Journal of Signal Processing Systems 50, 267–280, 2008
© 2007 Springer Science + Business Media, LLC. Manufactured in The United States.
DOI: 10.1007/s11265-007-0121-2

Discovering Biclusters by Iteratively Sorting with Weighted Correlation Coefficient in Gene Expression Data

LI TENG AND LAIWAN CHAN
Department of Computer Science and Engineering, The Chinese University of Hongkong, Hong Kong, People's Republic of China

Received: 31 May 2007; Accepted: 13 June 2007

Abstract. We propose a framework for biclustering gene expression profiles. This framework applies dominant set approach to create sets of sorting vectors for the sorting of the rows in the data matrix. In this way, the coexpressed rows of gene expression vectors could be gathered. We iteratively sort and transpose the gene expression data matrix to gather the blocks of coexpressed subset. Weighted correlation coefficient is used to measure the similarity in the gene level and the condition level. Their weights are updated each time using the sorting vector of the previous iteration. In this way, the highly correlated bicluster is located at one corner of the rearranged gene expression data matrix. We applied our approach to synthetic data and three real gene expression data sets with encouraging results. Secondly, we propose ACV (average correlation value) to evaluate the homogeneity of a bicluster or a data matrix. This criterion conforms to the intuitive biological notion of coexpressed set of genes or samples and is compared with the mean squared residue score. ACV is found to be more appropriate for both additive models and multiplicative models.

Keywords: biclustering, gene expression data, microarray data, weighted correlation coefficient

1. Introduction

DNA microarray technology has recently become a central role in biological and biomedical research. It enables measuring the expression level of many thousands of genes within a number of different experimental conditions simultaneously. Clustering is wildly used to analyze gene expression data to identify groups of genes that exhibit similar expression patterns. However, clustering has its limitations. The clustering process groups related genes that behave similar across all measured conditions. This assumption is reasonable when the dataset contains few conditions from a single, focused experiment, but does not hold for larger datasets containing hundreds of heterogeneous conditions from many experiments. In expression data analysis, more interesting is the finding of a subset of genes showing strikingly similar up-regulation and down-regulation under a subset of conditions.

Biclustering was introduced in the 1970s [1]. Cheng and Church [2] were the first to apply it to gene expression data analysis. Other names such as coclustering, bidimensional clustering and subspace clustering, often refer to the same problem formulation. Biclustering algorithms identify groups of genes that show similar activity patterns under a specific subset of the conditions. Many algorithms have been proposed and have been used mainly in identification of co-regulated genes, gene functional annotation and sample classification. Most of the approaches could be divided into 5 groups, iterative row and column clustering combination, divide and conquer, greedy iterative search, exhaustive bicluster enumeration and distribution parameter identification (for a review see [3]). The iterative row and column clustering combination methods, sometimes called two-way clustering [4, 5], rely on the results obtained by clustering each dimensions of the data matrix separately. Then in some way the algorithms combine the results of clustering on two dimensions to form stable clusters. When performing clustering on one dimension, the structure of the other dimension is unknown and could be broken. Divide and conquer algorithms, such as the direct clustering by Hartigan [6], are potentially very fast. However, the approaches have the significant drawback that good biclusters could miss by splitting. Greedy iterative search methods base on the idea of creating biclusters by adding or removing rows/columns, using a criterion that maximizes a local gain. These algorithms could be fast, however, they would make wrong decisions and loose good biclusters. Cheng and Church proposed the mean squared residue score and several greedy row/column removal/addition algorithms and combined them in an overall approach to find biclusters with low mean squared residue scores. The mean squared residue score and some similar criterions that based on the residue are widely used in gene expression biclustering [7–10]. The mean squared residue score could be affected by the magnitude of the original dataset and constant biclusters which could be considered trivial tend to be found using this criterion. Exhaustive bicluster enumeration methods are straightforward. While due to the complexity, those methods could only be executed under some restrictions on the size of the biclusters. Distribution parameter identification methods, such as the plaid model proposed by Lazzeroni and Owen [11], assume a statistical model and try to identify the distribution parameters by iteratively minimizing a certain criterion. Among them, many algorithms rely on the normalization steps. Variants normalization approaches have to be carried out before biclustering. Some normalization steps could be complex and the structures of the original biclusters could be changed. And most algorithms have very high computation cost.

Traditionally, similarity measure, such as the correlation coefficient, between two genes (conditions) is computed across all conditions (genes) with equal weight. In biclustering, this is not appropriate as we are interested in measuring the similarity within a subset of genes and conditions only. During the intermediate steps of biclustering, we do not know precisely the subset of the data matrix prescribing our target bicluster. Usually, we would start with an arbitrary subset and iteratively update the subset. As the selected subsets of genes (conditions) during the intermediate steps are discrete and depend on the selection algorithm, hence the quality of the final result is hard to guarantee.

In this paper, we solve this problem by assigning continuous weighting factors to different genes (conditions) in the intermediate steps. The suggested weighted similarity measurement is the weighted correlation coefficient. Here we emphasize on the similarities within a subset of genes (conditions) rather than in the whole dataset. Updating of the weighting factors is also an important issue affecting the quality of the final result. It has to be related with our sorting or selection algorithm in the intermediate steps. In Section 2, we propose an effective method of biclustering by alternatively sorting the genes and condition using dominant set [12]. The sorting vectors are used as the weighting factors in the subsequent step. Unlike the two-way clustering or divide and conquer algorithms, our approach does not break the structure of the whole data. And we find multiple overlapping biclusters. This conforms to the biological assumption that some genes (conditions) could be involved in more than one cellular process. This method has low computation cost comparing to the exhaustive enumeration and distribution parameter identification methods.

In Section 3, we propose a new criterion, the average correlation value (ACV), to evaluate the homogeneousness of a cluster under different situations. This criterion conforms to the intuitive biological notion of coexpressed subset of genes or samples. We compare it with the mean squared residue score, a widely used method in this field, using both theoretical analysis and empirical methods. ACV is found to be applicable in more circumstances.

In Section 4 we first test our approach on the synthetic data then we describe the experimental results on three gene expression datasets and compare them with the results of other methods.

In Section 5, its analytical comparison with other biclustering methods is discussed.

