

A NEW STRATEGY OF GEOMETRICAL BICLUSTERING FOR MICROARRAY DATA ANALYSIS*

HONGYA ZHAO

Department of Electronic and Engineering, City University of Hong Kong, Hong Kong

ALAN W. C. LIEW

Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong

HONG YAN

*Department of Electronic and Engineering, City University of Hong Kong, Hong Kong
School of Electronic and Information Engineering, University of Sydney, NSW2006, Sydney,
Australia*

In this paper, we present a new biclustering algorithm to provide the geometrical interpretation of similar microarray gene expression profiles. Different from standard clustering analyses, biclustering methodology can perform simultaneous classification on the row and column dimensions of a data matrix. The main object of the strategy is to reveal the submatrix, in which a subset of genes exhibits a consistent pattern over a subset of conditions. However, the search for such subsets is a computationally complex task. We propose a new algorithm, based on the Hough transform in the column-pair space to perform pattern identification. The algorithm is especially suitable for the biclustering analysis of large-scale microarray data. Our simulation studies show that the method is robust to noise and computationally efficient. Furthermore, we have applied it to a large database of gene expression profiles of multiple human organs and the resulting biclusters show clear biological meanings.

1 Introduction

DNA microarray technology is a high-throughput and parallel platform that can provide expression profiling of thousands of genes in different biological conditions, thereby enabling the rapid and quantitative analysis of gene expression patterns on a global scale. It aids the examination of the integration of gene expression and function at the cellular level, revealing how multiple gene products work together to produce physical and chemical responses to both static and changing cellular needs [14]. As an increasing number of large-scale microarray experiments are carried out, analysis of the expression data produced by these experiments remains a major challenge. A key step of the analysis is the identification of groups of genes that exhibit similar expression patterns. Therefore cluster analysis has emerged as one of the most valuable tools to elicit complex structures and gather information about how genes work in combination with microarray data. A large number of clustering methods have been proposed for the analysis of gene function

* This work is supported by grant 9010003 of City University of Hong Kong interdisciplinary research and grant CityU122005 of Hong Kong Research Grant Council.

on a global scale [17]. Usually, gene expression data are arranged in a matrix, where each gene corresponds to one row and each condition to one column. Each element of the matrix represents the expression level of a gene under an experimental condition. Thus, clustering methods can be applied to group genes by comparing rows or conditions by comparing columns. However, conventional clustering methods have their limitations: they require that the related genes (conditions) behave similarly across all measured conditions (genes) in one cluster. In fact, many activation patterns are common to a group of genes only under specific experimental conditions. As such, an interesting cellular process may be involved in a subset of genes co-regulated or co-expressed only under a subset of conditions, but to behave almost independently under other conditions. Discovering such local expression patterns may be the key to uncovering many genetic pathways that are not apparent otherwise. Thus it is highly desirable to move beyond the clustering paradigm, and to develop approaches capable of discovering local patterns in microarray data.

Beyond the traditional clustering method, the term ‘biclustering’, also called coclustering, bidimensional clustering, and subspace clustering, was first formulated by Hartigan [6]. It was first applied to expression matrices for simultaneous clustering of both genes and conditions by Cheng and Church [3]. Since then different kinds of algorithm are proposed [5, 9, 11, 12]. And bicluster was recently summarized in two papers [10, 15]. The general strategy in these algorithms can be described as adding or deleting rows and/or columns in the data matrix in some optimal ways such that a merit function is improved by the action. In contrast, a different viewpoint of the biclustering is in terms of the spatial geometrical distribution of points in data space [4]. The biclustering problem is tackled as the identification and division of coherent sub-matrices of data matrices into geometrical structures (lines or planes) in data space. This novel perspective opened a door to the performance of biclustering using the methodology of detecting the geometric lines or planes within a unified framework. In the framework, a series of the well-known Hough transforms are conducted to detect lines and planes. No explicit cost function is required to define the procedure. As such, the Hough transform is noted for its ability to detect lines and planes in noisy data [7]. Thus, it is especially suitable for biclustering analysis since noise is one of the major issues in microarray data. However, if the number of conditions is small, the speed of the geometric biclustering algorithm is acceptable. With the augmentation of the dimension, vote accumulators in the Hough transform use so much memory that the computation time is significantly increased and become ineffective. In order to overcome the difficulty, a novel strategy is proposed in this paper based on geometric biclustering. In our algorithm, the Hough transform is only performed in the column-pair space. Instead of computing all genes, only useful genes (features) are extracted for the combination of the following iterations.

