Efficiently Mining Gene Expression Data via a Novel Parameterless Clustering Method

Vincent S. Tseng and Ching-Pin Kao

Abstract—Clustering analysis has been an important research topic in the machine learning field due to the wide applications. In recent years, it has even become a valuable and useful tool for in-silico analysis of microarray or gene expression data. Although a number of clustering methods have been proposed, they are confronted with difficulties in meeting the requirements of automation, high quality, and high efficiency at the same time. In this paper, we propose a novel, parameterless and efficient clustering algorithm, namely, Correlation Search Technique (CST), which fits for analysis of gene expression data. The unique feature of CST is it incorporates the validation techniques into the clustering process so that high quality clustering results can be produced on the fly. Through experimental evaluation, CST is shown to outperform other clustering methods greatly in terms of clustering quality, efficiency, and automation on both of synthetic and real data sets.

Index Terms—Machine learning, data mining, clustering, mining methods and algorithms.

1 Introduction

Clustering analysis has been applied in a wide variety of fields such as biology, medicine, psychology, economics, sociology, and astrophysics. The main goal of clustering analysis is to partition a given set of objects into homogeneous groups based on given features, such that objects within a group are more similar to each other and more different from objects that are in other groups [5]. In gene expression analysis, clustering groups the genes into biologically relevant clusters with similar expression patterns so that the genes clustered together tend to be functionally related. Hence, clustering can reveal the coexpression of genes which were uncharacterized previously. In recent years, clustering analysis has even become a valuable and useful tool for in-silico analysis of microarray or gene expression data [2], [3], [7], [17]. For example, Eisen et al. [7] applied a variant of hierarchical clustering to identify groups of coexpressed yeast genes. Alon et al. [2] used a two-way clustering technique to detect groups of correlated genes and tissues. Self-organizing maps were used by Tamayo et al. [17] to identify clusters in the yeast cell cycle data set and human hematopoietic differentiation data set.

Thorough and extensive overviews of clustering algorithms are given in [1] and [11]. Although a number of clustering methods have been proposed [4], [5], [8], [9], [13], [21], they are confronted with three problems. The first problem is that most clustering algorithms request the users to specify some parameters. In real applications, however, it is hard for biologists to determine the suitable parameters manually. Thus, an automated clustering method is required. The second problem is that almost all clustering algorithms aim to produce the clustering results based on the input parameters and their own criterions. Hence, they are incapable of producing the optimal clustering result. The final problem is that the existing clustering algorithms may not perform well when the optimal or near-optimal clustering result is enforced from the universal criterions.

On the other hand, a variety of clustering validation measures were used to evaluate the validity of the clustering results, the suitability of parameters, or the reliability of clustering algorithms. A good overview of clustering validation can be found in [1], in which numerous validation indexes are considered, like DB-index [1], [6], Simple matching coefficient [1], [11], Jaccard coefficient [1], [11], Hubert's Γ statistic [1], [11], etc. There also exist several other measures, like ANOVA [12], FOM [20], VCV [10]. Nevertheless, the roles of them are placed only on the phase of "post-validation." The study on how to integrate validation techniques with clustering methods tightly for improving the clustering quality has been absent.

In this paper, we aim at integrating clustering methods and validation techniques to form a new kind of clustering approach. We propose a novel, parameterless, and efficient clustering algorithm, namely, *Correlation Search Technique (CST)*, which fits for analysis of gene expression data. The primary characteristics of the proposed algorithm are as follows: First, the CST method is a parameterless clustering algorithm. Second, the quality of clustering results generated by CST is "nearly optimal." In [18], Tseng and Chen evaluated a number of validation indexes for measuring the quality of clustering results. It is concluded from the experimental results that Hubert's Γ might be the best index for both partition-based and density-based clustering methods when applied on both low-similarity and high-similarity data sets. So, we use the validation index, namely, Hubert's Γ statistic, as the key measure for clustering quality during the clustering process. The CST algorithm determines how to group the genes on-the-fly by utilizing Hubert's Γ statistic during clustering without any user-input parameter. It is precisely on such grounds that it can generate high quality clustering results automatically and

• *The authors are with the Department of Computer Science and Information Engineering, National Cheng Kung University, No. 1, Ta-Hsueh Road, Tainan, 701, Taiwan, R.O.C.*
E-mail: tsengsm@mail.ncku.edu.tw, zeno@dmlab.csie.ncku.edu.tw.

Manuscript received 8 Sept. 2004; revised 23 Dec. 2004; accepted 14 Apr. 2005; published online 1 Nov. 2005.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-0138-0904.

efficiently. To evaluate the performance of CST, four types of simulated gene expression data sets and one real yeast data set were tested. The experimental results show that the CST method can automatically produce the "nearly optimal" clustering result in much faster speed than other methods like k-means, CAST-FI [19], etc. Meanwhile, the nice feature of automated analysis provided by CST promises a wealth of information that can help the characterization of gene function and the understanding of many other molecular biological processes.

The rest of the paper is organized as follows: In Section 2, we present the idea and algorithm of Correlation Search Technique (CST). The related studies on clustering and validation techniques are described in Section 3. In Section 4, the performance evaluation results are described in detail. Finally, the conclusions are drawn in Section 5.

2 Correlation Search Technique

Before the proposed method, Correlation Search Technique (CST) algorithm, can be applied, we have to generate a similarity matrix S based on the given microarray data set. The matrix S stores the degree of similarity between each pair of genes in the data set, with the degrees in range of [0, 1]. We can obtain the similarity by using any similarity measurements (e.g., Euclidean distance, Pearson's correlation coefficient, etc.) according to the purposes of applications. Then, CST can automatically cluster the genes according to the similarity matrix S without any user-input parameters.

2.1 Preview of CST

The main idea behind CST is to integrate the clustering method with the validation technique so that it can cluster the genes quickly and automatically. Moreover, based on the fact that biologists prefer quick obtaining of suboptimal results to long waiting for optimal results in real applications, CST aims at producing a "near-optimal" clustering result in fast speed. The validation index we used here is Hubert's Γ statistic [1], [11] and its definition is as follows:

Let $\boldsymbol{X} = [X(i,j)]$ and $\boldsymbol{Y} = [Y(i,j)]$ be two $n \times n$ proximity matrices on the same n genes. From the viewpoint of correlation coefficient, $X(i,j)$ indicates the observed correlation coefficient of genes i and j, and $Y(i,j)$ is defined as follows:

$$Y(i,j) = \begin{cases} 1 & \text{if genes } i \text{ and } j \text{ are clustered in the same cluster} \\ 0 & \text{otherwise.} \end{cases} \quad (1)$$

The Hubert's Γ (gamma) statistic represents the point serial correlation between the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$, and it is defined as follows when the two matrices are symmetric:

$$\Gamma = \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{X(i,j)-\overline{X}}{\sigma_X}\right)\left(\frac{Y(i,j)-\overline{Y}}{\sigma_Y}\right), \quad (2)$$

where $M = n(n-1)/2$ is the number of entries in the double sum, and σ_X and σ_Y denote the sample standard deviations, while $\overline{X}$ and $\overline{Y}$ denote the sample means of the entries of matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$.