2. Biclustering Method

Correlation coefficient [Eq. (1)] is a wildly used similarity measurement in gene expression analysis. It conforms well to the intuitive biological notion of what it means for two genes to be "coexpressed". This statistic captures similarity in "trend" but places no emphasis on the magnitude of the two series of measurements [14]. Further, researchers from biology regard that negative relative is as informative as the positive relative. Using correlation coefficient, it is easy to measure both positive relativity and negative relativity.

$$r = \frac{\sum xy - \sum x \sum y/n}{\sqrt{\left[\sum x^2 - \left(\sum x\right)^2 \Big/ n\right]\left[\sum y^2 - \left(\sum y\right)^2 \Big/ n\right]}} \tag{1}$$

where x and y stands for two gene expression vectors with dimension n.

Correlated coefficient measures the overall similarity of the genes under all conditions, unless we particularly specify a subset of concerned conditions. However, it is common that the gene expressions for some conditions behave differently from other conditions. Figure 1a shows an example of two gene expression vectors. These two gene vectors appear uncorrelated and with a very low correlation coefficient of 0.0619. However, these two genes exhibit strong correlation under some conditions highlighted by the black rectangle in Fig. 1b. With such a low correlation coefficient, we have no clue that some of the conditions are indeed highly correlated. In fact, this is the mainly objective of biclustering to discover local patterns. Thus, this traditional correlation coefficient is not appropriate for local pattern discovering, as its drawback falls on the equal treatment of all conditions.

2.1. Weighted Correlation Coefficient

Microarrays allow the measurement of expression levels for a large number of genes within a number of different experimental conditions. The conditions may correspond to different toxins or time points. In other cases, the conditions may refer to samples from different organs, from tumors or healthy tissue, or from different individuals. In tradition, the genes or the samples would be compared under all conditions and we find the genes or samples behave similarly across all measured conditions. When the dataset contains many heterogeneous conditions, this assumption does not work. Some samples would only appear similar under a subset of features. In that case, all features should not be considered as equal when calculating the correlation coefficient. In gene expression biclustering, different conditions/genes should not contribute equally when measuring the

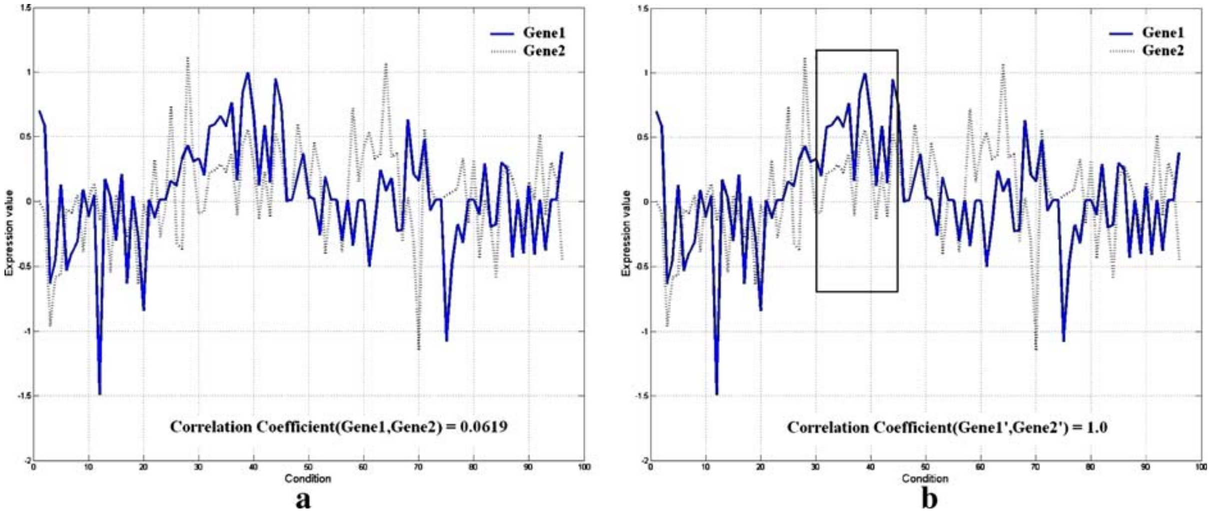


Figure 1. Drawback of the traditional correlation coefficient. **a** An example of two gene expression vectors. **b** These two genes exhibit strong correlation under some conditions highlighted by the *black rectangle*.

similarity between genes/conditions. Features with higher weights contribute more to the measurement. Suppose we have found a subset of features which have high similarities across the samples and we assign higher weights to these features. Then when measuring the similarities between the samples using Eq. (2), the samples with high similarities across the high-weighted features would be found. And this subset of features and samples would be regarded as a bicluster.

To achieve this, we use the weighted correlation coefficient proposed by JM Bland and DG Altman [13] who found that multiple observations from each subject produced a spurious increase in the sample size and a corresponding spurious "significant" relationship. They used the number of observations as weights and the actual formula for a weighted correlation coefficient is

$$\frac{\sum_{i=1}^{n} m_i\overline{x_i}\,\overline{y_i} - \sum_{i=1}^{n} m_i\overline{x_i}\sum_{i=1}^{n} m_i\overline{y_i} \Big/ \sum_{i=1}^{n} m_i}{\sqrt{\left(\sum_{i=1}^{n} m_i\overline{x_i}^2 - \left(\sum_{i=1}^{n} m_i\overline{x_i}\right)^2 \Big/ \sum_{i=1}^{n} m_i\right)\left(\sum_{i=1}^{n} m_i\overline{y_i}^2 - \left(\sum_{i=1}^{n} m_i\overline{y_i}\right)^2 \Big/ \sum_{i=1}^{n} m_i\right)}}, \tag{2}$$

Here m_i is the number of observations of the *i*th feature. $\overline{x_i}$ and $\overline{y_i}$ is the average value of the *i*th feature over m_i times of observations of the two samples respectively. The features with more number of observations are more creditable so they are assigned higher weights. When all the m_i are equal, they cancel out, giving the usual formula for a correlation coefficient in Eq. (1). Under the similar assumption that features contribution not equally to the similarity measurement, we redefine m_i as the feature weight for the *i*th feature. The higher the value of m_i is, the more important a feature is. When calculating weighted correlation coefficient between two samples, the features with more importance should have more impacts than the other features. And by this we concentrate on the impact of some features and take less consideration of the other features. As shown in Fig. 2, we assign a high weight of 0.0563 to the features indicing from 30th to 45th and a weight of 0.0013 to other features. Then the two genes have a comparative large weighted correlation coefficient of 0.7452, in contrary to the traditional correlation coefficient of 0.0619.

