

國立台灣大學公共衛生學院流行病學與預防醫學研究所

碩士論文

Institute of Epidemiology and Preventive Medicine

College of Public Health

National Taiwan University

Master Thesis

利用相關性矩陣降維進行雙分群分析：

以基因表現資料為例

A biclustering method with correlation matrix

for gene expression profiling

陳建瑋

Chian Wei Chen

指導教授：蕭朱杏博士

Advisor: Chuhsing Kate Hsiao, Ph.D.

中華民國 103 年 7 月

July, 2014



致謝



這個碩士論文，首先要非常感謝我的指導教授蕭朱杏老師。蕭老師擁有豐富的研究知識，以身作則帶領我們做研究。從老師身上我學會了不僅是如何做研究，更學習到認真負責任的生命態度。每當我感到困惑或陷入研究瓶頸時，她總是能抽出時間陪我聊天並給予方向。跟老師聊完後，目標明確許多也不再那麼徬徨。蕭老師總是不斷的鼓勵學生，“你絕對會解決了這個問題！”，“你會按時畢業，真的！”很幸運這兩年能在蕭老師的實驗室學習、成長！

要謝謝郭柏秀老師與李美賢老師擔任口試委員，從不同角度來評論這篇論文，兩位老師提的問題與建議都讓論文架構與內容更豐富完整。

另外要感謝我的同學劉智彥，他教了我很多 R 的語言。也要謝謝 R01 的碩班同學們，我們一同學習、研究，也一同玩樂、聊天。也要謝謝蕭老師實驗室的成員很親切、什麼問題都難不倒的彥雯學姐，穿著變化多端也最可愛的家瑋，快要是極端氣候達人的于瑄，直升博班的書如，世亨學長與小麥。在 lab meeting 上用英文互相提問，看到大家抽籤被抽到的尷尬表情還有絞盡腦汁想問題與回答的模樣還是很有趣。也要感謝我的女友珮綺，碩班的這兩年我們互相學習和成長，彼此打氣加油的模樣是最美的記憶。最後，我要感謝我的家人支持我念碩士學位，溫暖而彼此關懷的家庭是能夠心無旁騖念書的穩定依靠。

建瑋

中文摘要



雙分群分析方法近年來在統計學上是相當重要的分析工具，特別是在歸類哪些基因在某些特定實驗下會有相似的基因表現。雙分群分析目標是找出哪些基因在一個特定實驗集合下的基因表現會有相同表現趨勢。先前研究大多是類別分析方法的推廣，集中研究於基因在所有的實驗條件之間的相似性。在本篇論文當中我們提出利用基因表達之間的相關性矩陣以及實驗條件之間的相關性矩陣降維進行雙分群分析，簡稱 BiCor。利用這兩個相關性矩陣，每次的迭代運算都會刪除最不相關的基因或實驗條件。根據預先指定的收斂條件，結果會得到較小的矩形陣列，此矩形陣列裡的基因表現從基因角度以及實驗條件角度看來都有相似的趨勢。我們更進一步定義真實偵測率 (TDR) 與成功被偵測率 (DTR) 用來評估 BiCor 的表現。最後利用模擬試驗與實際資料進行分析，比較 BiCor 和其他現有雙分群分析方法優劣。

關鍵字：雙分群，相關性，基因表現

Abstract



Biclustering has become an important analytical tool in recent statistical practice, particularly when it is of interest to group genes under certain experimental conditions.

The goal of such biclustering analysis is to identify sets of genes sharing similar expression patterns across subsets of samples. Previous developed approaches were mostly extensions of clustering methods and thus focused more on similarity between genes across all experimental conditions. Here we proposed a bicluster algorithm via correlation matrices, called BiCor, between gene expression patterns and between conditions. Each of these two matrices was visited iteratively to remove the most irrelevant genes or conditions. Under a pre-specified convergence criterion, the resulting smaller rectangular contains expression levels that are considered similar at both the gene and the condition level. We further defined the true discovery rate (TDR) and discovered true rate (DTR) to assess the performance of the proposed algorithm. Simulation studies and applications were conducted to evaluate and compare the proposed BiCor with other existing algorithms.