The paper is organized as follows. First, we demonstrate that all biclustering patterns of interest in data matrices can be formulated with the linear relation in column-pair space. Based on this premise, a visualization tool, the AMPP plot, is proposed to separate the genes into different groups of biclusters. Then, the complete algorithm is given on the

basis of the Hough transform and AMPP in Sec. 3. The characteristics of the algorithm are discussed in simulation study. Lastly, we apply the algorithm to bicluster the microarray expression matrix of multiple human organs. The genes in the different biclusters are further analyzed with the gene ontology (GO) tool to infer their biological process, molecular function and cellular component.

2 Linear Pattern of Biclusters in Column-pair Space

An interesting criterion to evaluate in a biclustering algorithm is the identification of type of biclusters. In this paper we focus on five major classes corresponding to significant gene expression. Table 1 shows five different types of biclusters that are of interest in microarray analysis. These biclusters are: (a) constant bicluster, (b) constant rows, (c) constant columns, (d) additive coherent values, where each row or column can be obtained by adding the constant to another row or column, (e) multiplicative coherent values, where each row or column can be obtained by multiplying another row or column by a constant value. In the case of gene expression data, constant biclusters reveal subsets of genes with similar expression values within a subset of conditions. A bicluster with constant values in the rows identifies a subset of genes with similar expression values across a subset of conditions, allowing the expression levels to differ from gene to gene. Similarly, a bicluster with constant columns identifies a subset of conditions within which a subset of genes present similar expression values assuming that the expression values may differ from condition to condition. However, one may be interested in identifying more complex relations between the genes and the conditions, such as coherent values on both rows and columns. In these cases, we can consider additive and multiplicative relations between rows or columns.

Obviously, it is unnecessary to show the relation of all columns together within a bicluster. It is enough to describe a bicluster pattern using an equation of two variables, as shown in the bottom rows of Table 1. Furthermore, it is advantageous to bicluster microarray data matrices in a column-pair space. Firstly, it is obvious that the first three classes of biclusters are special cases of the additive and multiplicative models when $b_{ij} = 0$ or $a_{ij} = 1$ in the column-pair space. Secondly, all five patterns in Table 1 can be generalized into the linear relation $x_j = a_{ij} x_j + b_{ij}$ although they appear to be substantially different from each other. Little attention is paid to the equation with two parameters because there is no corresponding biological meaning in gene expression. As such, we are more interested in the additive and multiplicative patterns, which are described by $x_j = x_j + b_{ij}$ and $x_j = a_{ij} x_j$, respectively. Thirdly, instead of the computation for all genes and conditions, the computation complexity and time are significantly decreased and become operable in the column-pair space. Of course, it is absolutely necessary to compare all pairs of conditions and combine the similar subblocks in order to identify biclusters. Compared to other methods, the geometric perspective we present here allows us to better detect linear relations that define various different bicluster patterns using a generic linear finding algorithm. The algorithm is provided in Sec. 3.

Table 1. Classes of different biclusters: (a) Constant bicluster (b) Constant rows (c) Constant columns (d) Additive coherent values (e) Multiplicative coherent values.

Constant			
x ₁	x ₂	x ₃	x ₄
50	50	50	50
50	50	50	50
50	50	50	50
50	50	50	50
x _i =x _j =a			

Constant Rows			
x ₁	x ₂	x ₃	x ₄
10	10	10	10
25	25	25	25
35	35	35	35
50	50	50	50
x _i =x _j			

Constant Columns			
x ₁	x ₂	x ₃	x ₄
10	25	35	50
10	25	35	50
10	25	35	50
10	25	35	50
x _i =a _i , x _j =a _j (a _i ≠ a _j)			

Additive			
x ₁	x ₂	x ₃	x ₄
15	12	20	35
20	17	25	40
25	22	30	45
30	27	35	50
x _i =x _i , b _{ij} (b _{ij} ≠ 0)			