In biclustering analysis we find coexpressed subset of genes or conditions. When measuring the similarity between the genes or the conditions, we use sample space and feature space to present the roles of these two dimensions of the subset. That is to say, when computing the similarity between genes, each gene is regarded as a sample and the conditions are regarded as features. And when computing the similarity between conditions, the roles of the gene and sample would be exchanged.

2.2. Assigning Weight and Sorting with Weighted Features

We use dominant set approach to sort the samples and assign weights to the features. Dominant set is a

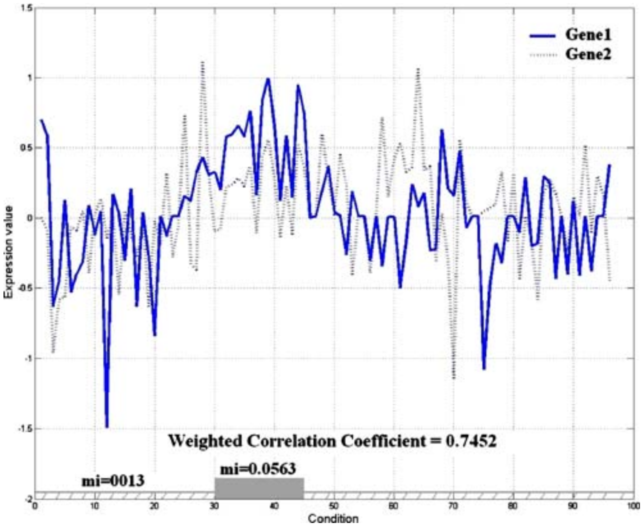


Figure 2. Example of the usage of weighted correlation coefficient.

novel combinatorial concept which arises from the study of a continuous formulation of the maximum clique problem, originally due to Motzkin and Straus [15]. Motivated by the analogies between the intuitive concept of a cluster and that of a dominant set of vertices, it was recently applied to image segmentation by Pavan and Pelillo [12]. Suppose we have an undirected edge-weighted graph $G=(V, E, w)$, where $V=(1, 2, ..., n)$ is the vertex set, E is the edge set, w is the weight function. E and w define the adjacency (or similarity) matrix M, which is a n by n matrix reflecting similarities between pairs of linked vertices. Pavan and Pelillo established a tight correspondence between the problem of finding dominant sets in an edge-weighted graph and that of finding solutions of a quadratic program. They proved that if x^* is a strict local solution of Eq. (3) then its support (the non-zero components) constructs a dominant set. By virtue of this theoretical result, they use the continuous optimization method such as replicator equations, a class of dynamical systems arising in evolutionary game theory, to solve problem (3). Equation (4) is the discrete replicator equation that they used in their work,

$$\text{maximize}\, f(x) = \frac{1}{2}X^T MX, \quad \text{subject to } X \in \Delta, \tag{3}$$

where $\Delta=\{X\in R^n\text{: } X\geq 0 \text{ and } e^T X=1\}$, e is a column vector of appropriate length consisting of unit entries.

$$x_i(t+1) = x_i(t)\frac{(MX(t))_i}{X^T(t)MX(t)}, \tag{4}$$

Where $x_i(t)$ is the ith component of vector X at the tth iteration. Given an initial estimation of vector $X(0)$, we compute Eq. (4) iteratively until the vector converges to a stationary point. The consequent vector $X(t)$ will be a strict local maximizer of program (3) in Δ. Then the vertices with the corresponding $x_i\neq 0$ forms a dominant set.

Xuping Fu et al. [16] used dominant set to clustering the genes. In that case each gene is regarded as a vertex and M is the similarity matrix of genes. They found that the value of x_i could be a criterion to judge whether one gene belongs to a dominant set and the genes with high x_i have stronger tendency to belong to a dominant set. As they sort the genes with the corresponding values of x_i, the genes with high similarities would be rearranged together. Our paper further enhances this dominant set approach to perform biclustering by iteratively refining the weight vectors and hence the similarity matrix during the process.

When we perform biclustering, we aim at finding a subset of genes and samples. Firstly, vector $X(t)$ is used as the sorting vector to sort the genes. After sorting, the genes with high similarities would be arranged at the bottom of the list. Then, vector $X(t)$ is used as the feature weight vector $\mathbf{m}$ in Eq. (2), when the roles of rows and columns exchanged, i.e., we transpose the data matrix and set $\mathbf{m}=X(t)$ when working on the conditions. In this feature weight vector, some features with high similarities across the genes have been assigned higher weights. And here we have $\sum_{i=1}^{n} m_i = 1$ according to Eq. (3), so we can use the following function to calculate the weighted correlation coefficient between samples a and b.

$$r' = \frac{\sum_{i=1}^{n} m_i a_i b_i - \sum_{i=1}^{n} m_i a_i \sum_{i=1}^{n} m_i b_i}{\sqrt{\left(\sum_{i=1}^{n} m_i a_i^2 - \left(\sum_{i=1}^{n} m_i a_i\right)^2\right)\left(\sum_{i=1}^{n} m_i b_i^2 - \left(\sum_{i=1}^{n} m_i b_i\right)^2\right)}}, \tag{5}$$

We use Eq. (5) to compute the similarity matrix M which is used in Eq. (4) for updating $X(t)$. To look for the positive correlated genes together with the negative correlated, we use the absolute value of the weighted correlation coefficient in this paper. This transposing and sorting process could be implemented iteratively for a few times. And in each sorting we use the vector $X(t)$ to sort the samples and use this vector as the feature weight vector in the next sorting.

To illustrate the algorithm Fig. 3 shows an ideal example. The gray image in Fig. 3a shows an 100×50 data matrix where the white elements construct an embedded bicluster. Firstly, we sort the rows with the initial feature importance vector [0.02,0.02,...,0.02]. Figure 3b is the result after the sorting. Then we transpose the data matrix as Fig. 3c shows and sort the new rows using the sorting vector on last step as the new feature importance vector. Ideally, we could get the final data matrix as Fig. 3d shows, where the bicluster is located at the right and bottom of the rearranged data matrix.

