

# An Efficient Approach to Identifying and Validating Clusters in Multivariate Datasets with Applications in Gene Expression Analysis

VINCENT SHIN-MU TSENG AND CHING-PING KAO  
*Department of Computer Science and Information Engineering*  
*National Cheng Kung University*  
*Tainan, 701 Taiwan*  
*E-mail: tsengsm@mail.ncku.edu.tw*  
*E-mail: zeno@dmlab.csie.ncku.edu.tw*

Gene expression data analysis has become an important topic in bioinformatics due to its wide application in the biomedical industry. Effective analysis of gene expression data is an essential part of various data mining methods, especially the clustering techniques. Various kinds of clustering methods have been proposed, yet they do not satisfy for the requirements of high efficiency, high quality and automation in the mining of gene expression data. In this paper, we propose an efficient and automatic clustering approach that is suitable for gene expression analysis. The proposed approach primarily employs similarity-matrix based clustering techniques, complemented by new heuristics for reducing the computation cost. In particular, a novel validation technique is incorporated for evaluating the quality of the discovered gene expression patterns. Because it includes empirical evaluation of different gene expression datum, the proposed approach is able to perform better than other methods in terms of efficiency, clustering quality and automation.

**Keywords:** data mining, clustering, gene expression, microarray, validation technique

## 1. INTRODUCTION

In recent years, the DNA microarray [8, 19] has become an important and widely used technology since it enables the possibility of examining the expressions of thousands of genes simultaneously in a single experiment. An important research issue underlying microarray applications is the analysis and interpretation of the gene expression data obtained via microarray experiments [21]. The gene expression patterns obtained by analyzing microarray data can be used for a variety of inference tasks, such as the measurement of a gene's involvement in a particular cellular event or process [21], predict regulatory elements [6], etc.

A key step in the analysis of gene expression data is the detection of gene groups that manifest similar expression patterns. The main algorithmic problem here is to cluster multi-conditions gene expression patterns. Basically, a cluster algorithm partitions entities into groups based on the given features of the entities, so that the clusters are homo-

geneous and well separated [7]. More specifically, the aim is to identify sets of genes that behave similarly across the conditions. A variety of clustering methods have been proposed for the mining of gene expression data [2, 4, 5, 9, 22, 24].

Although a number of clustering methods have been studied in the literature, they are not satisfactory in terms of: 1) automation, 2) quality, and 3) efficiency. With regard to automation, most clustering algorithms request users to input some parameters needed to conduct the clustering task. For example, k-means [15] requires the user to input the number of clusters  $k$  to be generated. However, in real applications, it is often difficult for, say, a biologist to manually determine the correct parameters for the clustering task. Hence, an automated clustering method is required. As for quality, an accurate and efficient validation method for evaluating the quality of clustering results is lacking. Consequently, it is difficult to provide the user with information regarding how good each clustering result is. As for efficiency, the existing clustering algorithms may not perform well when the optimal or near-optimal clustering result is required from the global point of view.

In this paper, we propose an integrated approach to identifying and validating clusters in multivariate datasets and apply it to the mining of multi-conditions gene expression data. This approach incorporates the density-based clustering method along with validation techniques to achieve automation and accuracy in clustering. Furthermore, an iterative computing process is adopted to reduce the amount of computation required for clustering so as to meet the requirement of efficiency. Through experiments conducted on real gene expression data, the proposed approach is shown to deliver higher efficiency, clustering quality and automation than other methods can.

The rest of the paper is organized as follows: In section 2, some related studies are introduced. Our approach is described in section 3. Experiments conducted to evaluate the performance of the proposed method are presented in section 4. Conclusions and future works are given in section 5.

## 2. RELATED WORK

In recent years, a number of clustering methods have been proposed, and they can be classified into several different types [14]: partitioning-based methods (e.g., k-means [15], k-medoids [14], PAM [14], and CLARA [14]), hierarchical methods (e.g., UPGMA [15], BIRCH [27], CURE [12], and ROCK [13]), density-based methods (e.g., CAST [5], DBSCAN [10], OPTICS [3], WaveCluster [20]), grid-based methods (e.g., CLIQUE [1], STING [25], WaveCluster [20]), model-based methods (e.g., SOM [16] and COBWEB [11]), etc. Among them, several methods have been applied to cluster gene expression datasets, such as in [2, 4, 5, 9, 22, 24].