Multiplicative			
x ₁	x ₂	x ₃	x ₄
2	6	4	12
5	15	10	30
6	18	12	36
8	24	16	48
x _j =a _{ij} x _i (a _{ij} ≠ 1)			

3 Geometric Algorithm of Biclustering

Based on the linear structures discussed above, we propose a new biclustering algorithm. First we identify genes of interest with linear structures discussed above and divide them into different patterns using the additive and multiplicative pattern plot (AMPP) described below in the column-pair space. Then these genes in the same patterns are combined step by step to form new biclusters. A robust method of line detection in the column-pair space is a key step in the proposed framework. The Hough transform (HT) is an effective, powerful, and robust technique widely used for line detection in 2-D images [1]. In this section, we first introduce the HT and then propose the AMPP as a visualization tool to separate the genes into corresponding additive and multiplicative patterns. The biclustering algorithm will then be developed based on the HT and AMPP.

3.1 Hough Transform and Line Detection

The Hough transform is a methodology that detects analytic lines and curves in images through a voting process in parameter space [1]. A line in x-y data space is defined by

$$y = kx + b \quad (1)$$

Note that a line in x-y space as defined by Eq. (1) corresponds to point (k,b) in k-b parameter space. Conversely, the line in Eq. (1) in k-b space corresponds to point (x,y) in x-y space. If n points $\{(x_i, y_i): i=1, \dots, n\}$ on a line in the x-y space are known, the line obtained from each such point should pass through the same point in k-b space, which is the point defining the line in x-y space. Therefore, to determine lines from points, we can initialize all entries of k-b space to 0 and increment an entry by 1 when the line representing a point in x-y space passes through it, and then find the entry in k-b space that has the highest count. If more than one line is to be detected, entries with local peak counts in k-b space are located and their coordinates are used as the slopes and y-intercepts of the line. The accumulator array in parameter space may be very large

because the range of the slope is large, especially for vertical lines. Alternatively, the polar form can be used to describe a line:

$$y = x \cos \theta + \rho \sin \theta \quad (2)$$

where ρ is the distance of a line to the original point and θ is the angle of the normal to the line with the x axis. Since ρ is limited from $-\sqrt{x^2 + y^2}$ to $\sqrt{x^2 + y^2}$ and θ is limited from $-\pi/2$ to $\pi/2$, the dynamic ranges of the parameters are compressed and a small accumulator array is sufficient to find all lines. Note that if the polar equation of a line is used, for each point in x-y space, a sinusoidal curve rather than a line can be drawn in the accumulator array. Again, array entries with local peak counts should be identified and used to detect the lines [7, 18].

3.2 Additive and Multiplicative Pattern Plot (AMPP)

Given points on a line in column-pair data space, we need to classify their corresponding genes into the additive or multiplicative patterns. We develop a visualization tool, named the AMPP for this task. As discussed in Sec. 2, only the additive and multiplicative patterns are of our interest in microarray analysis, so the difference in the patterns of gene expression is of concern. For example, given $\{(x_i, y_i): i=1, \dots, k\}$ which are the expressions of k genes under two conditions, we assume that the k points are on a line detected using the HT. Now we try to cluster them into two types of expression patterns. We employ $d_i = x_i - y_i$ and $r_i = \arctan(x_i/y_i)$ to show the difference in the additive and multiplicative models. Again, we use $r_i = \arctan(x_i/y_i)$ instead of the direct ratio x_i/y_i to reduce the dynamics range of the ratio. We plot d_i against r_i ($i=1, \dots, k$) in the AMPP. In the plot, the horizontal axis represents the change of additive patterns, and the vertical axis the multiplicative patterns. Based on the AMPP, we employ the boxplots to obtain the points in the additive and multiplicative models. The Boxplot, also called box-and-winker plot, was first proposed by John Tukey, as simple graphical summaries of the distribution of variables [2]. In a boxplot, the middle line represents the median, and the lower and upper sides of the rectangle show the medians of the lower and upper halves of the data. Along the horizontal boxplot, the points in the box are considered to be shifted with their median in the additive model and the points in box of the vertical boxplot are considered to be multiplied by their median in multiplicative model. The points in their intersect set are considered as the overlapped genes in the two patterns. The method is used in the following algorithm to recognize patterns after line detection with HF in column-pair space.