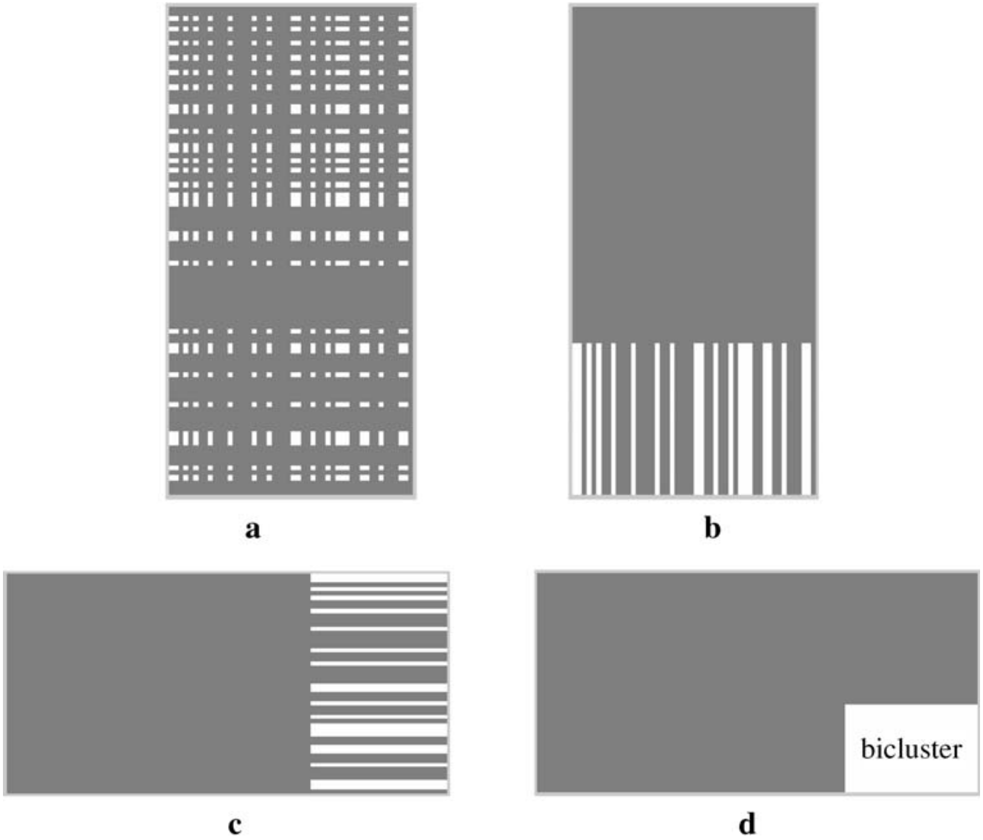


Figure 3. An example of the sorting and transposing process. **a** Original 100×50 data matrix where the white elements construct an embedded bicluster, **b** data matrix after first rearranging on the rows, **c** transpose the data matrix, **d** sort the new rows and get the bicluster in the right bottom corner of the data matrix.

2.3. *The Whole Algorithm*

When we sort the genes (the rows) of the original dataset, the conditions are regarded as features and the genes are regarded as samples. When we sort the conditions, the roles of the genes and conditions are exchanged. This transposing and sorting process could be implemented alternately for a few times. In our empirical results, we find that after two iterations, that means sorting the genes and the samples for twice respectively, good biclusters were observed in the rearranged data matrix most of the time. Next, we have to extract the bicluster from the data matrix. We calculate the correlation coefficients between adjacent rows (columns) and cut the rows (columns) when the correlation coefficient between two adjacent rows (columns) is less than a certain threshold. In this way we can get a bicluster.

The above algorithm enables us to find one bicluster. To find more biclusters, there are two often used approaches. One is to delete the rows and the columns that are involved in the previous bicluster, the other is to replace the corresponding elements by random values and then apply the algorithm again to the data. Instead of the above two approaches, we use another way based on the found biclusters; we update the initial feature weight vector. If a gene or condition has already been contained in a bicluster, the corresponding feature weight would be reduced and at the same time the weight for the feature that has not been grouped would be enhanced. For each gene (condition), we use a flag to stand for whether the gene (condition) has been contained in any bicluster. The initial values for all the flags are zero. When the gene (condition) has been grouped the corresponding flag turns to 1.

Suppose the initial feature weight vector is **m**=[m_1, m_2,..., m_n] with the corresponding flag vector **f**=[f_1, f_2, ..., f_n]. We would update the feature weight vector according to the following function,

$$\begin{aligned} &\text{If } f_i = 1, x_i' = \alpha x_i, \alpha < 1, \\ &\text{else } x_i' = \left(\frac{1-\alpha}{1-\sum_{i=1}^{n} x_i f_i} + \alpha \right) x_i \end{aligned} \tag{6}$$

In this way we decrease the effect of the features that have been contained in previously found biclusters and give more weights to the ungrouped data. The smaller α is, the more change could happen on x and the updated x could lead us to a new bicluster. We find that by using a value of around 0.2 would avoid finding the same bicluster.

Since we do not exclude the previously found biclusters after each run, the biclusters we find could be overlapping. This conforms to the situation that some genes (conditions) could be involved in multiple groups. We stop the algorithm when most of the genes or conditions have been grouped or when no appropriate biclusters could be found. The algorithm is very efficient. The running time of one iteration for Eq. (4) is $O(n^2)$ and the number of iterations is defined to be only four. In this step, more iteration would not give better biclusters. Table 1 summarizes the complete algorithm.

3. Criterion to Evaluate a Bicluster

The additive and multiplicative model of Eq. (7) is widely used to define biclusters with coherent values in the previous work [3]. A perfect biclusters with coherent values is a matrix, where all the values within the matrix follow either of the given expressions in Eq. (7)

$$\begin{aligned} a_{ij} &= \mu + \alpha_i + \beta_j, \\ a_{ij} &= \mu' \times \alpha_i' \times \beta_j', \end{aligned} \tag{7}$$

where μ or μ' is the typical value within the bicluster, α_i or α_i' is the adjustment for row i, β_j or β_j' is the adjustment for column j. These two models can be simplified to Eq. (8) if we use σ_i to replace μ+α_i and use σ_i' to replace $\mu' \times \alpha_i'$.

$$\begin{aligned} a_{ij} &= \sigma_i + \beta_j, \\ a_{ij} &= \sigma_i' \times \beta_j', \end{aligned} \tag{8}$$

The mean squared residue score proposed by Cheng and Church [2] and some similar criteria that based on residue [7–10] have been widely used to evaluate the quality of the biclusters. Suppose A is a

Table 1. The whole algorithm of biclustering.