The k-means algorithm partitions the dataset into  $k$  groups, based primarily on the distance between data items, where  $k$  is a parameter specified by the user. Hierarchical clustering methods have been applied extensively and shown to be valuable for analyzing gene expression patterns. For example, hierarchical clustering can be used to separate tumor from normal tissues and to differentiate between tumor types based on the gene expression patterns in each tissue. SOM (Self-Organizing Map) was developed as an artificial neural network algorithm to achieve better speech recognition and was used by

Tamayo [22] et al. CAST (Cluster Affinity Search Technique) takes as input a parameter called the *affinity threshold*  $t$ , where  $0 < t < 1$ , and tries to guarantee that the average similarity in each generated cluster is higher than the threshold  $t$ . The main advantage of CAST is that it can detect outliers more effectively.

Although a number of clustering algorithms have been proposed, they may not find the best clustering result efficiently and automatically based on the given dataset. An important problem involved here is how to validate the clustering result. Jain and Dubes [15] divided the cluster validation procedure into two main parts: external and internal criterion analysis. External criterion analysis validates a clustering result by comparing it to a given “standard,” which is another partition of the data objects. There are many statistical measures that assess the agreement between an external criterion and a clustering result. For example, Milligan et al. [17, 18] evaluated the performance of different clustering algorithms and different statistical measures of agreement for both synthetic and real data. The problem of external criterion analysis is that reliable external criteria are rarely available when gene expression data are analyzed. In contrast, internal criterion analysis uses information within the given data set to represent the goodness of fit between the input dataset and the clustering result. For example, compactness and isolation of clusters are possible measures of goodness of fit. A measure called the Figure of Merit (FOM) was used by Yeung et al. [26] to evaluate the quality of clustering performed on a number of real datasets.

The main drawback of the existing clustering methods when applied for gene expression pattern analysis is that they can not meet the requirements of automation, high quality and high efficiency simultaneously during the analysis process. In the following, we describe a new approach to gene expression analysis that integrates clustering and validation techniques such that automation, high quality and high efficiency are achieved simultaneously.

### 3. OUR APPROACH

In this section, we first define the problem. Then we describe our approach in detail, including the principles behind it and the computation reduction method.

#### 3.1 Problem Definition

The problem of multivariate gene expression clustering can be described briefly as follows. Given a set of genes with unique identifiers, a vector  $E_i = \{E_{i1}, E_{i2}, \dots, E_{in}\}$  is associated with each gene  $i$ , where  $E_{ij}$  is a piece of numerical data that represents the response of gene  $i$  under condition  $j$  (or say, variant  $j$ ). The goal of gene expression clustering is to group together genes with similar expressions over all the conditions. That is, genes with similar corresponding vectors should be classified into the same cluster.

#### 3.2 Proposed Method

The main steps in the proposed approach are as follows. Given a piece of gene expression data, the first step in our approach is to calculate a similarity matrix  $S$  in which

the entry  $S_{ij}$  represents the similarity of the expression patterns for genes  $i$  and  $j$ . Although a number of alternative measures could be used to calculate the similarity between gene expressions, we use Pearson's correlation coefficients [15] here due to its wide application range. Note that a similarity matrix needs to be computed and generated only once for a given gene expression dataset. This much reduces the computation overhead incurred by some clustering algorithms that calculate the similarities dynamically.

In the second step, a density-and-affinity based algorithm is applied as the base clustering algorithm. Along with a specified input parameter, the base clustering algorithm utilizes the similarity matrix  $S$  to conduct the clustering task. Thus, a clustering result will be produced by the base clustering algorithm based on the given input parameter. A good candidate for the base clustering algorithm is CAST (Cluster Affinity Search Technique) [5], which needs only one input parameter, called the *affinity threshold*  $t$ , where  $0 < t < 1$ . CAST generates one cluster at a time and selects the object with the most neighbors as a seed for the current cluster. It adds un-clustered objects with high affinity (the average similarity between the object and the cluster is greater than  $t$ ) to the current cluster and removes objects with low affinity from the current cluster iteratively. More detailed descriptions of CAST can be found in [5]. The reasons why we prefer to use CAST as the base clustering method are as follows:

- (1) CAST has the capability of isolating outliers.
- (2) Unlike other algorithms that are unstable (e.g., k-means), CAST is a stable algorithm.
- (3) The execution time of CAST is shorter than most of the other clustering algorithms.