3.3 Biclustering Algorithm

Given expression data matrix $D_{N \times n}$ with N genes and n experimental conditions, we denote the index of rows (genes) with $\mathbf{G} = \{g_1, \dots, g_N\}$ and the index of columns (conditions) with $\mathbf{C} = \{c_1, \dots, c_n\}$. We denote the expression matrix with the row and

column index as $\mathbf{D}=(\mathbf{G},\mathbf{C})$, then the bicluster is defined as $\mathbf{B}=(\mathbf{I},\mathbf{J})$, where

$\mathbf{I}=\{g_{i_1},\dots,g_{i_s}\}$ is a subset of \mathbf{G} and $\mathbf{J}=\{c_{j_1},\dots,c_{j_r}\}$ is a subset of \mathbf{C} . Based on the line detection with the HF and AMPP in the column-pair space, we propose the following algorithm to identify a set of biclusters $\{\mathbf{B}_k:\mathbf{B}_k=(\mathbf{I}_k,\mathbf{J}_k)\}$.

Parameters to be predetermined:

- Resolutions: quantization step size for voting in parameter space;
 - Minimum number of rows δ (genes to form one bicluster);
 - Minimum number of columns ζ (conditions to form one bicluster);
1. Select any two columns from \mathbf{C} as $\mathbf{J}_s=\{c_{s1},c_{s2}\}$, where $s=1,\dots,M=\binom{\|\mathbf{C}\|}{2}$, where $\|\mathbf{C}\|$ denotes the number of elements in the set \mathbf{C} , and transform each of $\{\mathbf{B}_s:\mathbf{B}_s=(\mathbf{G},\mathbf{J}_s)\}$ in the column-pair space to the polar parameter space.
 2. Given \mathbf{J}_s in $\mathbf{B}_s=(\mathbf{G},\mathbf{J}_s)$, there are $\|\mathbf{G}\|$ sinusoidal curves corresponding to \mathbf{B}_s in the parameter space. Then perform a voting count in the quantized parameter space to find the accumulator array p_s . Similar procedures are applied M times to \mathbf{J}_s . Denote the series of the curves passing p_s as \mathbf{G}_s , their corresponding accumulated count as $\|\mathbf{G}_s\|$ and $\mathfrak{R}=\arg\left\{\|\mathbf{G}_s\|\geq\delta:s=1,\dots,M\right\}$, we set the corresponding bicluster as $\mathbf{B}_r=(\mathbf{G}_r,\mathbf{J}_r)(r\in\mathfrak{R})$ in the column-pair space.
 3. With the help of AMPP in the column-pair space, we separate the genes in \mathbf{G}_r into three parts of the additive set \mathbf{I}_r^A , multiplicative \mathbf{I}_r^M and their overlap \mathbf{I}_r^{AM} . Therefore, three patterns are obtained, denoted as $\mathbf{B}_r^A=(\mathbf{I}_r^A,\mathbf{J}_r^A)$, $\mathbf{B}_r^M=(\mathbf{I}_r^M,\mathbf{J}_r^M)$, and $\mathbf{B}_r^{AM}=(\mathbf{I}_r^{AM},\mathbf{J}_r^{AM})$ where $\mathbf{J}_r=\mathbf{J}_r^A=\mathbf{J}_r^M=\mathbf{J}_r^{AM}$ ($r\in\mathfrak{R}$).
 4. First we consider the combination steps for the additive pattern. Set $i=1$ and begin with the set of $\mathbf{B}_i^A=\{(\mathbf{I}_i^A,\mathbf{J}_i^A):\mathbf{I}_i^A=\mathbf{I}_r^A,\mathbf{J}_i^A=\mathbf{J}_r^A\}$ including $\|\mathfrak{R}\|$ subclusters.
 5. We unite any two elements $\mathbf{B}_{iu}^A=(\mathbf{I}_{iu}^A,\mathbf{J}_{iu}^A)$ and $\mathbf{B}_{iv}^A=(\mathbf{I}_{iv}^A,\mathbf{J}_{iv}^A)$ of \mathbf{B}_i^A every time. We consider their intersection of rows and union of columns as a new subcluster. Denote the biclusters $\mathbf{B}_{i+1}^A=\{(\mathbf{I}_{i+1}^A,\mathbf{J}_{i+1}^A):\mathbf{I}_{i+1}^A=\mathbf{I}_{iu}^A\cap\mathbf{I}_{iv}^A,\mathbf{J}_{i+1}^A=\mathbf{J}_{iu}^A\cup\mathbf{J}_{iv}^A\}$.
 6. Repeat Step (5) until there is no new combined subcluster and $\|\mathbf{I}_i^A\|\geq\delta,\|\mathbf{J}_i^A\|\geq\zeta$. From $\mathbf{B}^A=\{(\mathbf{I}^A,\mathbf{J}^A):\|\mathbf{I}^A\|\geq\delta,\|\mathbf{J}^A\|\geq\zeta,\mathbf{B}^A\in\mathbf{B}_k^A,k=1,\dots,i-1\}$ we consider the biclusters obtained from the last step as the largest one.