From the viewpoint of distance, $X(i,j)$ indicates the observed distance of genes i and j, and $Y(i,j)$ is defined as (1) by exchange the "1" and "0." The value of Γ is between [-1, 1] and a higher value of Γ represents the better clustering quality.

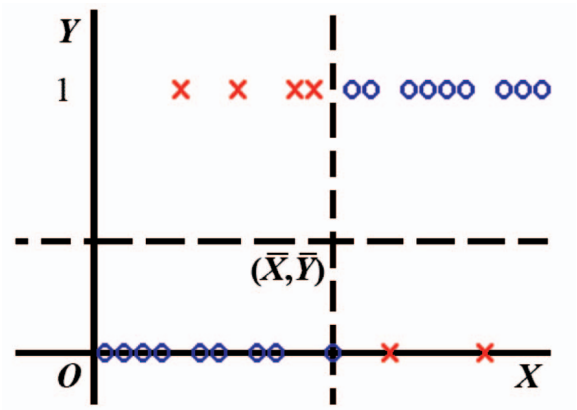


Fig. 1. The concept of Hubert's Γ statistic.

Fig. 1 illustrates the concept of Hubert's Γ statistic. For matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ as described above, the solid lines represent the original coordinate and the dotted lines represent the coordinate centered by $(\overline{X}, \overline{Y})$. A point $A_{i,j}$ that falls in quadrant I of dotted coordinate indicates that genes i and j have higher similarity and are clustered in the same cluster. A point $B_{i,j}$ that falls in quadrant II indicates that genes i and j have lower similarity, but are clustered in the same cluster. In contrast, a point $C_{i,j}$ that falls in quadrant III of dotted coordinate indicates that genes i and j have lower similarity and are clustered in different clusters. A point $D_{i,j}$ that falls in quadrant IV indicates that genes i and j have higher similarity, but are clustered in different clusters. In brief, given a clustering result, the more points that fall in quadrants I and III, the better the quality the clustering bears. Therefore, the point serial correlation between the matrices $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be used to measure the quality of clustering results.

Since the computing of Γ statistic is time-consuming, we simplify it as follows to enhance the performance of our approach:

$$\begin{aligned}\Gamma &= \frac{1}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(\frac{X(i,j)-\overline{X}}{\sigma_X}\right)\left(\frac{Y(i,j)-\overline{Y}}{\sigma_Y}\right) \\ &= \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(X(i,j)-\overline{X}\right)\left(Y(i,j)-\overline{Y}\right)}{M\sqrt{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{\left(X(i,j)-\overline{X}\right)^2}{M}}\sqrt{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\frac{\left(Y(i,j)-\overline{Y}\right)^2}{M}}} \\ &= \frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(X(i,j)-\overline{X}\right)\left(Y(i,j)-\overline{Y}\right)}{\sqrt{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(X(i,j)-\overline{X}\right)^2}\sqrt{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\left(Y(i,j)-\overline{Y}\right)^2}}.\end{aligned} \quad (3)$$

Recall that we aim to obtain "nearly optimal" clustering results rather than the optimal one in this work. To reduce the complexity, we estimate the standard deviation (Sigma) for Y in (3) from Binomial instead of Normal distribution, although there could be some minor loss in the accuracy. Furthermore, the numerator in (3) can be rewritten by expanded form as

$$\begin{aligned}
&\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\big(X(i,j)-\overline{X}\big)\big(Y(i,j)-\overline{Y}\big)\\
&=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\big(X(i,j)Y(i,j)-\overline{X}Y(i,j)-\overline{Y}X(i,j)+\overline{XY}\big)\\
&=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)Y(i,j)-\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)\\
&\quad-\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)}{M}\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)\\
&\quad+M\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)}{M}\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)}{M}\\
&=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)Y(i,j)-\frac{\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)}{M},
\end{aligned}\tag{4}$$

and parts of the denominator of (3) can be rewritten by expanded form as

$$\begin{aligned}
&\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}\big(G(i,j)-\overline{G}\big)^2\\
&=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}G(i,j)^2-\frac{\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}G(i,j)\right)^2}{M},G=X,Y.
\end{aligned}\tag{5}$$

By substituting (4) and (5) into (3), we have an expanded form of Hubert's Γ statistic as follows:

$$\Gamma=\frac{\left(M\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)Y(i,j)-\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)\right)}{\sqrt{M\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)^2-\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)\right)^2}\sqrt{M\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)^2-\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)\right)^2}}.\tag{6}$$

Let

$$P=\sqrt{M\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)^2-\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)\right)^2}.\tag{7}$$

In fact, we can do the computation of P as soon as the similarity matrix $\boldsymbol{X}$ is given. The value of $Y(i,j)$ is either zero or one, so the square of $Y(i,j)$ is equal to $Y(i,j)$. Hence, (6) can be simplified as

$$\Gamma=\frac{M\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)Y(i,j)-\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)}{P\sqrt{M\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)-\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)\right)^2}}.\tag{8}$$

Furthermore, the computation of P for the same data set is invariable, whether the clustering result is variable. P may be abridged while we choose the best clustering result. Accordingly, the measurement Γ' for the quality of clustering result is

$$\Gamma'=\frac{M\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)Y(i,j)-\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)}{\sqrt{M\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)-\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)\right)^2}}.\tag{9}$$

Finally, we refer to (9) as a *simplistic Hubert's Γ statistic*.