In the third step, a validation test is performed to evaluate the quality of the clustering result produced in step two. We adopt *Hubert's  $\Gamma$  statistic* [15] to measure the quality of clustering. Let  $\mathbf{X} = [X(i, j)]$  and  $\mathbf{Y} = [Y(i, j)]$  be two  $n \times n$  matrix, where  $X(i, j)$  indicates the similarity of genes  $i$  and  $j$ , and  $Y(i, j)$  is defined as follows:

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

Hubert's  $\Gamma$  statistic represents the point serial correlation between the matrixes  $\mathbf{X}$  and  $\mathbf{Y}$ , and is defined as follows:

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

where  $M = n(n-1)/2$  and  $\Gamma$  is between  $[-1, 1]$ . Let matrix  $\mathbf{X}$  be the similarity matrix derived from the gene expression data. Then, matrix  $\mathbf{Y}$  and *Hubert's  $\Gamma$  statistic* can be calculated easily based on matrix  $\mathbf{X}$ , as shown in (1) and (2), respectively. It is used to measure the correlation between the similarity matrix  $\mathbf{X}$  and the adjacent matrix  $\mathbf{Y}$  of the clustering result. For a clustering result, a greater value of  $\Gamma$  represents better clustering quality. In [23], S. M. Tseng and L. J. Chen evaluated a number of validity indexes for measuring the quality of clustering results. They concluded from the experimental results

that Hubert's  $\Gamma$  might be the best index for both partition-based clustering methods and density-based methods for both low-similarity and high-similarity data.

With the above steps, it is clear that high quality clustering can be achieved by applying a number of different values of the *affinity threshold*  $t$  as input parameters to the CAST algorithm, by calculating the *Hubert's  $\Gamma$  statistic* of each clustering result, and by choosing the one with the highest value of the *Hubert's  $\Gamma$  statistic* as the output. In this way, a local-optimal clustering result can be obtained by users automatically. An example is shown in Fig. 1, where the  $X$  axis represents the values of the *affinity threshold*  $t$  input to CAST and the  $Y$  axis shows the obtained *Hubert's  $\Gamma$  statistic* for each of the clustering results. The peak in the curve corresponds to the best clustering result, which has a *Hubert's  $\Gamma$  statistic* value of around 0.52 when  $t$  is set to be 0.25.

This approach is feasible in that firstly, CAST executes very quickly since the similarity matrix of gene expressions is obtained in advance; secondly, the *Hubert's  $\Gamma$  statistic* for each clustering result can be calculated easily. However, one problem encountered is how to determine suitable values of the affinity threshold  $t$ . The easiest way is to increment the value of the *affinity threshold*  $t$  with a fixed interval. For example, we may increase the value of  $t$  from 0.05 to 0.95 in increments of 0.05. We call this approach CAST-FI (Fixed Increment) in the following. The main disadvantage of CAST-FI is that many iterations of computations are required. Therefore, a new method is proposed in the next section to reduce the computation overhead.

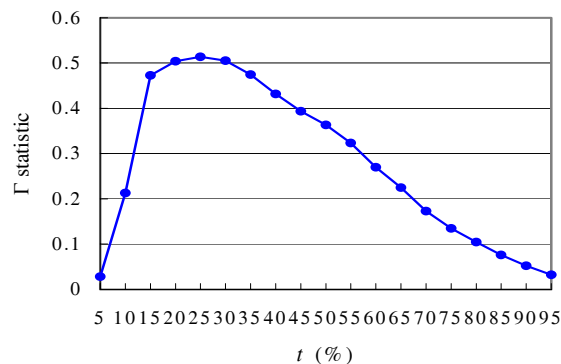


Fig. 1. *Hubert's  $\Gamma$  statistic* vs. values of  $t$ .

### 3.3 Computation Reduction Method

The idea behind the proposed method is to reduce the amount of computation by eliminating unnecessary executions of clustering so as to obtain a “nearly-optimal” clustering result instead of the optimal one. That is, we try to execute CAST as few times as possible. Therefore, we need to narrow down the range of the parameter affinity threshold  $t$  effectively. The proposed method works as follows:

1. Initially, a testing range  $R$  for setting the *affinity threshold*  $t$  is set to be  $[0, 1]$ . We divide  $R$  equally into  $m$  parts with the points  $P_1, P_2, \dots, P_{m-1}$ , where  $P_1 < P_2 < \dots < P_{m-1}$ .