7. As to the other two cases, $\{\mathbf{B}_r^M : r \in \mathfrak{R}\}$ and $\{\mathbf{B}_r^{AM} : r \in \mathfrak{R}\}$, the combination steps are similar.

4 Simulation Study

Gene expression data from microarray experiments are often degraded by noise. Furthermore, it is important to find the multiple biclusters with overlapping patterns. Therefore, in the simulation study, the following two questions were investigated: robustness to noise and ability to identify the multiple overlapping biclusters.

4.1. Synthetic Data with Noisy Biclusters

Here we investigate the performance of our algorithm on noisy data. We use an additive pattern as an example. We embed an additive bicluster pattern of 20 rows by 8 columns into a dataset of size 100 by 30. One column of the additive pattern is generated from $U(-5, 5)$ and the additive factors of other columns are randomly obtained from $U(-5, 5)$. The background is also generated from the uniform distribution $U(-5, 5)$. Gaussian noise with variance from 0.3 to 0.9 is generated to degrade additive patterns in the bicluster. We apply the new algorithm to the simulated data with parameters $\delta = 15$ and $\zeta = 5$. It is anticipated that our algorithm is robust to noise. In fact, in HT the accumulator arrays are used instead of a point, which accommodates the noise in the data, and thus all genes of interest are already included in the biclusters in the column-pair space. With the combination of clusters step by step, the exact 20x8 additive biclusters are identified after six iterations. Besides the additive bicluster one multiplicative bicluster is also discovered with our methodology, however, it is completely overlapped by the additive bicluster. Thus, more patterns may be discovered with our algorithm.

4.2. Synthetic Data with Multiple Overlapping Biclusters

To show that our algorithm can successfully detect biclusters of different types, we have generated biclusters with additive, multiplicative and their overlapping patterns. Furthermore, we have also examined the ability of our algorithm to simultaneously identify multiple biclusters, especially when the overlap is present. We embed two overlapped biclusters into noisy background generated by the uniform distribution $U(-5, 5)$. Gaussian noise with variance from 0.3 to 0.9 is generated to degrade the bicluster data. The dataset has 100 rows by 30 columns, and two embedded biclusters have the following sizes: 30x6 Bicluster 1 of additive pattern, 25x8 Bicluster 2 of multiplicative and their overlap is a 10x3 submatrix. The random row and column permutations are then performed to obtain the final testing dataset. In this experiment, Bicluster 1 is found with all six conditions and 27 rows. Bicluster 2 is found with eight conditions and 24 rows. The overlap part is identified with all three columns and 10 rows.