2.2 CST Algorithm

The input of CST is a symmetric similarity matrix $\boldsymbol{X}$, where $X(i,j)\in[0,1]$. CST is a greedy algorithm that constructs clusters one at a time. The currently constructed cluster is denoted by C_{open}. Each cluster is started by a seed and constructed incrementally by adding (or removing) elements to (or from) C_{open} one at a time. The temporary clustering result of each addition (or removal) of x is computed by simplistic Hubert's Γ statistic, i.e., (9), and is denoted by $\Gamma_{\text{add}}(x)$ (or $\Gamma_{\text{remove}}(x)$). In addition, the current maximum of simplistic Hubert's Γ statistic is denoted by $\Gamma_{\max}$. We say that an element x has high positive correlation if $\Gamma_{\text{add}}(x)\geq\Gamma_{\max}$, and x has high negative correlation if $\Gamma_{\text{remove}}(x)\geq\Gamma_{\max}$. CST takes turns between adding high positive correlation elements to C_{open} and removing high negative correlation elements from it. When C_{open} is stabilized by addition and removal procedure, this cluster is complete and the next one is started.

The addition and removal procedures strengthen the quality of C_{open} gradually. Moreover, the removal procedure exterminates the cluster members that were inaccurately added at early clustering stages. In addition, a heuristics is added to CST for selecting an element with the maximum number of neighbors to start a new cluster.

To save computation time, we simplify the simplistic Hubert's Γ statistic, i.e., (9), further. Let

$$S_Y=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j),\tag{10}$$

$$Q=\sqrt{M\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)-\left(\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j)\right)^2},\tag{11}$$

and

$$R=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}X(i,j)\sum_{i=1}^{n-1}\sum_{j=i+1}^{n}Y(i,j).\tag{12}$$

For each addition (or removal) stage, the computation of S_Y will be the same, no matter which element is chosen. Consequently, the computations of Q and R may be abridged while we are deciding which element should be added to (or removed from) C_{open} and the measurement Γ'' of effect of each added (or removed) element is

Input: An n-by-n similarity matrix X

0. Initialization:
   $M = n\,(n - 1) / 2$
   $S_X = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X(i, j)$
   $S_Y = 0$
   $S_{XY} = 0$
   $C = \emptyset$ /* The collection of closed clusters */
   $U = \{1, 2, \ldots, n\}$ /* Elements not yet assigned to any cluster */
   $\Gamma_{\max} = 0$
1. **while** $(U \neq \emptyset)$ **do**
   $C_{open} = \emptyset$
   $a(\bullet) = 0$
   1.1. **SEED:** Pick an element $u \in U$ with most neighbors
      $U = U - \{u\}$ /* Remove u from U */
      For all $i \in U$ set $a(i) = X(u, i)$ /* Update the affinity*/
      $C_{open} = \{u\}$ /* Insert u into C_{open} */
   1.2. **ADD: while** MaxValidaty( ) $\geq \Gamma_{\max}$ **do**
      Pick an element $u \in U$ with maximum $a(\bullet)$
      $U = U - \{u\}$ /* Remove u from U */
      $S_Y = S_Y + |C_{open}|$
      $S_{XY} = S_{XY} + a(u)$
      For all $i \in U \cup C_{open}$ set $a(i) = a(i) + X(u, i)$ /* Update the affinity*/
      $C_{open} = C_{open} \cup \{u\}$ /* Insert u into C_{open} */
      $\Gamma_{\max}$ = MaxValidaty( )
   1.3. **REMOVE: while** MaxValidaty( ) $> \Gamma_{\max}$ **do**
      Pick an element $v \in C_{open}$ with minimum $a(\bullet)$
      $C_{open} = C_{open} - \{v\}$ /* Remove v from C_{open} */
      $S_Y = S_Y - |C_{open}|$
      $S_{XY} = S_{XY} - a(u)$
      For all $i \in U \cup C_{open}$ set $a(i) = a(i) - X(u, i)$ /* Update the affinity*/
      $U = U \cup \{v\}$ /* Insert v into U */
      $\Gamma_{\max}$ = MaxValidaty()
   1.4. Repeat steps ADD and REMOVE as long as there are no elements been removed.
   1.5. $C = C \cup \{C_{open}\}$
   end
2. Done, return the collection of cluster, C.

Fig. 2. The pseudocode of CST.

$$\Gamma'' = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} X(i,j)Y(i,j). \tag{13}$$

The pseudocode of CST is shown in Fig. 2. The subroutine MaxValidaty(•) computes the possible maximum of measurement Γ', yet, i.e., (9), while a certain element is added (or removed). In the addition stage, it is equal to

$$\frac{\left(M * (S_{XY} + \max\{a(u) | u \in U\}) - S_X * (S_Y + |C_{open}|)\right)}{\sqrt{M * (S_Y + |C_{open}|) - (S_Y + |C_{open}|)^2}}. \tag{14}$$

For the removal stage, it becomes:

$$\frac{\left(M * (S_{XY} - \min\{a(v) | v \in C_{open}\}) - S_X * (S_Y - |C_{open}| + 1)\right)}{\sqrt{M * (S_Y - |C_{open}| + 1) - (S_Y - |C_{open}| + 1)^2}}. \tag{15}$$

3 Related Work

The related work of clustering analysis can be divided into three categories [1]. The first category is on similarity measurements that may affect the final clustering results directly. Euclidean distance and correlation coefficient are most popular similarity measures. Although this is an important task for clustering analysis, it is not the focus of this research. The second category is on the clustering methods, which are the core of clustering analysis and have received extensive attentions. The third category is on validation techniques that are applied to evaluate the validity of the clustering results, the suitability of parameters, or the reliability of clustering algorithms. In addition, all validation techniques may be roughly divided into two main types: scalar measurements and intensity image methods [10]. We will introduce several clustering methods and validation techniques in the following sections.

TABLE 1
Association Table

		Matrix U	
		1	0
Matrix V	1	a	b
	0	c	d

3.1 Clustering Methods

The most well-known clustering method is probably the k-means algorithm [1], [11], which is a partitioning-based method. It assumes that the number of clusters is known a priori. Hierarchical methods, for example, UPGMA [14], BIRCH [21], CURE [8], and ROCK [9], are another popular kind of clustering methods. Agglomerative or divisive are used to represent distance matrices as ultrametric trees. In recent years, neural networks, such as self-organizing maps [13], competitive learning networks [15], and adaptive resonance theory (ART) networks [4], have also been used for clustering. In [19], we proposed a novel clustering methods that integrates efficient clustering algorithms with a validation technique such that the quality of results can be evaluated on-the-fly.

3.2 Scalar Measurements

There are two usual kinds of scalar measurements. One kind is external indices and the other is internal indices. External indices are used to measure the extent to which cluster labels (a clustering result) match externally supplied class labels (a partition that is known a priori). In contrast, internal indices are used to measure the goodness of a clustering structure without the external information. Notice that Hubert's Γ statistic stated in Section 2 is an internal index. In the following sections, we will introduce two more popular external indices, namely, *Simple matching coefficient* and *Jaccard coefficient* [1], [11].