Pseudocode for the whole procedure of biclustering

```
Input: the initial gene expression data matrix $A$, $f$, $\alpha$ and the Threshold,
WHILE biclusters could be found,
  $(n, m)=size(A)$;
  $\mathbf{m} = \left(\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m}\right)$; // initialize the feature vector for the m conditions.
  For $i$=0:3,
    $\mathbf{m}'=updatem(m, f)$;
    $M=weighted\_corr(A, \mathbf{m}')$; // compute the similarity matrix
    $X=dominantset(M)$; // use dominant set approach to find the sorting vector
    $B=sort(A, X)$; // sort the rows of matrix $A$
    $A=transpose(B)$;
    $\mathbf{m}=X$; //update the feature weight vector
  End
  $[newflag, BI]=find\_bicluster(A, Threshold)$; //extract the bicluster from the sorted data matrix
  Output $BI$;
  $f=newflag$; // update the flag vector
 END
```

n by m matrix. a_{ij} Stands for the element of the ith row and jth column. The mean squared residue score for matrix A is calculated by the following function:

$$H(A) = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\left(a_{ij} - a_{iM} - a_{Nj} + \overline{a_{ij}}\right)^2 \qquad (9)$$

where

$$a_{iM} = \frac{1}{m}\sum_{j=1}^{m} a_{ij}, \quad a_{Nj} = \frac{1}{n}\sum_{i=1}^{n} a_{ij},$$
$$\overline{a_{ij}} = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} a_{ij} \qquad (10)$$

They supposed that a low mean squared residue score plus a large variation from the constant may be a good criterion for identifying a bicluster. And they proposed a greedy approach to find biclusters. For the additive model, $H(A)$=0. While, for the multiplicative model:

$$H(A) = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\left[\left(\beta_j - \frac{\sum \beta_j}{m}\right)\left(\alpha_i - \frac{\sum \alpha_i}{n}\right)\right]^2$$
$$\neq 0, \qquad (11)$$

The values of $H(A)$ in Eq. (11) have no upper bound. A scaling (multiply a row or column by a constant) would easily change its value. Thus, if a bicluster is of the multiplicative model, the mean squared residue score is not a good criterion to evaluate the biclusters of multiplicative model.

In this paper, we propose the average correlation value (ACV) to evaluate the property of a bicluster. A bicluster should be a subset of genes and samples of which the genes or the samples are highly coexpressed. Based on this assumption, the average correlation of the genes or the samples could be a measurement to evaluate the property of a bicluster. The ACV of matrix A could be computed by the following function,

$$\overline{R}(A) = \max\left\{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|r\_\ \mathrm{row}_{ij}\right| - n}{n2 - n}, \frac{\sum_{k=1}^{m}\sum_{l=1}^{m}\left|r\_col_{kl}\right| - m}{m2 - m}\right\},$$
$$\overline{R}(A) \subseteq [0, 1] \qquad (12)$$

Where r_row_{ij} is the correlation between the ith and jth rows of A and r_col_{kl} is the correlation between the kth and lth columns of A. So a high value of $\overline{R}(A)$ means that the rows or columns of A are highly coexpressed.

ACV always gives the desirable value for both the additive model and the multiplicative model. To compare the two criteria, we list seven possible different types of biclusters in Fig. 4. Among them, A1, A2, A3, A4, A5 and A7 are also used in

1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1
1	1	1	1

A1

1	1	1	1
2	2	2	2
3	3	3	3
4	4	4	4
5	5	5	5

A2

1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4
1	2	3	4

A3

1	2	5	0
2	3	6	1
4	5	8	3
5	6	9	4
6	7	10	5

A4

1	2	0.5	1.5
2	4	1	3
4	8	2	6
3	6	1.5	4.5
5	10	2.5	7.5

A5

10	20	5	15
2	4	1	3
4	8	2	6
3	6	1.5	4.5
5	10	2.5	7.5

A6

70	13	19	10
49	40	49	35
40	20	27	15
90	15	20	12
50	38	45	30

A7

Figure 4. Example of different types of biclusters. *A1* Constant bicluster, *A2* constant rows, *A3* constant columns, *A4* coherent values (addictive model), *A5* coherent values (multiplicative model), *A6* coherent values (multiplicative model) and *A7* coherent evolution on the columns.

Table 2. Corresponding mean squared residue score and average correlation value of A1–A7.

	A1	A2	A3	A4	A5	A6	A7
$H(A_i)$	0	0	0	0	0.6250	2.4250	131.8750
$\overline{R}(A_i)$	1	1	1	1	1	1	0.8498

[3]. A1, A2 and A3 are described by the additive model. A5 and A6 are cases of the multiplicative model. And A5 is modified to A6 by magnifying its first row. A7 is regarded as a perfect bicluster with noise. We list the mean squared residue score and ACV of each bicluster in Table 2 respectively.

A1–A6 all have the perfect ACV values for biclusters. And A7 has a fairly high ACV and this could be evidence that A7 is a reasonable bicluster. However, the mean squared residue scores of A5–A7 could not give any indications if the matrices are biclusters or not. Besides, ACV could be a criterion to evaluate the quality of the original dataset and to determine whether further biclustering is necessary. A high ACV means that the conditions or the genes in one dataset are highly correlated in the whole scale. In that case, the biclusters tend to contain all the genes or conditions. Then it is more reasonable and feasible to analyze the data by clustering methods, such as hierarchical clustering or K means clustering, considering the computation cost. And a low ACV means neither the conditions nor the genes are similar in the whole scale and this could be a reason for using biclustering to find highly correlated subsets of genes and conditions.

4. Experimental Results

We have implemented our algorithm in Matlab. We ran it on synthetic as well as real data. The synthetic datasets were used to testify the algorithm. For the real gene expression data experiments we used three datasets; the yeast dataset used by Cheng and Church [2], the human cancer data from [17] and the human fibroblasts to serum dataset from [18]. These three datasets have been well studied in the literature. Our results on the gene expression dataset were compared with that of Cheng and Church [2]. The experiments were executed on Windows with a 3.2GHz CPU and 0.99G main memory. The executive time differs a lot for different datasets. On average, it takes less than few minutes for one run.

4.1. Synthetic Data

As a heuristic algorithm, the property of the data affects the performance of the algorithm. Our algorithm based on the computation of the weighted correlation coefficient among the rows (columns). The size of the dataset and the size of the potential bicluster would be important factors that affect the results. In theory the bigger the bicluster is, the easier it can be found. We tested our approach with different combinations of the size of dataset and the bicluster.