- $m \geq 3$ . Then, the value of each  $P_i$  is taken as the *affinity threshold*  $t$  for executing CAST, and the  $\Gamma$  statistic of the clustering result for each of  $P_i$  is calculated. We call this process a “run.”
2. When a clustering run is completed, the clustering at point  $P_b$  that produces the highest  $\Gamma$  statistic is considered to be the best clustering. The testing range  $R$  is then replaced by the range  $[P_{b-1}, P_{b+1}]$  that contains the point  $P_b$ .
  3. The above process is repeated until the testing range  $R$  is smaller than a threshold  $\delta$  or the difference between the maximal and minimal quality values is smaller than another threshold  $\sigma$ .
  4. The best quality clustering result obtained through the tested process is output as the answer.

In this way, we can obtain a clustering result that has “nearly-optimal” clustering quality with much less computation. In the next section, through empirical evaluation, we shall evaluate how good the generated clustering result is and to what extent the amount of computation could be reduced by our approach.

## 4. EXPERIMENTAL EVALUATION

To validate the feasibility and performance of the proposed approach, we implemented the approach in C++ and applied it to both of real gene expression data and synthetic data. We will describe the experimental setup in section 4.1 and provide detailed experimental results for different types of data in sections 4.2, 4.3 and 4.4.

### 4.1 Experimental Setup

To evaluate the performance of our approach, we used microarray expression data of yeast *saccharomyces cerevisiae* obtained from the Lawrence Berkeley National Lab (LBNL) (<http://rana.lbl.gov/EisenData.htm>). The dataset contained the expressions of 6221 genes under 80 experimental conditions. Based on this dataset, we generated two smaller datasets with different properties for testing. First, we randomly chose 2000 genes from the dataset and called it Dataset I. The average similarity of this dataset was 0.137 as determined by using Pearson’s correlation coefficient as a measurement of similarity. Thus, Dataset I represented a low-similarity dataset. Secondly, to generate a dataset with higher similarity, we applied CAST to cluster Dataset I with the parameter  $t$  set to be 0.6. Then, we selected a large cluster from the clustering result of Dataset I and duplicated the gene expression patterns in this cluster to generate a dataset with around 1900 genes. Additionally, 100 outliers were mixed with the 1900 genes to form Dataset II, with about 2000 genes in total. The average similarity of Dataset II was about 0.644; thus, it represented a high similarity dataset. Finally, we used the original dataset as Dataset III, which contained the expressions of 6221 genes under 80 conditions and represented a large dataset.

We compared our approach with *CAST-FI*, *E-CAST* and the well-known clustering method, namely, *k-means* [15]. *E-CAST* is an enhanced CAST algorithm proposed by Bellaachia [4]. It automatically computes an affinity threshold  $t$  before generating each

cluster. For our approach, the parameters  $m$ ,  $\delta$  and  $\sigma$  were set to default values of 4, 0.01 and 0.01, respectively. For k-means, the value of  $k$  was tested in two ways: 1)  $k$  was varied from 3 to 21 in increments of 2; 2)  $k$  was varied from 3 to 39 in increments of 2, respectively. Since k-means was not a stable algorithm, the curve of *Hubert's  $\Gamma$  statistic* vs. values of  $t$  was not parabolic. Therefore, we could not use the proposed method for k-means. This is also why we set the value of  $k$  in two different amounts of ranges for k-means so as to examine its performance under different bounded execution time.

The quality of the clustering results was measured the using the Hubert's  $\Gamma$  statistic. The experimental results for Datasets I, II and III will be described in the following sections.

#### 4.2 Dataset I: Low Similarity Dataset

The total execution times and the best clustering quality for the tested methods applied to Dataset I are listed in Table 1. The notation "CAST-FI" indicates that CAST was run iteratively by varying the *affinity threshold*  $t$  from 0.05 to 0.95 in fixed increments of 0.05, while the notation "Our Approach" indicates our proposed approach with the computation reduction method described in section 3.3. The increment of 0.05 for the affinity threshold in CAST-FI was just an experimental setting, with the aim to finding the nearly optimal result under constrained execution time. Of course, any alternative value could be used. With a smaller value of it, a more accurate clustering result would be produced, but the execution time would also be longer.