Let $\boldsymbol{U} = [U(i,j)]$ and $\boldsymbol{V} = [V(i,j)]$ be two $n \times n$ binary matrices on the same n data items. Let these matrices represent two distinct clustering results with the general form defined as follows:

$$G(i,j) = \begin{cases} 1 & \text{if genes } i \text{ and } j \text{ are clustered in the same cluster} \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

Let us consider the contingency table of the matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ as shown in Table 1. Here, a indicates the number of entries on which both $\boldsymbol{U}$ and $\boldsymbol{V}$ have values "1," and b indicates the number of entries on which $\boldsymbol{U}$ have values "1" and $\boldsymbol{V}$ have values "0," and so on.

The Simple matching coefficient is defined by

$$S = \frac{a+d}{a+b+c+d}, \tag{17}$$

which is the total number of matching entries divided by total number of entries. The Jaccard coefficient is defined by

$$S = \frac{a}{a+b+c}, \tag{18}$$

which is similar to the matching coefficient, except that the "negative" matches (d) are ignored. The Simple matching coefficient usually varies over a smaller range than the Jaccard coefficient because "negative" matches are often a dominant factor.

3.3 Intensity Image Methods

One main problem of scalar validity measures is that representing the accuracy of different clustering methods by a single real number may lose much information. Accordingly, intensity image methods use all of the information generated by the clustering methods to produce an $n \times n$ gray-scale intensity image. It converts the similarity (or dissimilarity) into gray scales and reorders the data of similarity matrix for displaying the clustering result. For instance, the VCV [10] approach retains and organizes the information that is lost through the massive aggregation of information by scalar measures.

4 Experimental Evaluation

In order to evaluate the performance and accuracy of the CST method, we need some gene expression data sets in which the cluster structures are known a priori. Accordingly, we tested the CST method on four synthetic data sets generated by a correlation-inclined cluster data set generator. Meanwhile, a real data set, namely, the yeast cell-cycle data set by Spellman et al. [16], was also used for testing CST. In the synthetic data generator, the users can set up some parameters for generating various kinds of gene expression data sets with variations in terms of the number of clusters, the number of genes in each cluster, the number of outliers (or noise), etc. First, a number of seed genes are produced by the generator or input by the user. All the seed genes must have same dimensions (number of conditions) with very low similarity mutually so as to ensure that the generated clusters will not overlap. Second, the generator produces clusters one by one. It randomly produces genes in each cluster and all produced genes are similar to the seed gene of this cluster in the viewpoint of correlation. Third, the outliers are produced randomly and added into the data set. It should be noted that the outliers may fall into some clusters. In this way, all genes in the same cluster will have very high similarity and they will have high dissimilarity with genes in other clusters.

We first generate four seed sets with size three, five, six, and ten, respectively. The seeds in each set have 15 dimensions and their correlation coefficient is less than 0.1. The profiles of these four seed sets are shown in Fig. 3. Then, we load these four seed sets into our data set generator to generate four synthetic gene expression data sets for testing, namely, Data Set I, Data Set II, Data Set III, and Data Set IV, respectively. Data Set I contains three gene clusters of sizes 900, 700, and 500, with additional 400 outliers. Data Set II contains six gene clusters of sizes 500, 450, 400, 300, 250, and 200, with additional 400 outliers. Data Set III contains five gene clusters of sizes 1200, 1000, 800, 700, and 500, with 800 additional outliers. Data Set IV contains 10 gene clusters of sizes 650, 550, 550, 500, 450, 400, 350, 300, 250, and 200, with 800 additional outliers. Table 2 shows the cluster

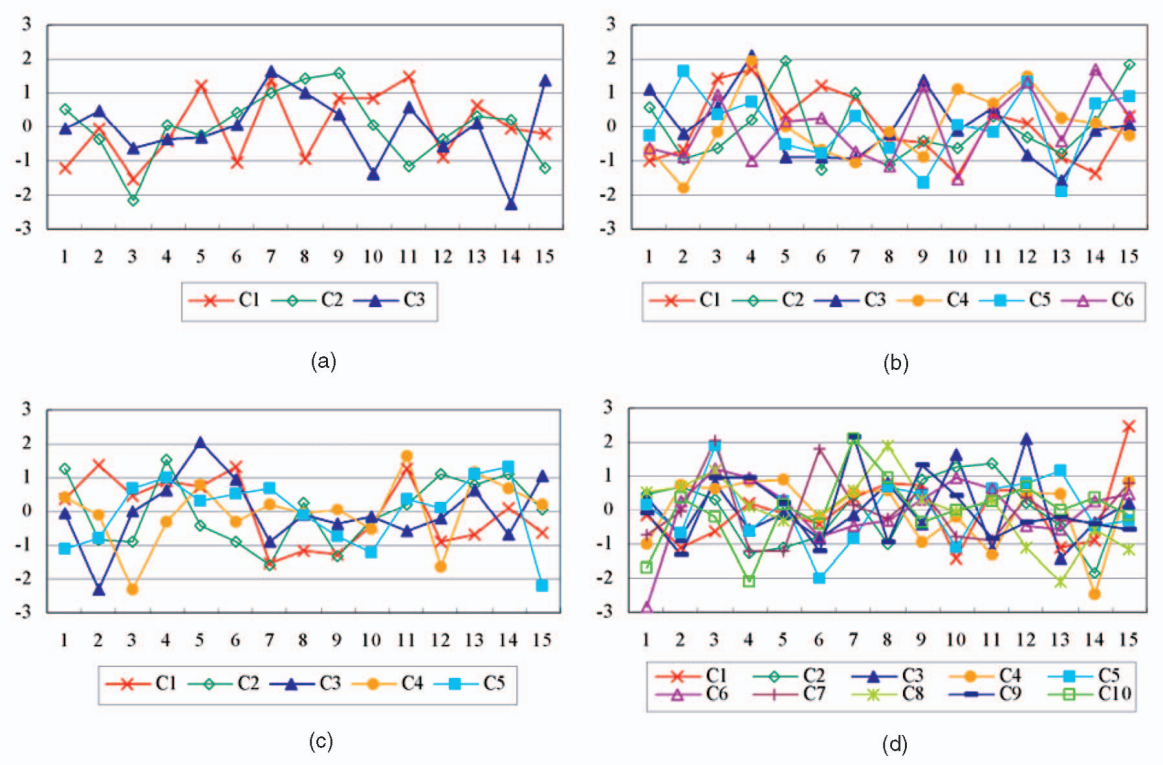


Fig. 3. Profiles of the four seed sets. (a) Data Set I. (b) Data Set II. (c) Data Set III. (d) Data Set IV.