We use the notation of (n, m) to denote the size of the dataset and the bicluster. n Stands for the number of rows and m stands for the number of columns. We randomly constructed two datasets of size (200, 30) and (500, 30) respectively. Different sizes of biclusters were inserted one at a time and for each case we tested for ten runs. Table 3 shows the average result of the size of the bicluster we found in each case. Our approach works well for finding large biclusters. And it fails when the bicluster is very small comparing to the size of the whole dataset because in that case the local similarities is too small to effect and could easily be covered by noise.

4.2. Gene Expression Data

We used three gene expression datasets. The original yeast dataset contain 2,884 genes and 17 conditions.

Table 3. The size of the bicluster we found in different cases.

	Size of bicluster inserted					
Size of dataset	(20, 10)	(40, 15)	(50, 20)	(100, 20)	(150, 20)	(100, 30)
(200, 50)	(16, 8)	(40, 13)	(50, 19)	(100, 20)	(150, 20)	(100, 30)
(500, 50)	Fail	(35, 13)	(50, 19)	(100, 20)	(150, 20)	(100, 30)

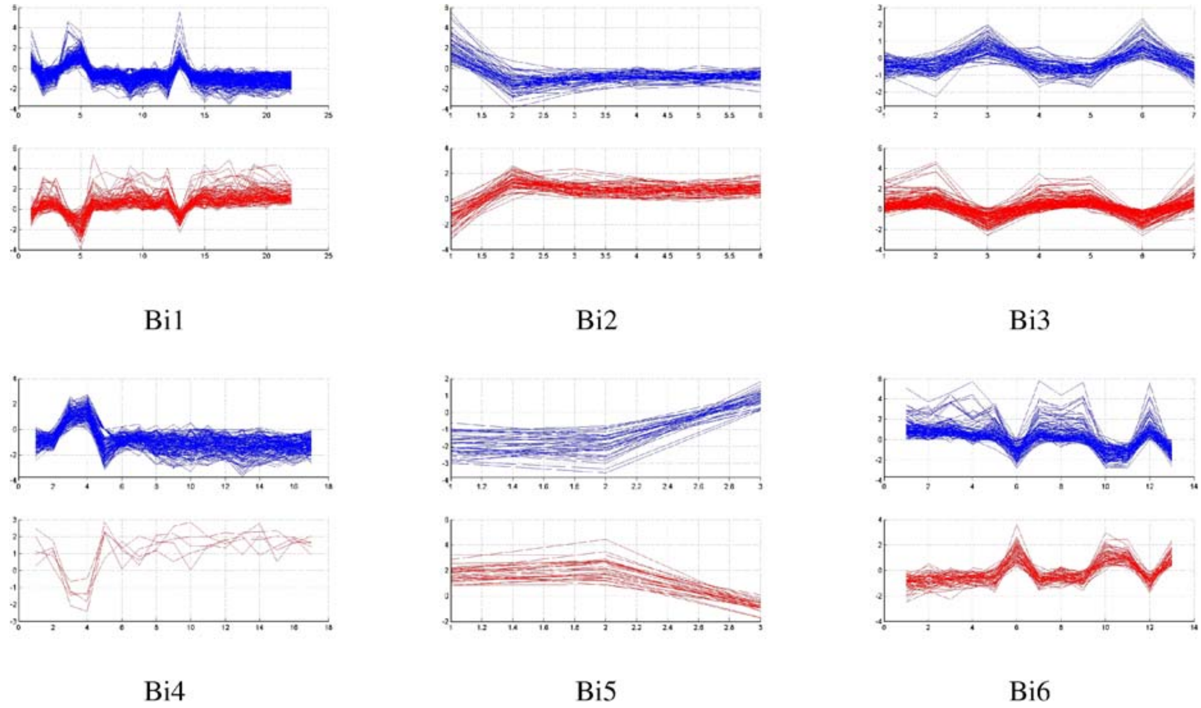
Table 4. Comparison of the three datasets.

	Yeast data set	Human data set 1	Human data set 2
Size	2879×17	4026×96	517×18
Magnitude	[0–595]	[−7.49–6.42]	[0.1–86.92]
Mean squared residue score	1,111.3	0.5837	2.4729
ACV	0.9172	0.1819	0.4306

These genes were selected according to [19] and there were 34 null elements. We deleted the two null rows and three rows that were all zeros. The human data were downloaded from the Web site for supplementary information for the article [17]. There were 4,026 genes and 96 conditions with 19,667 missing values (5.09% of the matrix elements). Missing data in the human dataset were replaced with the value 0.012031 which is the average value of all other elements. These data could be regarded as the background of the whole dataset and would not affect the biclustering result. For the human fibroblasts to serum dataset we used only a subset of size 517×18 which is one of the clustering results of [20].

4.3. Results Analysis

Table 4 shows some basic statistics of the three datasets. Human data set 1 stands for the human cancer data and human data set 2 stands for human fibroblasts to serum dataset. The mean squared residue score and ACV of the three datasets were compared respectively. The scale for the three datasets has a large variance and the mean squared residue score differs a lot according to the data scale. This poses challenges to setting the parameters when using the mean squared residue score as the biclustering criterion. The yeast dataset has a very high ACV of 0.9172 for all the conditions of the dataset are highly correlated with a minimum value of 0.8404. In the result of Cheng and Church [2], most of the biclusters of this dataset contain more than 85% of all the conditions, especially the biclusters found at early time. That means many genes are nearly coexpressed under all the conditions. It is more reasonable to analyze the yeast dataset by clustering methods considering the computation cost and similar result would be achieved. The two human datasets have low ACV. That means the conditions or genes in the two datasets do not



Bi1

Bi2

Bi3

Bi4

Bi5

Bi6

Figure 5. The first six biclusters found in the human cancer data.

Table 5. Biclustering result of the human cancer dataset.

	Original	Bi1	Bi2	Bi3	Bi4	Bi5	Bi6
Size	4026×96	313×22	125×6	208×7	146×17	63×3	157×13
ACV	0.182	0.77	0.89	0.81	0.76	0.96	0.84
MSRS	0.58	0.73	1.38	0.72	0.37	1.80	1.1

MSRS Mean squared residue score.

correlated in the whole scale and we can only find coexpressed subset of genes and conditions by biclustering.