Key words: Bicluster, correlation, gene expression

Table of Contents



致謝.....	i
中文摘要.....	ii
Abstract.....	iii
List of Tables	v
List of Figures	vi
List of Appendices	vii
Introduction.....	1
Method	4
Simulation.....	11
Application.....	17
Discussion.....	21
References.....	24
Appendix.....	42

List of Tables



Table 1 Examples of performance evaluation.....	26
Table 2 Parameter settings for the four bicluster algorithms.	27
Table 3 Performance comparison of the four bicluster algorithms.	28
Table 4 Different simulation settings for BiCor algorithm.	29
Table 5 Performance of BiCor under different data-dependent thresholds.	30
Table 6 Bicluster result of four bicluster algorithms	32
Table 7 Parameter settings of 3 bicluster algorithms in <i>Arabidopsis</i> data.....	34
Table 8 23 experiment conditions of <i>Arabidopsis</i> data.....	34
Table 9 Genes coding for enzymes in the two isoprenoid pathways	35
Table 10 Bicluster result of four bicluster algorithms and two cluster results.....	36
Table 11 Parameter settings of 5 bicluster algorithms in <i>Arabidopsis</i> data	36

List of Figures



Figure 1 An example of a two-component mixture model for correlations.....	37
Figure 2 Low true discovery rate and low discovered true rate.....	37
Figure 3 High true discovery rate and low discovered true rate.....	37
Figure 4 Low true discovery rate and high discovered true rate.....	37
Figure 5 Performance of four algorithms.....	38
Figure 6 Performance of BiCor with fixed thresholds.....	38
Figure 7 Performance of BiCor with fuzzy threshold.....	39
Figure 8 Mixture model of correlation of rows and columns in <i>Arabidopsis</i> data.....	39
Figure 9 Mixture model of correlation of rows and columns in <i>Arabidopsis</i> data.....	40
Figure 10 Correlation of 20 genes and 118 experiments of MVA pathway in <i>Arabidopsis</i> data.....	41
Figure 11 Correlation of 19 genes and 118 experiments of MEP pathway in <i>Arabidopsis</i> data.....	41

List of Appendices

Appendix 1 Code for generating $A_{(50+150)*(50+150)}$ matrix.....	42
Appendix 2 Code for random generation fo the $(a+b)*(c+d)$ matrix	43
Appendix 3 Code for the biclustering methods via correlation matrix (BiCor)	43



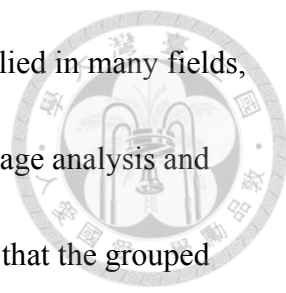
Introduction



In recent years, there have been various efforts to overcome the limitations of standard clustering approaches for the analysis of gene expression data by grouping genes and experimental conditions simultaneously. Such analyses are usually called biclustering and are used to identify sets of genes sharing similar expression patterns across subsets of samples, meaning the genes may work together under these conditions. Biclustering methods can be used not only in gene expression profiling data but also in other biological data.

Biclustering is a method that identifies sets of genes sharing similar expression patterns across subsets of experiment conditions. The difference between traditional clustering and biclustering algorithm is that the clustering method clusters only rows or only columns in a data matrix; while the biclustering method clusters rows and columns simultaneously. With biclustering, genes belonging to different groups of conditions may be identified. In addition, a gene functions under different experiment conditions can be identified if it is grouped in different biclusters.

In contrast, cluster analysis was used to group a set of objects such as subjects who behave similarly in one group than in others. The main purpose of cluster analysis is exploratory data mining. There are many cluster analysis methods such as K-means, hierarchical clustering, Independent component analysis (ICA) and



Principal component analysis (PCA). These methods have been applied in many fields, such as bioinformatics (gene expression data), machine learning, image analysis and pattern recognition. An underlying assumption of cluster analysis is that the grouped objects (subjects or genes) behave similarly across all experiment conditions (measurement), treating all conditions exchangeable. In reality, however, genes tend to co-regulate in some experiment conditions but not in all experiment conditions. In this case, therefore, a biclustering algorithm would serve the purpose better than a clustering algorithm.