5 Applications

We apply our algorithm to the gene expression database of multiple human organs [13]. The database captures 18,927 unique genes for 19 different organs from 158 normal human tissues from 30 donors. In fact, there are several replicated tissues for every organ and they are considered as the replicated samples of the same organ. So in every organ we take the median of the replicated measurements of every gene for further analysis. We also filter the genes with low variance and entropy methods and have obtained one 5298x19 expression matrix of 5298 genes under 19 experimental conditions for the following bicluster analysis. In the first step, $19 \times 18 / 2 = 171$ subclusters can be obtained in the column-pair space. In all these subclusters, the number of columns is two and that of rows is the largest number of lines passing one accumulator array after HT in the corresponding parameter space. We demonstrate all subclusters in Fig. 1. The indexes of rows and columns in the square show 19 different organs. The values of the cross points are the number of genes in every bicluster in the column-pair space. We set the diagonal value to zero. Obviously, the square matrix is symmetric. We use different gray scales to represent different count values. The darker the color is, the larger the value is. For example, the largest value in the square matrix is 468 between the comparison of colon and ileum, that is, their gene expression patterns are very similar, which are in logical agreement with known organ functions.

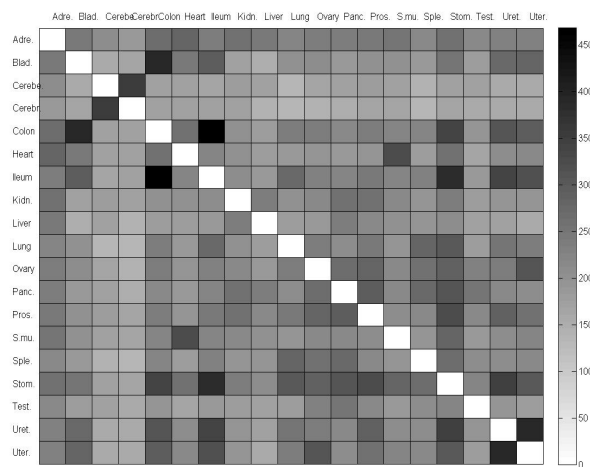


Figure 1. Heat map of the symmetric square matrix of highest count in the column-pair space. The rows and columns represent 19 organs.

In the following steps, we combine the small subclusters in the column-pair space into larger biclusters. We discover that the results of the procedures coincide with the corresponding organ function. For example, we combine colon, ileum, bladder and stomach into one significant bicluster with the largest number of common genes in some iteration. In the combination steps, the number of conditions is increased and the number

of genes is decreased until the stop criteria. At last, one additive and six multiplicative biclusters are recorded with the parameters $\delta = 25$ and $\zeta = 5$. The largest additive and one multiplicative biclusters are shown in Fig. 2

Given the biclusters, we try to test the results and infer their organ functions. Gene Ontology (GO) has become a well accepted standard in organizing gene function categories [16]. Thus GO provides us with a systematic tool to determine the functional and biological significance of genes and gene products in our results. GO describes the attributes of genes in three key domains, molecular function, biological process and cellular component. We calculate p-values of each gene in the biclusters with the hypergeometric probability distribution. In one additive bicluster of six organs, bladder, colon, ileum, stomach, ureter, and uterus, the smallest p-value is 0.0004 corresponding to the GO term 0004479, which is related to methionyl-tRNA formyltransferase activity in molecular function. In one multiplicative bicluster of 11 organs, bladder, colon, heart, ileum, heart, lung, ovary, prostate, stomach, ureter and uterus, the smallest p-value is 0.0003 corresponding to GO term 0030145, which is related to cell differentiation in biological process. We discovered that significant genes in the bicluster patterns are mostly related to molecular function rather than biological process analyzed in [13].