In the original human cancer dataset, there are 62 patient samples and 34 normal samples. And the patient samples can be further categorized into three groups [17]. Figure 5 shows the first six biclusters we find in the human cancer data. Bi1 to Bi6 stands for the first six biclusters respectively. In the figures, the horizontal axis is the index of conditions and the vertical axis is the magnitude of the expression value. In each bicluster there are two types of genes; one is positively correlated and the other is negatively correlated. Table 5 shows that our algorithm tends to find big biclusters first. And the resulted biclusters contain reasonable portions of genes and conditions. The biclusters have very high ACV. This demonstrates the correctness of finding the interesting biclusters. In the 22 conditions of Bi1, apart from three blood B cells, the other 19 conditions are all from patient samples (four DLCL samples, five FL samples and ten CLL samples). For the three subgroups of the patient samples, 10 out of 11 CLL samples were included. In the result of [17], the FL and CLL samples are clustered very closely under the whole gene level, while in our results some DLCL samples were also find coexpressed under a subset of genes with part of FL samples and nearly all the CLL samples. In Bi2 there are five patient samples and one normal sample. All the conditions of Bi3 are patient samples and Bi4 contains 15 patients samples and two normal samples. That shows the biclusters have much homogeneity. And under the restriction of whole-level similarity, clustering could not discover this result. In the result of Cheng and Church [2] most biclusters have conditions from more than primary clusters on conditions. However, it is difficult to compare the results from different methods because there is no unique solution for the datasets. Further, smaller biclusters tend to be more homogeneous. In our result, the biggest bicluster has a high ACV of 0.77, but the highest ACV happens for Bi5 which has the smallest size.

Figure 6 shows three biclusters we found in the human fibroblasts to serum dataset. Table 6 shows the properties of the biclusters. In most cases, the original dataset has a comparably small mean squared residue score, while the biclusters we find have larger mean squared residue score. That's because the biclusters we found have larger variance in the data and that doesn't conform to the criterion of mean squared residue score. So our algorithm is more applicable when noises exist.

5. Discussion

In this paper we have proposed a biclustering method based on alternately sorting and transposing the gene

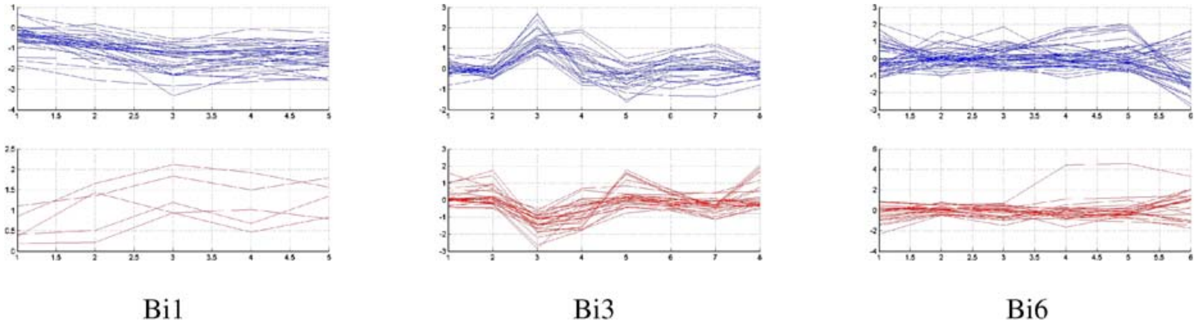


Figure 6. Three biclusters in the human fibroblasts to serum dataset.

Table 6. Biclustering result of the human fibroblasts to serum dataset.

	Original	Bi1	Bi3	Bi6
Size	517×18	36×5	51×8	64×6
ACV	0.485	0.840	0.721	0.631
MSRS	0.474	0.142	0.446	0.429

MSRS Mean squared residue score.

expression data matrix and using weighted correlation coefficient to measure the similarities. By assigning higher weights to the potentially correlated features and then finding a subset of samples which have high weighted correlation coefficient, we can locate a highly correlated bicluster on the lower right corner of the rearranged gene expression data matrix. By using correlation coefficient based measurement, we capture similarity in “trend” but place no emphasis on the magnitude. This makes normalization not a prerequisite as normalization adds extra computational cost and sometimes changes the overall data structure which could affect the result of biclustering to a large extent. Using this framework, we do not break the whole structure of the data matrix as what the other two-way clustering method or divide and conquer methods do. In addition, we allow the finding of multiple biclusters without excluding the previously found biclusters. So it is possible for us to find overlapping biclusters and this conforms to the biological assumption that some genes (conditions) could be involved in multiple cellular processes.

We find that dominant set is a perfect algorithm for the framework of our approach. We can use it to sort the samples and assign the feature weights at the same time. The biclusters we found contain adequate portions of genes and conditions. By using the absolute value of the correlation coefficient, we could find both negative relativity and positive relativity in one bicluster. The algorithm is flexible to use in different applications. For example, we can concentrate on only the positive related genes by using a different form of the weighted correlation coefficient values. The algorithm is also easy to run without much computation cost comparing to the previous algorithms.

In Table 7, our approach is compared with some other methods theoretically. In fact, since the problem of finding the best bicluster is NP-hard [3], all methods resort to heuristics that are not easily amenable to direct complexity analysis. Iterative methods, used in the large majority of the approaches, are heavily dependent not only on the computational complexity per iteration that can be computed in a relatively straightforward way, but also on the number of iterations necessary to reach a solution. The number of iterations cannot, in general, be defined a priori, and this represents the major factor constraining the efficiency of these methods.

Inspired by the intuitive biological notion of finding coexpressed subset of genes or samples, we define the average correlation value (ACV) to evaluate the property of a cluster. This criterion is more general than the mean squared residue score. It measures biclusters of both the additive model and the multiply model. Constant biclusters are tended to be found when using the mean squared residue score. And the mean squared residue score has a big variance according to the magnitude of the dataset. This causes difficulties in setting the specific parameters to different datasets. Although normalization methodshave been used to handle the problem, they add more computation costs. And there has no general accepted normalization method yet. Besides, by applying ACV to the original dataset to measure the homogeneousness of the data,

Table 7. Comparison between several methods.

	Our approach	δ-Biclusters [2]	Plaid models [11]	OP clusters [21]
Approach	Greedy	Greedy	Dist–Ident	Greedy
Bicluster type	Additive and multiplicative model	Additive model	Additive and multiplicative model	Order preserving patterns
Discovery	One at a time	One at a time	One at a time	Simultaneous
For one iteration	$O(m^2)$	$O(nm)+O(nm)+O(\log n+\log m)$	–	–

we can also determine whether biclustering is needed to be carried out or not. The dataset with large ACV does not need any further biclustering, for genes or samples in that dataset are highly correlated almost in the whole scale. This conforms to the assumption of clustering methods, and this could help us to save computational cost.