The idea of biclustering was first proposed by Hartigan (1972). Currently there are four biclustering algorithms widely used in research, δ -size bicluster, Cheng and Church's algorithm (CC, Cheng et al., 2000), Statistical-Algorithmic Method of Bicluster Analysis (SAMBA, Tanay et al., 2002), Iterative Signature Algorithm (ISA, Ihmels et al., 2002, 2004), and Binary inclusion-maximal biclustering algorithm (Bimax, Prelic' et al., 2006). A special feature of CC is that it was applied under a fixed δ size of bicluster. This fixed δ size denotes the upper limit of the mean squared residual (MSR) of the bicluster. For SAMBA, it uses bipartite graph and binomial distribution to find the potential bicluster. For ISA, its pros is that ISA uses iterative method to see if the output bicluster will be the same when different initial genes are considered as input. The advantage of BiMax is that BiMax can

find the largest number of biclusters because BiMax searches every possible biclusters.

These four methods did not consider correlation between genes or between conditions

but assumed genes independent and experiment conditions independent as well. In

fact, genes may co-express in a condition but not in other conditions, leading to

correlation between conditions for this certain set of genes. Hence, correlation is an

important and intuitive characteristic that should be accounted for in biclustering

algorithms.

In this paper, we calculate first the gene-gene and condition-condition correlation matrices, and then iteratively reduce the size to a bicluster if the criterion is satisfied.

One advantage of the proposed BiCor is that the algorithm does not need to be

normalized because the operation of correlation matrices is not affected by the

original scales. Next, we carried out simulations and compare BiCor with other

biclustering algorithms. Finally we conclude with a discussion and conclusion.

Method



Notations

Let A_{n*m} be the data matrix of gene expressions from n genes $\{G_1, G_2, G_3, \dots, G_n\}$ and m experiment conditions $\{C_1, C_2, C_3, \dots, C_m\}$. Each row vector \mathbf{x}_i , $i=1, \dots, n$ of the matrix A_{n*m} is of dimension $1 \times m$, and can be written as $\mathbf{x}_i = (g_{i1}, g_{i2}, \dots, g_{im})$ where g_{ij} stands for the expression level of the gene G_i under the condition C_j . Similarly, each column vector \mathbf{y}_j , $j=1, \dots, m$ of the matrix A_{n*m} is of dimension $n \times 1$, and can be written as $\mathbf{y}_j = (g_{1j}, g_{2j}, \dots, g_{nj})$ standing for the gene expression levels of genes $\{G_1, G_2, G_3, \dots, G_n\}$ under the same condition C_j . Thus the matrix A_{n*m} is

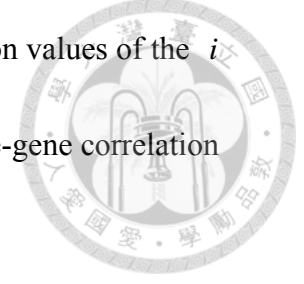
$$A_{n*m} = \begin{pmatrix} g_{11} & \cdots & g_{1m} \\ \vdots & \ddots & \vdots \\ g_{n1} & \cdots & g_{nm} \end{pmatrix}.$$

Our algorithm use Pearson correlation coefficient to measuring similarity between expression patterns of two genes G_i and $G_{i'}$ or between two conditions C_j and $C_{j'}$. For instance, the Pearson correlation coefficient between G_i and $G_{i'}$ is defined as :

$$Corr(G_i, G_{i'}) = Corr(\mathbf{x}_i, \mathbf{x}_{i'}) = \frac{\sum_{l=1}^m (g_{il} - \bar{g}_i)(g_{i'l} - \bar{g}_{i'})}{\sqrt{\sum_{l=1}^m (g_{il} - \bar{g}_i)^2 \sum_{l=1}^m (g_{i'l} - \bar{g}_{i'})^2}}. \quad (1)$$

Here the g_{il} and $g_{i'l}$ are the l -th expression value of the i -th gene and i' -th

gene, and the \bar{g}_i and $\bar{g}_{i'}$ are the mean values over m expression values of the i -th gene and i' -th gene. Since there are n genes in total, the gene-gene correlation matrix becomes an $n \times n$ matrix $G_{n \times n} = (g_{ij})_{n \times n}$.