TABLE 2
Cluster Structure of Simulated Data Sets

	Dataset I	Dataset II	Dataset III	Dataset IV
# Clusters	3	6	5	10
Cluster Size	900, 700, 500.	500, 450, 400, 300, 250, 200.	1200, 1000, 800, 700, 500.	650, 550, 550, 500, 450, 400, 350, 300, 250, 200.
# Outliers	400	400	800	800
# Total Patterns	2500	2500	5000	5000

structures of these four data sets. Fig. 3 shows the pattern profiles of each cluster in Data Set I.

For the real data set, the yeast cell-cycle data set contains 72 expressions, including alpha factor, cdc15, cdc28, and elutriation. Spellman et al. [16] used hierarchical clustering and analysis of the 5′ regions of the genes to identify eight groups, namely, CLN2 (76 genes), Y′ (31 genes), Histone (9 genes), MET (20 genes), CLB2 (35 genes), MCM (34 genes), SIC1 (27 genes), and MAT (13 genes). Since several groups are similar in terms of the expression profiles, the eight groups are further grouped into four main clusters according to the hierarchical structure of Spellman's analysis, namely, G1 (CLN2 and Y′ groups), S/G2 (Histone and MET groups), M (CLB2 and MCM groups), and M/G1 (SCI1 and MAT groups). Since we would like to cluster the genes with main considerations on the expression profiles, we take the G1, S/G2, M, and M/G1 groups as the standard for the clustering results.

We compare the proposed method CST with the well-known clustering methods, namely, k-means [1], [10], CAST-FI [19] and the efficient method we proposed in [19] (we name it Smart-CAST here). For k-means, the value of k is varied from 2 to 21 in step of 1, and from 2 to 41 in step of 1, respectively. For CAST-FI, the value of affinity threshold t is varied from 0.05 to 0.95 in fixed increment of 0.05. The quality of clustering results was measured by using Hubert's Γ statistic, Simple matching coefficient [1], [11], and Jaccard coefficient [1], [11]. Simple matching coefficient and Jaccard coefficient can evaluate the quality of clustering results in preclustering data set. We also use an intensity image to exhibit the visualized results of the clustering methods. Since correlation coefficient is widely adopted in most studies on gene expression analysis, we evaluate only correlation coefficient for the similarity measure. The readers are referred to [19] for investigating the effect of other similarity measures like Euclidean distance on clustering methods like k-means and CAST.

4.1 Simulated Data Set I and Data Set II

Table 3 and Table 4 show the total execution time and the best clustering quality of the tested methods on Data Set I and Data Set II, respectively. The notation "M" indicates the

TABLE 3
Experimental Results (Data Set I)

Methods	Time (s)	# Clusters	Γ Statistic	Matching Coefficient	Jaccard Coefficient
CST	< 1	65 (M=3)	0.800	0.981	0.926
Smart CAST	12	85 (M=3)	0.800	0.986	0.944
CAST-FI	54	91 (M=3)	0.799	0.986	0.945
k-means (k=2-21)	396	6 (M=6)	0.456	0.825	0.427
k-means (k=2-41)	1638	6 (M=6)	0.456	0.825	0.427

number of main clusters, and we take clusters whose size smaller than 50 and 5 as outliers for simulated data sets and the yeast cell-cycle data set, respectively.

It is observed that CST, Smart-CAST, and CAST-FI outperform k-means substantially in both of execution time and clustering quality in terms of validation measures (i.e., Hubert's Γ statistic, Simple matching coefficient and Jaccard coefficient). In particular, CST performs 396 times to 1,638 times faster than k-means on Data Set I and 409 times to 1,540 times faster on Data Set II, respectively. In addition, the results also show that the clustering quality generated by CST is very close to that of Smart-CAST and CAST-FI for all of Hubert's Γ statistic, Simple matching coefficient, and Jaccard coefficient. It means that the clustering quality of CST is as good as Smart-CAST and CAST-FI, even though the execution of CST is substantially faster than all other methods.

Table 3 also shows that k-means produced six main clusters as the best clustering result for Data Set I, with the sizes of clusters being 580, 547, 457, 450, 352, and 114. This is true for both cases that k is varied from 2 to 21 and from 2 to 41. This clustering result does not match the original cluster structure. In contrast, CST produces 65 clusters as the best clustering result. In particular, it is clear that three main clusters are generated, with sizes of 912, 717, and 503. This tightly matches the original cluster structure. The same observation applies to Data Set II. Moreover, CST also generates quite a number of clusters with small size, which are mostly outliers. This means that CST is superior to k-means in filtering out the outliers from the main clusters.

The intensity images of the original cluster structures and the clustering results for Data Set I and Data Set II are shown in Fig. 4 and Fig. 5, respectively. From Fig. 4a, we can easily observe that there are three main clusters and quite a number of outliers in Data Set I. It is observed that Fig. 4c and Fig. 4d are very similar to Fig. 4a, meaning that the clustering results of CST and Smart-CAST are very similar to the original cluster structure in Data Set I. The same observation applies to Data Set II. This can provide more accurate clustering results and insight for gene expression analysis.

4.2 Simulated Data Set III and Data Set IV

We conducted the same experiments by replacing Data Set I and Data Set II with Data Set III and Data Set IV. The experimental results of the tested methods are presented in Table 5 and Table 6. It is obvious that the observation on Table 5 for Data Set III is similar to that on Data Set I and Data Set II. However, Table 6 shows quite different results. The k-means algorithm with k ranged as 2 to 21 produced 21 main clusters as the best clustering result for Data Set IV. There exist eight clusters sized between 51 and 200, 11 clusters sized between 201 and 400, and two clusters sized between 401 and 600. The k-means algorithm with k ranged as 2 to 41, however, produced 24 main clusters as the best clustering result. There exist 15 clusters sized between 51 and 200, and 9 clusters sized between 201 and 400.