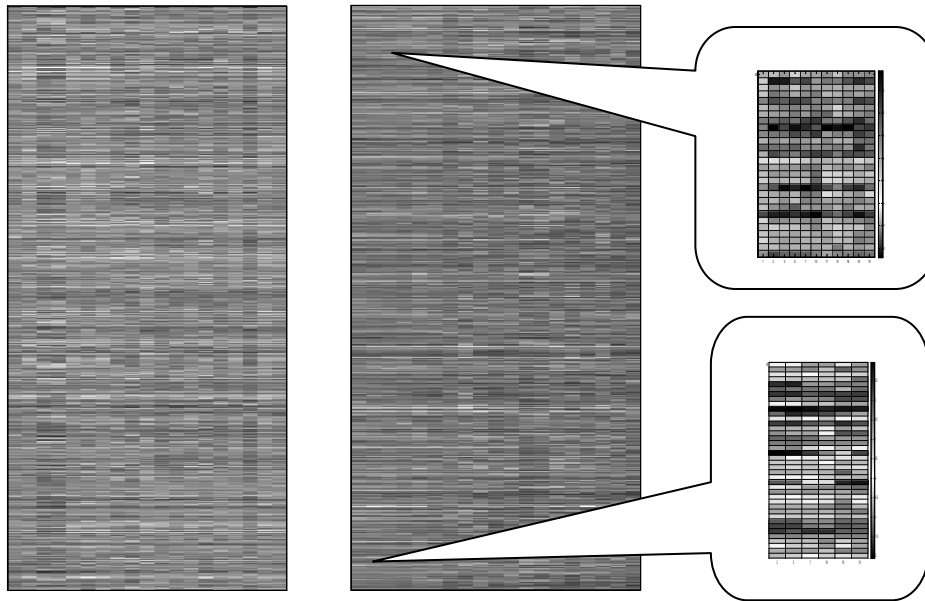


Figure 2. Bicluster analysis of 5298x19 gene expression data matrix of multiple human organs. The original data matrix is left one; Two bicluster patterns after permutation of genes and conditions in right: one multiplicative bicluster 28x11 in the upper-left corner and the other additive 40x6 one in the left bottom part.

References

1. D. H. Ballard and C. M. Brown. Computer Vision. Prentice Hall, Englewood Cliffs, NJ, 1982.
2. W. S. Cleveland. Visualizing data. Murray Hill, N.J.: At & T Bell Laboratories, 1993.
3. Y. Cheng and G. M. Church. Biclustering of expression data. *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB)*, 93-103, 2000.
4. X. Gan, A. W. C. Liew and H. Yan. Biclustering gene expression data based on high dimensional geometric method. *Proc. Int'l Conf. Machine Learning and Cybernetics*, IEEE SMC Society, **6**: 3388-3393, 2005.
5. G. Getz, E. Levine and E. Domany. Coupled two-way clustering analysis of gene microarray data. *Proc. Natural Academy of Sciences USA*, **97**: 12079-12084, 2000.
6. J. A. Hartigan. Direct clustering of a data matrix. *J. American Statistical Association*, **67**(337), 123-129, 1972.
7. J. Illingworth and J. Kittler. A survey of the Hough transform. *Computer Vision, Graphics, and Image Processing*, **44**(1):87-116, 1988.
8. Y. Kluger, R. Basri, J. T. Chang and M. Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Research*, **13**:703-716, 2003.
9. Lazzeroni, L and Owen, A. Plaid models for gene expression data. *Statistica Sinica*, **12**:61-86, 2002.
10. S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Trans. Computational Biology Bioinformatics*, **1**:24-45, 2004.
11. E. Segal, B. Taskar, A. Gasch, N. Friedman and D. Koller. Rich probabilistic models for gene expression. *Bioinformatics*, **17**: S243-S252, 2001.
12. Q. Sheng, Y. Moreau and B. De Moor. Biclustering microarray data by Gibbs sampling. *Bioinformatics*, **19**:Sii196-Sii205, 2003.
13. C. Son, S. Bilke, S. Davis, B. Greer, J. Wei, C. Whiteford, Q. Chen, N. Cenacchi and J. Khan. Database of mRNA gene expression profiles of multiple human organs. *Genome Research*, **15**: 443-450, 2005.
14. R. B. Stoughton. Applications of DNA Microarrays in Biology. *Annu Rev Biochem*, **74**:53-82, 2004.
15. A. Tanay, R. Sharan and R. Shamir. Biclustering Algorithms: A Survey. In *Handbook of Computational Molecular Biology*. Edited by: Aluru S. Chapman & Hall/CRC Computer and Information Science Series, 2005.
16. The Gene Ontology Consortium. Gene ontology tool for the unification of biology. *Nat. Genet.*, **25**:25-29, 2000.
17. S. Wu, A. W. C. Liew, H. Yan, H. Yang. Cluster analysis of gene expression data based on self-splitting and merging. *IEEE Trans. Information Technology Biomedicine*, **8**:5-15, 2004.
18. L. Xu and E. Oja. Randomized Hough Transform (RHT): Basic Mechanisms, Algorithms, and Computational Complexities. *CVGIP: Image Understanding*, **57**:131-154, 1993.