The correlation between conditions C_j and $C_{j'}$ is defined in a similar way,

$$\text{Corr}(C_j, C_{j'}) = \text{Corr}(\mathbf{y}_j, \mathbf{y}_{j'}) = \frac{\sum_{l=1}^m (g_{lj} - \bar{g}_j)(g_{lj'} - \bar{g}_{j'})}{\sqrt{\sum_{l=1}^m (g_{lj} - \bar{g}_j)^2 \sum_{l=1}^m (g_{lj'} - \bar{g}_{j'})^2}}.$$

And the resulting column-column correlation matrix $C_{m \times m} = (c_{ij})_{m \times m}$ is of dimension $m \times m$.

A bicluster B_k can be defined as a subset of genes I_k possessing a similar behavior over a subset of experiment conditions (measurements) J_k . Thus, a bicluster B_k can be represented as $B_k = (I_k, J_k)$. This bicluster $B_k = (I_k, J_k)$ contains a subset $I_k (I_k \subseteq \{G_1, G_2, G_3, \dots, G_n\})$ of genes and a subset $J_k (J_k \subseteq \{C_1, C_2, C_3, \dots, C_m\})$ of experiment conditions where each gene in I_k is correlated with a correlation value greater than or equal to a pre-specified threshold (δ_{gene} or $\delta_{condition}$), with all other genes in I_k over the measurements in J_k . That is, $|\text{Corr}(\mathbf{x}_i^{(k)}, \mathbf{x}_{i'}^{(k)})| > \delta_{gene}$ if both gene i and gene i' belong to I_k and $|\text{Corr}(\mathbf{y}_j^{(k)}, \mathbf{y}_{j'}^{(k)})| > \delta_{condition}$ if the conditions j and j' are in the same subset. Note that we used the notations $\mathbf{x}_i^{(k)}$ and $\mathbf{y}_j^{(k)}$ to indicate that the correlations are measured over the genes and conditions in the corresponding bicluster B_k only, not over the original n genes and m experiment

conditions.

Algorithm



The proposed algorithm starts with the expression matrix $A_{n \times m}$ with n genes and m experiment conditions. For the purpose of illustration, we first perform our bicluster algorithm on the column-column correlation matrix.

Step 1: Calculate the column-column correlation matrix $C_{m \times m}$ and identify the pair

with the minimum absolute value of correlation. For example, if conditions

C_j and $C_{j'}$ has the minimum absolute value of correlation in the

column-column correlation matrix $C_{m \times m}$. Then delete one of the pair $(C_j, C_{j'})$

whose summation of absolute correlation values over rows, $\sum_{k=1}^n |c_{jk}|$ or

$\sum_{k=1}^n |c_{j'k}|$, is smaller. After removing one condition, the original expression

data matrix would be reduced to the matrix $A_{n \times (m-1)}^{(1)}$.

Step 2: For the remaining n genes and $m-1$ experiment conditions, calculate the

row-row correlation matrix $G_{n \times n}^{(1)}$ and identify the pair of genes

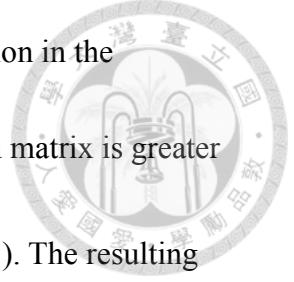
corresponding to the minimum absolute value of correlation. For instance, if

G_i and $G_{i'}$ has the smallest absolute value of correlation in the row-row

correlation $G_{n \times n}^{(1)}$, then compute the two sums of correlations, $\sum_{k=1}^m |r_{ik}|$ and

$\sum_{k=1}^m |r_{i'k}|$, and remove either G_i or $G_{i'}$ with the smaller value. The original

expression matrix is now updated to $A_{(n-1) \times (m-1)}^{(2)}$.



Step 3: Repeat Steps 1 and 2 until every absolute value of correlation in the column-column correlation matrix and row-row correlation matrix is greater than or equal to a pre-specified threshold (δ_{gene} or $\delta_{condition}$). The resulting matrix then leads to a bicluster $B_k = (I_k, J_k)$ where $|Corr(\mathbf{x}_i^{(k)}, \mathbf{x}_{i'}^{(k)})| > \delta_{gene}$ if gene i and gene i' belong to I_k and $|Corr(\mathbf{y}_j^{(k)}, \mathbf{y}_{j'}^{(k)})| > \delta_{condition}$ if conditions j and j' belong to J_k .