Acknowledgment

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administration Region, China.

References

1. J. Hartigan, "Clustering Algorithms," Wiley, 1975.
2. Y. Cheng and G. Church, "Biclustering of Expression Data," in *Proc. Eighth Int'l Conf. Intelligent Systems for Molecular Biology (ISMB'00)*, 2000, pp. 93–103.
3. S.C. Madeira and A.L. Oliveira, "Biclustering Algorithms for Biological Data Analysis: A Survey," *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, vol. 1, no. 1, 2004, pp. 24–45.
4. G. Getz, E. Levine and E. Domany, "Coupled Two-Way Clustering Analysis of Gene Microarray Data," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 97, 2000, pp. 12079–12084.
5. C. Tang, L. Zhang, I. Ahang and M. Ramanathan, "Interrelated Two-Way Clustering: An Unsupervised Approach for Gene Expression Data Analysis," in *Proc. Second IEEE Int'l Symp. Bioinformatics and Bioeng.*, 2001, pp. 41–48.
6. J.A. Hartigan, "Direct Clustering of a Data Matrix," *J. Am. Stat. Assoc. (JASA)*, vol. 67, no. 337, 1972, pp. 123–129.
7. H. Cho, I.S. Dhillon, Y. Guan and S. Sra, "Minimum Sum-Squared Residue Cococlustering of Gene Expression Data," in *Proc. Fourth SIAM Int'l Conf. Data Mining*, 2004.
8. J. Yang, W. Wang, H. Wang and P. Yu, "δ-Clustering: Capturing Subspace Correlation in a Large Data Set," in *Proc. 18th IEEE Int'l Conf. Data Eng.*, 2002, pp. 517–528.
9. J. Yang, W. Wang, H. Wang and P. Yu, "Enhanced Biclustering on Expression Data," in *Proc. Third IEEE Conf. Bioinformatics and Bioeng.*, 2003, pp. 321–327.
10. H. Wang, W. Wang, J. Yang and P.S. Yu, "Clustering by Pattern Similarity in Large Data Sets," in *Proc. 2002 ACM SIGMOD Int'l Conf. Management of Data*, 2002, pp. 394–405.
11. L. Lazzeroni and A. Owen, "Plaid Models for Gene Expression Data," Technical Report, Stanford University, 2000.
12. M. Pavan and M. Pelillo, "A new Graph-Theoretic Approach to Clustering and Segmentation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2003, pp. 3068–3077.
13. J.M. Bland and D.G. Altman, "Calculating Correlation Coefficients with Repeated Observations: Part 2–Correlation Between Subjects," *BMJ*, vol. 310, 1995, p. 633.
14. M.B. Eisen, P.T. Spellman, P.O. Brown and D. Botstein, "Cluster Analysis and Display of Genome-Wide Expression Patterns," *Proc. Natl. Acad. Sci. U.S.A.*, vol. 95, 1998, pp. 14863–14868.
15. T.S. Motzkin and E.G. Straus, "Maxima for Graphs and A New Proof of A Theorem of Turan," *Can. J. Math.*, vol. 17, 1965, pp. 533–540.
16. X. Fu, L. Teng, Y. Li, W. Chen, Y. Mao, I.-F. Shen and Y. Xie, "Finding Dominant Sets in Microarray Data," *Front. Biosci.*, vol. 10, 2005, pp. 3068–3077.
17. A.A. Alizadeh, M.B. Eisen, R.E. Davis, C. Ma, I.S. Lossos, A. Rosenwald, J.C. Boldrick, H. Sabet, T. Tran, X. Yu, J.I. Powell, L. Yang, G.E. Marti, T. Moore, J. Hudson, L. Lu, D.B. Lewis, R. Tibshirani, G. Sherlock, W.C. Chan, T.C. Greiner, D.D. Weisenburger, J.O. Armitage, R. Warnke, R. Levy, W. Wilson, M.R. Grever, J.C. Byrd, D. Botstein, P.O. Brown and L.M. Staudt, "Distinct Types of Diffuse Large B-Cell Lymphoma Identified by Gene Expression Profiling," *Nature*, vol. 403, 2000, pp. 503–510.
18. V.R. Iyer, M.B. Eisen, D.T. Ross, G. Schuler, T. Moore, J.C.F. Lee, J.M. Trent, L.M. Staudt, J. Hudson Jr., M.S. Boguski, D. Lashkari, D. Shalon, D. Botstein and P.O. Brown, "The Transcriptional Program in the Response of Human Fibroblasts to Serum," *Science*, vol. 283, 1999, pp. 83–87.
19. S. Tavazoie, J.D. Hughes, M.J. Campbell, R.J. Cho and G.M. Church, "Systematic Determination of Genetic Network Architecture," *Nat. Genet.*, vol. 22, 1999, pp. 281–285.
20. X.L. Ji, L.L. Jesse and Z.R. Sun, "Mining Gene Expression Data Using a Novel Approach Based on Hidden Markov Models," *FEBS Lett.*, vol. 542, 2003, pp. 125–131.
21. J. Liu and W. Wang, "OP-Cluster: Clustering by Tendency in High Dimensional Space," in *Proc. Third IEEE Int'l Conf. Data Mining*, 2003, pp. 187–194.

Li Teng received her BS degree in computer science & technology from Huazhong University of Science and Technology in 2002, and M.S. degree in computer science and engineering from Fudan University. She is currently a Ph.D. student at the CSE Department of the Chinese University of Hong Kong. Her research interests include bioinformatics, data mining and machine learning.
Iteng@cse.cuhk.edu.hk

Lai-wan Chan received her B.A., M.A. and Ph.D. degrees in Engineering from the University of Cambridge. She is currently a Professor in the Computer Science and Engineering Department, and the Associate Dean (Education) in the Faculty of Engineering. Her research interests are in data mining, financial engineering, bioinformatics and artificial neural networks. She has designed and applied various neural networks and ICA techniques in engineering and bioinformatics to perform temporal and spatial data analysis. Applications include portfolio management, risk analysis, trading systems, gene expression data analysis and time series analysis. Other research works include the learning and modelling of the feedforward and the recurrent networks, the modular learning methods, and applications of neural networks in image recognition, Cantonese speech recognition and multimedia database. Professor Chan has published more than 100 scientific papers in the above research areas.
Iwchan@cse.cuhk.edu.hk