**Table 1. Experimental results obtained using the tested methods with Dataset I.**

Methods	Time (sec)	Number of clusters	$\Gamma$ Statistic
Our Approach	27	57	0.514
CAST-FI	246	57	0.514
E-CAST	1412	574	0.287
k-means ( $k = 3\sim 21$ )	404	5	0.447
k-means ( $k = 3\sim 39$ )	1092	5	0.447

It is obvious that our approach and CAST-FI outperformed E-CAST and k-means substantially in terms of both execution time and clustering quality. In particular, our approach performed 15 times and 40 times faster than k-means when the  $k$  range was [3, 21] and [3, 39], respectively. The results show that E-CAST performed very poorly in terms of execution time and clustering quality. This shows that E-CAST was not suitable for analyzing this low similarity dataset since the automatically computed affinity threshold  $t$  was high ( $t \geq 0.5$ ). Consequently, the number of clusters is higher because of the lower similarity. Meanwhile, this results in high cost in computing  $t$ . In addition, the results also show that the highest  *$\Gamma$  statistic* value generated by our approach was very close to that generated by CAST-FI, meaning that the clustering quality of our approach was as good as that of CAST-FI. However, our approach was about 8 times faster than CAST-FI. Therefore, it can be observed that our approach outperformed the other clustering methods greatly in terms of both quality and computation time.

Table 2 shows the distribution of clusters produced by each tested method. It is shown that k-means generated 5 clusters as the best clustering result, while the size of each cluster ranged between 101 and 400. This phenomenon occurred no matter  $k$  whether varied from 3 to 29 or from 3 to 39. E-CAST produced a large number of clusters that were small in size (1~10 and 11~100) because the similarity of Dataset I was low. However, our approach produced 57 clusters as the best clustering result. In particular, it is clear that 4 main clusters were generated, with two clusters between 101 to 400 in size and another two between 401 to 600 in size. Moreover, our approach also generated a number of clusters that were small in size (1~10 and 11~100), which were mostly outliers (or noise). This means that our approach is superior to k-means in filtering out outliers from main clusters. Therefore, it can provide more accurate clustering results and insight for gene expression analysis.

**Table 2. Distribution of produced clusters for Dataset I.**

Cluster size Methods	1~10	11~100	101~400	401~600
Our Approach	38	15	2	2
CAST-FI	38	15	2	2
E-CAST	546	27	1	0
k-means ( $k = 3\sim 21$ )	0	0	5	0
k-means ( $k = 3\sim 39$ )	0	0	5	0

The following observations can be made based on this experiment:

1. In terms of clustering quality, our approach and CASI-FI perform much better than E-CAST and k-means, especially in isolating outliers. This means that density-and affinity based methods are superior to partitioning-based methods in clustering low-similarity gene expression data.
2. Our approach finding the best clustering result much faster than CASI-FI, while the resulting clustering quality is very similar to that of CASI-FI. This illustrates the advantage of the computing reduction method described in section 3.3.

#### 4.3 Dataset II: High Similarity Dataset

We conducted the same experiments but replaced Dataset I with Dataset II, representing a dataset with higher similarity. Tables 3 and 4 show the experimental results obtained using the tested methods and the cluster size distribution for Dataset II, respectively. The following observations can be made based on the empirical results:

1. It is obvious that our approach and CAST-FI outperform E-CAST and k-means substantially in terms of the clustering quality ( $\Gamma$  statistic). Compared to the experimental results for Dataset I, the degree of improvement our approach achieved over k-means with Dataset II was much higher in terms of the clustering quality. In fact, observing

**Table 3. Experimental results obtained using the tested methods with Dataset II.**

Methods	Time (sec)	Number of clusters	$\Gamma$ Statistic
Our Approach	13	63	0.833
CAST-FI	41	62	0.833
E-CAST	32	93	0.696
k-means ( $k = 3\sim 21$ )	77	12	0.309
k-means ( $k = 3\sim 39$ )	267	12	0.309

**Table 4. Distribution of produced clusters for Dataset II.**

Methods \ Cluster size	1~10	11~100	101~400	401~600
Our Approach	62	0	0	0
CAST-FI	61	0	0	0
E-CAST	84	6	2	1
k-means ( $k = 3\sim 21$ )	4	5	3	0
k-means ( $k = 3\sim 39$ )	4	5	3	0

the size distribution of the generated clusters shown in Table 4, we found that both of our approach and CAST-FI produced a main cluster that was large in size (1901-2000) and many other small clusters, which were actually outliers. This matches the real distribution of Dataset II as described in section 4.1. In contrast, E-CAST partitioned the large cluster into several clusters of diverse sizes, while k-means produced clusters of uniform sizes. Consequently, the clustering results differ from the original data distribution. This indicates that E-CAST and k-means can not perform well with a high similarity dataset. In particular, k-means can not identify outliers correctly.