If we start the bicluster algorithm from the row-row correlation matrix, then step 2 will be performed as an initial step before step 1. The Steps are as follows:

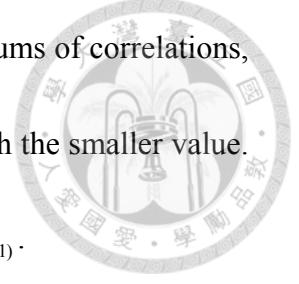
Step 1: Calculate the row-row correlation matrix $G_{n \times n}$ and identify the pair with the minimum absolute value of correlation. For example, if genes G_j and $G_{j'}$ has the minimum absolute value of correlation in the row-row correlation matrix $G_{n \times n}$. Then delete one of the pair $(G_j, G_{j'})$ whose summation of absolute correlation values over rows, $\sum_{k=1}^n |r_{jk}|$ or $\sum_{k=1}^n |r_{j'k}|$, is smaller. After removing one condition, the original expression data matrix would be reduced to the matrix $A_{(n-1) \times m}^{(1)}$.

Step 2: For the remaining $n-1$ genes and m experiment conditions, calculate the column-column correlation $C_{m \times m}^{(1)}$ and identify the pair of conditions corresponding to the minimum absolute value of correlation. For instance, if C_i and $C_{i'}$ has the smallest absolute value of correlation in the

column-column correlation $C_{m \times m}^{(1)}$, then compute the two sums of correlations,

$\sum_{k=1}^n |c_{jk}|$ or $\sum_{k=1}^n |c_{j'k}|$, and remove either C_i and $C_{i'}$ with the smaller value.

The original expression matrix is now updated to $A_{(n-1) \times (m-1)}^{(2)}$.



Step 3: Repeat Steps 1 and 2 until every absolute value of correlation in the

column-column correlation matrix and row-row correlation matrix is greater

than or equal to a pre-specified threshold (δ_{gene} or $\delta_{condition}$). The resulting

matrix then leads to a bicluster $B_k = (I_k, J_k)$ where $|Corr(\mathbf{x}_i^{(k)}, \mathbf{x}_{i'}^{(k)})| > \delta_{gene}$

if gene i and gene i' belong to I_k and $|Corr(\mathbf{y}_j^{(k)}, \mathbf{y}_{j'}^{(k)})| > \delta_{condition}$ if

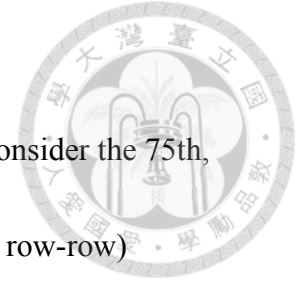
conditions j and j' belong to J_k .

It is worth noting that no matter which direction (row or column) we start with, the resulting DTR differs by only 0.02, and the selected rows and columns in the identified bicluster only differ by 1-2 rows or 1-2 columns.

Choice of Thresholds

We propose three choices for both the threshold δ_{gene} and $\delta_{condition}$. The first one is user-defined. Since the correlation between expression levels in a bicluster may depend on the strains of cells considered in the study and the treatments applied on the cells, expert's opinion on the degree of correlation should be ascertained and to construct the threshold values. For example, one researcher may prefer $\delta_{gene}=0.3$ and $\delta_{condition}=0.5$ in the algorithm; while another may select a more strict standard as δ_{gene}

$=0.5$ and $\delta_{condition} = 0.5$.



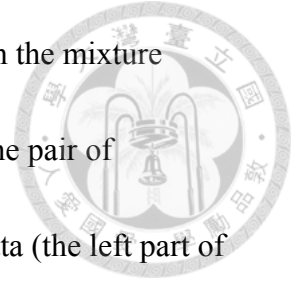
The second choice is data-dependent. For example, one may consider the 75th, 85th or 95th percentile of the correlations in the column-column (or row-row)

correlation matrix as the threshold. Two advantages are associated with this choice.

First, such choice would be practical especially when no expert knowledge is available. Second, the different percentiles may help to investigate the sensitivity of the final biclusters in the threshold values. In the rest of the paper, we adopt this choice and will carry out comparison for different choices.