Fig. 6 shows the quality of clustering results for k-means with k varied from 2 to 41 on Data Set IV. It is obvious that k-means is an unstable algorithm by observing from the figure that Hubert's Γ statistic versus values of k is not a parabolic curve. Since the range of the value of k is too wide, we cannot apply the automation technique as in Smart-CAST for k-means. We also observe that the shape of

TABLE 4
Experimental Results (Data Set II)

Methods	Time (s)	# Clusters	Γ Statistic	Matching Coefficient	Jaccard Coefficient
CST	< 1	58 (M=6)	0.711	0.977	0.841
Smart CAST	13	86 (M=6)	0.713	0.985	0.888
CAST-FI	55	98 (M=6)	0.712	0.986	0.898
k-means (k=2-21)	409	20 (M=13)	0.388	0.899	0.306
k-means (k=2-41)	1540	20 (M=13)	0.388	0.899	0.306

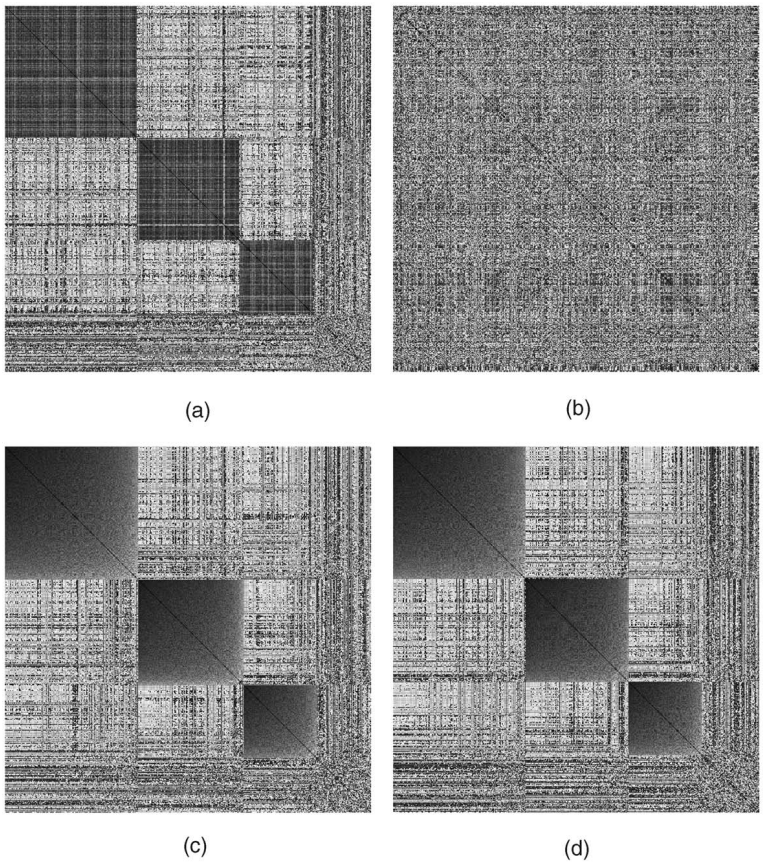


Fig. 4. Intensity images on Data Set I. (a) The original cluster structure. (b) The actual input to the clustering algorithms. (c) Clustering result of CST. (d) Clustering result of CAST.

Γ statistic is very similar to that of the Jaccord coefficient, but the shape of the Simple matching coefficient is approximately a monotone incremental curve. The Jaccard coefficient is often more sensitive than the Simple matching coefficient because "negative" matches are often a dominant factor and are ignored. We praise Hubert's Γ statistic as well as the Jaccard coefficient.

Let us now return to the discussion on Table 6. The clustering result of CST has lower values in Γ statistic and Jaccard coefficient than Smart-CAST and CAST-FI do. This is because CST produces less number of clusters and the clusters slightly absorb more outliers. However, the cluster structure generated by CST is actually very similar to the original data set still. Furthermore, CST outperforms other methods greatly in the execution time. In overall, the empirical evaluation results show that CST outperforms k-means substantially in both of efficiency and quality, and it outperforms Smart-CAST and CAST-FI greatly in efficiency with comparable clustering quality.

The intensity images of the original cluster structure and the clustering results on Data Set III and Data Set IV are shown in Fig. 7 and Fig. 8, respectively. It is obvious that the

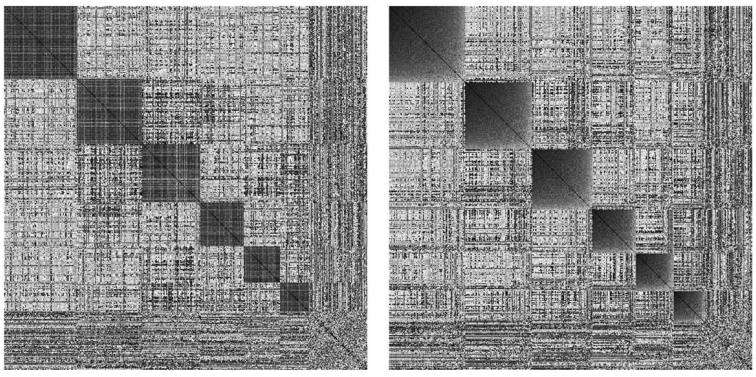


Fig. 5. Intensity images on Data Set II. (a) The original cluster structure. (b) Clustering result of CST.

TABLE 5
Experimental Results (Data Set III)

Methods	Time (s)	# Clusters	Γ Statistic	Matching Coefficient	Jaccard Coefficient
CST	2	74 (M=5)	0.739	0.979	0.876
Smart CAST	60	113 (M=5)	0.742	0.986	0.913
CAST-FI	320	90 (M=5)	0.741	0.981	0.888
k-means (k=2-21)	966	12 (M=12)	0.411	0.879	0.343
k-means (k=2-41)	3895	12 (M=12)	0.411	0.879	0.343

TABLE 6
Experimental Results (Data Set IV)

Methods	Time (s)	# Clusters	Γ Statistic	Matching Coefficient	Jaccard Coefficient
CST	4	57 (M=10)	0.586	0.968	0.669
Smart CAST	92	117 (M=10)	0.610	0.989	0.869
CAST-FI	355	127 (M=10)	0.609	0.990	0.879
k-means (k=2-21)	807	21 (M=21)	0.344	0.930	0.304
k-means (k=2-41)	3417	40 (M=24)	0.350	0.942	0.323

observation from Fig. 7 and Fig. 8 is similar to that on Data Set III and Data Set IV. Moreover, Fig. 8 also shows that the clusters produced by CST absorb slightly more outliers. In the empirical evaluations, intensity images can really help evaluate the validity of the clustering results and prove the correctness of our method.

4.3 The Yeast Cell-Cycle Data Set

Table 7 presents the total execution time and the best clustering quality of the tested methods on the real data set, i.e., the yeast cell-cycle data set as described previously. The CST method produced four main clusters with one outlier, while other methods generated five main clusters. It is observed that CST, Smart-CAST, and CAST-FI outperform k-means substantially in both of execution time and clustering quality. In particular, CST delivers the best execution time and clustering quality. Specifically, CST performs 75 to 325 times faster than k-means with k ranged as [2, 21] and [2, 41], respectively.