- As for execution time, again our approach is much faster than the other methods. Compared to CAST-FI, our approach achieved clustering quality as good as that achieved by CAST-FI in a much shorter amount of execution time. This shows that our approach can achieve high efficiency and accuracy with a high similarity dataset.

#### 4.4 Dataset III: Large Dataset

We conducted the same experiments as described in section 3.1 and 3.2 but replaced the dataset with the original dataset. Compared to Datasets I and II, Dataset III represented a real, large dataset. The results obtained using E-CAST are not listed here since its execution time exceeded two hours and the clustering quality was very poor. Table 5 shows the experimental results obtained using the other tested methods. Compared to the results for Datasets I and II, the execution time under Dataset III was longer due to its larger size. However, our approach still outperformed the other methods substantially in terms of both clustering quality and execution time. Table 6 shows the distribution of clusters with the different methods. Again, it is observed that our approach and CAST-FI is superior to k-means in detecting outliers and isolating them as small clusters, even in the case of a large dataset.

**Table 5. Experimental results obtained using the tested methods with Dataset III.**

Methods	Time (sec)	Number of clusters	$\Gamma$ Statistic
Our Approach	530	123	0.508
CAST-FI	4453	97	0.504
E-CAST	2210	5	0.446
k-means ( $k = 3\sim 21$ )	5983	5	0.446
k-means ( $k = 3\sim 39$ )	530	123	0.508

**Table 6. Distribution of produced clusters for Dataset III.**

Cluster size Methods	1~10	11~100	101~400	401~600
Our Approach	74	40	6	3
CAST-FI	58	32	4	3
E-CAST	0	0	0	5
k-means ( $k = 3\sim 21$ )	4	5	3	0
k-means ( $k = 3\sim 39$ )	74	40	6	3

## 5. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an efficient approach to identifying and validating clusters in multivariate data sets and applied it to the mining of gene expressions in multi-condition experiments. Performance experiments on real and synthetic microarray datasets showed that the proposed approach can achieve higher efficiency and clustering quality than other methods with different datasets. Moreover, the proposed approach can discover the “nearly-best” clustering result without requiring users to set parameters. Therefore, the proposed approach can achieve a high degree of automation, efficiency and clustering quality, compared to other clustering methods for gene expression mining.

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

1. We will seek to reduce the initial range of the input parameter, *affinity threshold*  $t$ , for executing CAST. This will significantly reduce the amount of computation required after the correct range is estimated.
2. We will design a memory-efficient clustering method which will be integrated into our iteratively clustering approach. This will be especially useful when the number of tested genes in the microarray is large.
3. We will extend our approach to capture the pattern structure embedded in the data set. This will provide more insight into the relationships between the data points in the dataset.

## ACKNOWLEDGMENT

The authors would like to thank the referees for the valuable and insightful comments. This research was partially supported by National Science Council, Taiwan, R.O.C., under grant no. NSC91-2213-E-006-132.

