Proc. Nat. Acad. Sci. USA*, vol. 96, pp. 9212–9217, 1999.

[15] G. Zweiger, "Knowledge discovery in gene-expressionmicroarray data: Mining the information output of the genome," *Trends Biotechnol.*, vol. 17, pp. 429–436, 1999.

[16] K. P. White, S. A. Rifkin, P. Hurban, and D. S. Hogness, "Microarray analysis of drosophila development during metamorphosis," *Science*, vol. 286, pp. 2179–2184, 1999.

[17] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, and D. Haussler, "Knowledge-based analysis of microarray gene expression data by using support vector machines," *Proc. Nat. Acad. Sci. USA*, vol. 97, pp. 262–267, 2000.

[18] C. J. Roberts, B. Nelson, M. J. Marton, R. Stoughton, M. R. Meyer, H. A. Bennett, Y. D. He, H. Dai, W. L. Walker, T. R. Hughes, M. Tyers, C. Boone, and S. H. Friend, "Signaling and circuitry of multiple MAPK pathways revealed by matrix and global gene expression profiles," *Science*, vol. 287, pp. 873–880, 2000.

[19] D. T. Ross, U. Scherf, M. B. Eisen, C. M. Perou, C. Rees, P. Spellman, V. Iyer, S. S. Jeffrey, M. V. D. Rijn, M. Waltham, A. Pergamenschikov, J. C. F. Lee, D. Lashkari, D. Shalon, T. G. Myers, J. N. Weinstein, D. Botstein, and P. O. Brown, "Systematic variation in gene expression patterns in human cancer cell lines," *Nature Genet.*, vol. 24, pp. 227–235, 2000.

[20] A. V. Lukashin and F. Rainer, "Analysis of temporal gene expression profiles: Clustering by simulated annealing and determining the optimal number of clusters," *Bioinformatics*, vol. 17, pp. 405–414, 2001.

[21] L. K. Szeto, A. W. C. Liew, H. Yan, and S. S. Tang, "Gene expression data clustering and visualization based on a binary hierarchical clustering framework, special issue on Biomedical visualization for bioinformatics," *J. Visual Languages Comput.*, vol. 14, pp. 341–362, 2003.

[22] B. J. T. Morgan and A. P. G. Ray, "Non-uniqueness and inversions in cluster analysis," *Appl. Statist.*, vol. 44, pp. 117–134, Jan. 1995.

[23] Y. J. Zhang and Z. Q. Liu, "Self-Splittng competitive learning: A new on-line clustering paradigm," *IEEE Trans. Neural Networks*, vol. 13, pp. 369–380, Mar. 2002.

[24] R. J. Cho, M. J. Campbell, E. A. Winzeler, L. Steinmetz, A. Conway, L. Wodicka, T. G. Wolfsberg, A. E. Gabrielian, D. Landsman, D. J. Lockhart, and R. W. Davis, "A genome-wide transcriptional analysis of the mitotic cell cycle," *Mol. Cell*, vol. 2, pp. 65–73, 1998.

Shuanhu Wu received the B.Sc and M.Sc degrees in computing mathematics in 1986 and 1989, respectively, and the Ph.D. degree in electronic and information engineering in 2001, all from Xi'an Jiaotong University, Xian, China.

From 1989 to 1998, he worked as an engineer at Henan Oil field, Nanyang, China, working on seismic signal processing. From March 2002 to March 2003, he worked as a Research Assistant and Research Fellow in the Department of Computer Engineering and Information Technology, City University of Hong Kong, Hong Kong. He is currently a Lecturer at the Information School at Wu Yi University, Guangdong, China. His current research interests include bioinformatics, image and video coding, signal and image processing, and pattern recognition.

Alan Wee-Chung Liew (M'03) received the B.E. degree with first class honors in electrical and electronic engineering from the University of Auckland, Auckland, New Zealand, in 1993 and the Ph.D. degree in electronic engineering from the University of Tasmania, Australia, in 1997.

He is currently a Senior Research Fellow in the Department of Computer Engineering and Information Technology, City University of Hong Kong. His current research interests include bioinformatics, signal and image processing, pattern recognition, and wavelets.

Hong Yan (S'88–M'89–SM'93) received the B.E. degree from Nanking Institute of Posts and Telecommunications, Nanking, China, in 1982, the M.S.E. degree from the University of Michigan, Ann Arbor, in 1984, and the Ph.D. degree from Yale University, New Haven, CT, in 1989, all in electrical engineering.

During 1982 and 1983, he worked on signal detection and estimation as a graduate student and research assistant at Tsinghua University, Beijing, China. From 1986 to 1989, he was a Research Scientist at General Network Corporation, New Haven, where he worked on design and optimization of computer and telecommunications networks. He joined the University of Sydney, Sydney, Australia, in 1989 and became Professor of Imaging Science in 1997. He is currently also Professor of Computer Engineering at City University of Hong Kong. His research interests include image processing, pattern recognition and bioinformatics. He is author or coauthor of one book and over 200 refereed technical papers in these areas.

Prof. Yan is a Fellow of the International Association for Pattern Recognition (IAPR), a Fellow of the Institution of Engineers, Australia (IEAust), and a member of the International Society for Computational Biology (ISCB).

Mengsu Yang received the B.Sc. degree in chemistry from Xiamen University, Fujiang, China, in 1984, the M.Sc degree in organic chemistry from Simon Fraser University, Burnaby, BC, Canada, in 1989, and the Ph.D degree in analytical chemistry from University of Toronto, Toronto, ON, Canada, in 1993.

He obtained his postdoctoral training in molecular biology in The Scripps Research Institute, San Diego, CA (1993–1994). He joined City University of Hong Kong in 1994 and is currently Professor of Chemistry and Director of the Applied Research Centre for Genomics Technology at City University of Hong Kong. He has published over 90 peer-reviewed scientific papers on the development of novel analytical techniques for biomedical applications and the studies of biomolecular interactions in cellular processes.

Prof. Yang has been recognized for his work by the Best Paper Awards at the Eurasia Chemical Conference (1996) and the Asia-Pacific Conference of Tumor Biology (2001). He was awarded the K. C. Wong Education Foundation Scholar Award in 2003. He is a member of the International Advisory Committee of *The Analyst* (a Royal Society of Chemistry publication) and a Member of the Editorial Board of *Life Science Instruments* (a Chinese Society of Chemistry publication). He holds honorary professorships at The University of Hong Kong and Zhejiang University, China.