To investigate the clustering quality, Fig. 9 shows the intensity images for the original cluster structure of the data set and the clustering results. From Fig. 9a, we can easily observe that there are four main clusters in the yeast data set. It is observed that Fig. 9b is very similar to Fig. 9a, meaning that the clustering result generated by CST is very close to the original cluster structure. In fact, CST produced four main clusters, namely, G1, S/G2, M, and M/G1 clusters, with one outlier named SRD1. This matches the actual structure of this yeast data set, as described previously. In contrast, Smart-CAST, CAST-FI, and k-means produced five main clusters as the best clustering results, which is kind of deviated from the actual structure of the data set. Hence, it is shown that CST outperforms other methods substantially in terms of execution time and clustering quality on the real yeast cell-cycle data set.

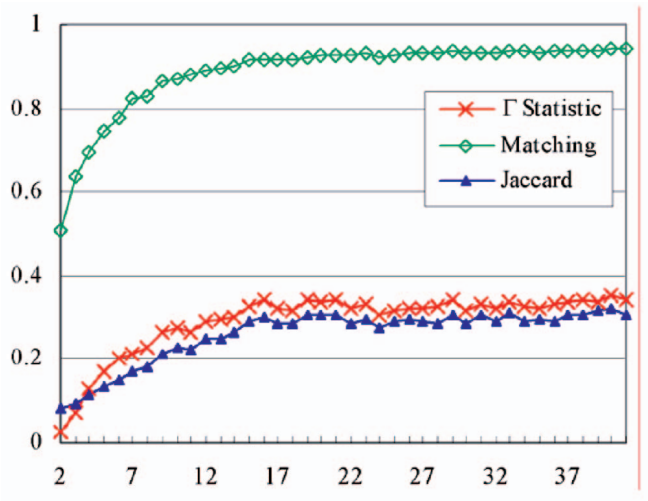
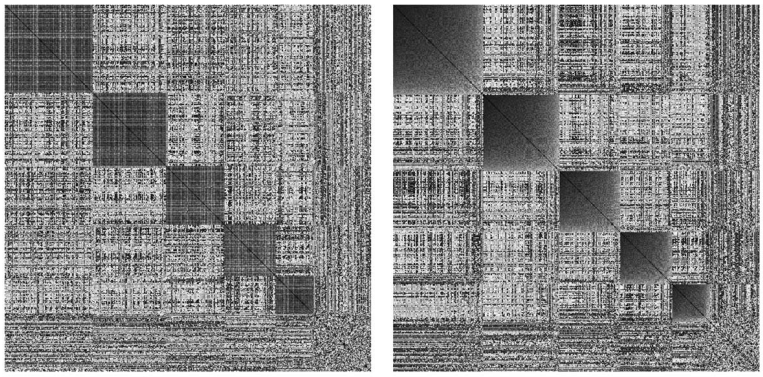


Fig. 6. The quality of clustering results of k-means on Data Set IV with k ranged as [2, 41].

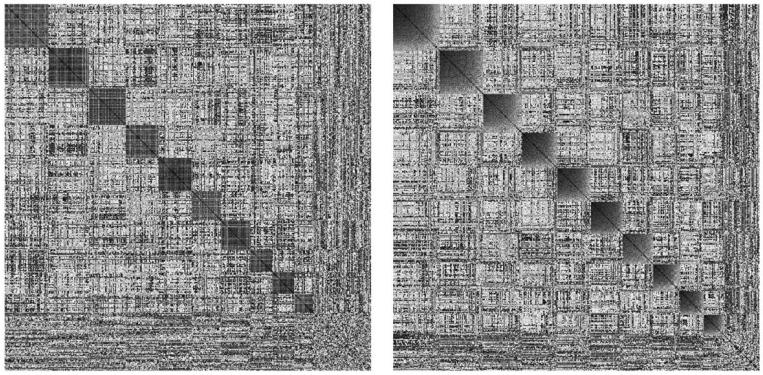
5 Conclusions

Clustering analysis is a valuable and useful technique for in-silico analysis of microarray or gene expression data, but most clustering methods incur problems in the aspects of automation, quality, and efficiency. In this paper, we explore the issue of effective integration of clustering methods and validation techniques. We propose a novel, parameterless, and efficient clustering algorithm, namely, CST, which is fit for analysis of gene expression data. CST clusters the genes using Hubert's Γ statistic on-the-fly and produces a "nearly optimal" clustering result. Through



(a) (b)

Fig. 7. Intensity images on Data Set III. (a) The original cluster structure. (b) Clustering result of CST.

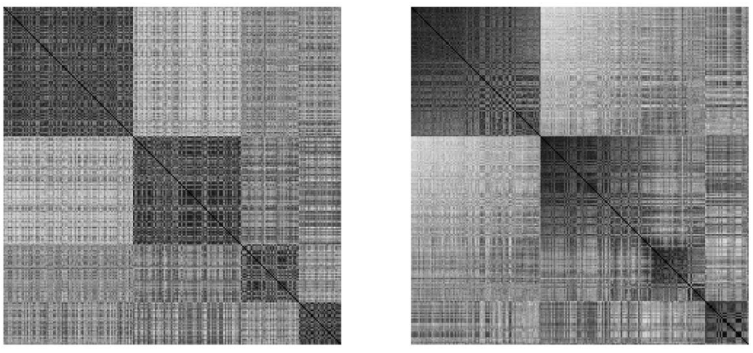


(a) (b)

Fig. 8. Intensity images on Data Set IV. (a) The original cluster structure. (b) Clustering result of CST.

TABLE 7
Experimental Results (Yeast Data Set)

Methods	Time (s)	# Clusters	Γ Statistic	Matching Coefficient	Jaccard Coefficient
CST	< 1	5 (M=4)	0.703	0.983	0.941
Smart CAST	< 1	5 (M=5)	0.706	0.978	0.922
CAST-FI	3	5 (M=5)	0.705	0.973	0.908
k-means (k=2-21)	75	5 (M=5)	0.683	0.914	0.724
k-means (k=2-41)	325	5 (M=5)	0.683	0.914	0.724



(a) (b)

Fig. 9. Intensity images of the yeast data set. (a) The original cluster structure. (b) Clustering result of CST.

performance evaluation on synthetic and real gene expression data sets, the CST method is shown to achieve higher efficiency and clustering quality than other methods without requesting any user-input parameter. Therefore, CST can provide a high degree of automation, efficiency, and clustering quality, which are lacking in other clustering methods for gene expression analysis.