## REFERENCES

1. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, "Automatic subspace clustering of high dimensional data for data mining applications," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1998, pp. 94-105.
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," in *Proceedings of National Academy of Science*, Vol. 96, 1999, pp. 6745-6750.
3. M. Ankerst, M. M. Breunig, H. P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," in *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data*, 1999, pp. 49-60.
4. A. Bellaachia, D. Portnoy, Y. Chen, and A. G. Elkahoulou, "E-CAST: a data mining algorithm for gene expression data," in *Proceedings of the ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIODDD 2002)*, 2002, pp. 49-54.
5. A. Ben-Dor and Z. Yakhini, "Clustering gene expression patterns," *Journal of Computational Biology*, Vol. 6, 1999, pp. 281-297.
6. A. Brazma, I. Jonassen, J. Vilo, and E. Ukkonen. "Predicting gene regulatory elements in silico on a genomic scale," *Genome Research*, Vol. 8, 1998, pp. 1202-1215.
7. M. S. Chen, J. Han, and P. S. Yu, "Data mining: an overview from a database perspective," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, 1996, pp. 866-883.
8. J. DeRisi, L. Penland, P. O. Brown, M. L. Bittner, P. S. Meltzer, M. Ray, Y. Chen, Y. A. Su, and J. M. Trent, "Use of a cDNA microarray to analyze gene expression patterns in human cancer," *Nature Genetics*, Vol. 14, 1996, pp. 457-460.
9. M. B. Eissen, P. T. Spellman, P. O. Brown, and D. Botstein, "Clustering analysis and display of genome wide expression patterns," in *Proceedings of the National Academy of Sciences*, Vol. 95, 1998, pp. 14863-14868.
10. M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD '96)*, 1996, pp. 226-231.
11. D. H. Fisher, "Knowledge acquisition via incremental conceptual clustering," *Machine Learning*, Vol. 2, 1987, pp. 139-172.
12. S. Guha, R. Rastogi, and K. Shim, "CURE: an efficient clustering algorithm for large databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1998, pp. 73-84.
13. S. Guha, R. Rastogi, and K. Shim, "ROCK: a robust clustering algorithm for categorical attributes," in *Proceedings of the 15th International Conference on Data Engineering*, 1999, pp. 512-521.
14. J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufmann, San Francisco, 2001.
15. A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall, New Jersey, 1988.
16. T. Kohonen, "The self-organizing map," in *Proceedings of the IEEE*, Vol. 78, 1990,

- pp. 1464-1480.
17. G. W. Milligan, S. C. Soon, and L. M. Sokol, "The effect of cluster size, dimensionality and the number of clusters on recovery of true cluster structure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 5, 1983, pp. 40-47.
  18. G. W. Milligan and M. C. Cooper, "A study of the comparability of external criteria for hierarchical cluster analysis," *Multivariate Behavioral Research*, Vol. 21, 1986, pp. 441-458.
  19. M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, Vol. 270, 1995, pp. 467-470.
  20. G. Sheikholeslami, S. Chatterjee, and A. Zhang, "WaveCluster: a multi-resolution clustering approach for very large spatial databases," in *Proceedings of the 24th International Conference on Very Large Data Bases (VLDB '98)*, 1998, pp. 428-439.
  21. 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, 1998, pp. 3273-3297.
  22. 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," in *Proceedings of the National Academy of Science*, Vol. 96, 1999, pp. 2907-2912.
  23. S. M. Tseng and L. J. Chen, "An empirical study of the validity of gene expression clustering," in *Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences (METMBS '02)*, 2002, pp. 126-131.
  24. J. Wang, J. Delabie, H. C. Aasheim, E. Smeland, and O. Myklebost, "Clustering of the SOM easily reveals distinct gene expression patterns: results of a reanalysis of lymphoma study," *BMC Bioinformatics*, Vol. 3, 2002, pp. 36.
  25. W. Wang, J. Yang, and R. Muntz, "STING: a statistical information grid approach to spatial data mining," in *Proceedings of 23rd International Conference on Very Large Data Bases (VLDB '97)*, 1997, pp. 186-195.
  26. K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo, "Validating clustering for gene expression data," *Bioinformatics*, Vol. 17, 2001, pp. 309-318.
  27. T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1996, pp. 103-114.



**Vincent Shin-Mu Tseng (曾新穆)** received his B.S. and M.S. from Department of Computer Science and Information Engineering at National Chiao Tung University, Taiwan, R.O.C., in 1988 and 1990, respectively. He received the Ph.D. degree from Institute of Computer and Information Science at National Chiao Tung University in 1997. During January 1998 and July 1999, he was an invited postdoctoral research fellow in Computer Science Division of University of California, Berkeley, U.S.A.

Since August 1999, he has been on the faculty of Department of Computer Science and Information Engineering at National Cheng Kung University, Taiwan. Dr. Tseng has a wide variety of research specialties covering data mining, Internet technology, bioinformatics and real-time systems. He has published numerous research papers in referred journals and international conferences. He is a member of the Association for Computing Machinery, IEEE and honorary member of Phi Tau Phi Society. He has served as program committee 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, etc.



**Ching-Pin Kao (高慶斌)** was born on 1976 in Tainan, Taiwan, R.O.C. He received the B.S. degree from Yuan Ze University in 1999, and the M.S. degrees from the National Cheng Kung University in 2001, both in Computer Science and Engineering. He is currently a Ph.D. student in Computer Science and Information Engineering at National Cheng Kung University. His research interests include data mining, clustering techniques, bioinformatics and gene expression analysis.