In the future, we will further explore the following issues:

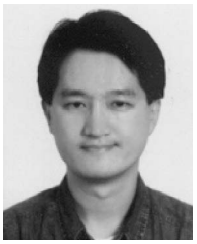
1. Reduction of memory requirements: CST needs considerable memory to store the similarity matrix. Hence, a memory-efficient scheme is needed, especially when the number of genes in the microarry is large.
2. Applications of CST on more real microarray data sets: We have shown that CST delivers excellent performance on the yeast cell-cycle data set. We will apply CST on more kinds of real microarray data sets to evaluate the validity of the clustering results, with the aim to promote related bioinformatics researches like disease markers discovery.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their valuable comments. This research was partially supported by the National Science Council, Taiwan, R.O.C., under grant no. NSC93-2321-B-006-009.

REFERENCES

[1] M.S. Aldenderfer and R.K. Blashfield, *Cluster Analysis.* Beverly Hills, Calif.: Sage Publications, 1984.
[2] 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'l Academy of Sciences,* vol. 96, pp. 6745-6750, 1999.
[3] A. Ben-Dor and Z. Yakhini, "Clustering Gene Expression Patterns," *J. Computational Biology,* vol. 6, pp. 281-297, 1999.
[4] G.A. Carpenter and S. Grossberg, "A Massive Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine," *Computer Vision, Graphics, and Image Processing,* vol. 37, pp. 54-115, 1987.
[5] M.-S. Chen, J. Han, and P.S. Yu, "Data Mining: An Overview from a Database Perspective," *IEEE Trans. Knowledge and Data Eng.,* vol. 8, no. 6, pp. 866-883, Dec. 1996.
[6] D.L. Davies and D.W. Bouldin, "A Cluster Separation Measure," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 2, pp. 224-227, 1979.
[7] M.B. Eisen, P.T. Spellman, P.O. Brown, and D. Botstein, "Clustering Analysis and Display of Genome Wide Expression Patterns," *Proc. Nat'l Academy of Sciences,* vol. 95, pp. 14863-14868, 1998.
[8] S. Guha, R. Rastogi, and K. Shim, "CURE: An Efficient Clustering Algorithm for Large Databases," *Proc. ACM Int'l Conf. Management of Data,* pp. 73-84, 1998.
[9] S. Guha, R. Rastogi, and K. Shim, "ROCK: A Robust Clustering Algorithm for Categorical Attributes," *Proc. 15th Int'l Conf. Data Eng.,* pp. 512-521, 1999.
[10] R.J. Hathaway and J.C. Bezdek, "Visual Cluster Validity for Prototype Generator Clustering Models," *Pattern Recognition Letters,* vol. 24, nos. 9-10, pp. 1563-1569, 2003.
[11] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data.* Englewood Cliffs, N.J.: Prentice Hall, 1988.
[12] M.K. Kerr and G.A. Churchill, "Bootstrapping Cluster Analysis: Assessing the Reliability of Conclusions from Microarray Experiments," *Proc. Nat'l Academy of Science,* vol. 98, no. 16, pp. 8961-8965, 2001.
[13] T. Kohonen, "The Self-Organizing Map," *Proc. IEEE,* vol. 78, no. 9, pp. 1464-1479, 1990.
[14] F.J. Rohlf, "Classification of Aedes by Numerical Taxonomic Methods (Diptera, Culicidae)," *Annals of the Entomological Soc. Am.,* vol. 56, pp. 798-804, 1963.
[15] D.E. Rumelhart and D. Zipser, "Feature Discovery by Competitive Learning," *Cognitive Science,* vol. 9, pp. 75-112, 1985.
[16] P.T. Spellman, G. Sherlock, M.Q. Zhang, V.R. Iyer, K. Anders, M.B. Eisen, P.O. Brown, D. Botstein, and B. Fucher, "Comprehensive Identification of Cell Cycle-Regulated Genes of the Yeast Saccharomyces Cerevisiae by Microarray Hybridization," *Molecular Biology of the Cell,* vol. 9, no. 12, pp. 3273-3297, 1998.
[17] 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'l Academy of Sciences,* vol. 96, no. 6, pp. 2907-2912, 1999.
[18] S.M. Tseng and L.J. Chen, "An Empirical Study of the Validity of Gene Expression Clustering," *Proc. Int'l Conf. Math. and Eng. Techniques in Medicine and Biological Sciences (METMBS '02),* 2002.
[19] S.M. Tseng and C.P. Kao, "Mining and Validating Gene Expression Patterns: An Integrated Approach and Applications," *Informatica,* vol. 27, pp. 21-27, 2003.
[20] K.Y. Yeung, D.R. Haynor, and W.L. Ruzzo, "Validating Clustering for Gene Expression Data," *Bioinformatics,* vol. 17, no. 4, pp. 309-318, 2001.
[21] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An Efficient Data Clustering Method for Very Large Databases," *Proc. 1996 ACM SIGMOD Int'l Conf. Management of Data,* pp. 103-114, 1996.



Vincent S. Tseng received the BS and MS degrees in computer science and information engineering from National Chiao Tung University, Taiwan, R.O.C., in 1988 and 1990, respectively. He received the PhD degree in computer and information science from National Chiao Tung University in 1997. From January 1998 and July 1999, he was an invited postdoctoral research fellow in the Computer Science Division at the University of California, Berkeley. Since August 1999, he has been on the faculty of the Department of Computer Science and Information Engineering at National Cheng Kung University, Taiwan. He has also been the director of the Department of Medical Informatics at National Cheng Kung University Hospital, Taiwan, since February 2004. Dr. Tseng has a wide variety of research specialties covering data mining, mobile Web technology, bioinformatics, and real-time systems. He has published numerous research papers in referred journals and international conferences. He is a member of the IEEE, the IEEE Computer Society, and the ACM, and an honorary member of the Phi Tau Phi Society. He has served on program committees for a number of international conferences including ACM SIGKDD Workshop on Data Mining in Bioinformatics (BioKDD), 2003, Pacific-Asia Symposium on Knowledge Discovery and Data Mining (PAKDD), 2002, International Conference on Real-Time Technology and Applications (RTAS), 2001 and 2003, etc.

Ching-Pin Kao received the BS degree from Yuan Ze University in 1999 and the MS degree from the National Cheng Kung University in 2001, both in computer science and engineering. He is currently a PhD student in computer science and information engineering at the National Cheng Kung University. His research interests include data mining, clustering techniques, bioinformatics, and gene expression analysis.