The third choice is also data-dependent but requires prior statistical inference from a mixture model. Under the assumption that the correlations in the column-column (or row-row) correlation matrix consists of a bicluster and noise, we use the mixtools package in R software to fit a mixture model of two normal components. The threshold is then determined as the value where two normal density functions intersect. Figure 1 is an illustration of a mixture model for correlations using data ($S_{100 \times 100}$) randomly selected from $A_{200 \times 200}$ with seed 1. A two-component normal mixture model was then fitted as the figure showed. We can see that there were two distributions in the mixture model, where the proportion of the red distribution was 84%. The mean of the red distribution was 0.018, while the mean for the green distribution was 0.15. The intersection occurs at correlation=0.18. If there are three

biclusters in data, then there will be three correlation distributions in the mixture model. In this case, we can find the rightmost bicluster first, with one pair of thresholds in the proposed algorithm, and then use the remaining data (the left part of the histogram of correlations) to undergo further biclustering procedure to separate the rest two distributions.



Simulation



Simulation settings

To evaluate the performance of the proposed biclustering algorithm and to compare with other existing methods such as CC, Bimax, and ISA, we performed simulation studies. First we constructed a larger population matrix of expression levels $A_{200 \times 200}$ containing a true bicluster $B_{50 \times 50}$ as well as other noises. Let $B_{50 \times 50}$ denote the expression levels from 50 truly clustered genes and 50 clustered conditions. The expression levels in $B_{50 \times 50}$ were generated in a conditional fashion, where the first random vector in $B_{50 \times 50}$ was from a multivariate normal distribution

$$MVN(\mathbf{0}, \Sigma_{50 \times 50} = \begin{pmatrix} 1 & 0.5 & 0.5 \\ 0.5 & \ddots & 0.5 \\ 0.5 & 0.5 & 1 \end{pmatrix}_{50 \times 50}),$$

and the other random vectors were from another multivariate normal with a conditional mean vector of the first generated random vector. Such conditioning was to ensure that the correlations between genes $Corr(\mathbf{x}_i, \mathbf{x}_j)$ would be no less than 0.3 ($Corr(\mathbf{x}_i, \mathbf{x}_j) \geq 0.3$); while the correlation between conditions would be around 0.5.

The reason for a larger threshold for the correlations between conditions was because, in the real data, the correlation among genes was smaller than the correlation between conditions. For the remaining components in $A_{200 \times 200}$, they were all generated from a standard normal distribution. The large population matrix $A_{200 \times 200}$ now contained 200

genes and 200 conditions. All the generations were carried out with `mytnorm` package in R software.



Next we selected randomly 100 gene samples under 100 conditions from the population matrix $A_{200 \times 200}$ with 30 genes and 30 conditions from the true bicluster $B_{50 \times 50}$ to form a sample matrix $S_{100 \times 100}$. This sample matrix was then analyzed with BiCor and other biclustering algorithms. Such replications were carried out 100 times and the resulting identified biclusters were collected for comparison. Another simulation setting selected 15 genes and 20 conditions from $B_{50 \times 50}$ to evaluate the performance.

Criteria for performance evaluation

To evaluate the performance of bicluster algorithm, Li considered gene ontology weighted enrichment score and protein-protein interaction score (Li et al., 2012), Prelic considered proportion of disconnected gene pairs and average shortest distance in the graph for metabolic pathway map (MPM) for *A. thaliana* and a protein-protein interaction network (PPI) for *S. cerevisiae* (Prelic A. et al., 2006). Here we proposed two criterion for performance evaluation. The first one is *true discovery rate* (TDR). It indicates the proportion of the true bicluster among the identified bicluster,

True discovery rate of genes

$$= \frac{\text{number of true genes in the identified bicluster}}{\text{number of genes in the identified bicluster}}$$



True discovery rate of conditions

$$= \frac{\text{number of true conditions in the identified bicluster}}{\text{number of conditions in the identified bicluster}}$$

The bigger the true discovery rate of genes (or conditions) is, the better the identified bicluster. A value close to 1 implies a large proportion of true genes (or conditions) in the identified bicluster. This measure, however, cannot evaluate if the identified bicluster recovers most of the original true bicluster. Therefore, we propose the second criterion *discovered true rate* (DTR).

Discovered true rate of genes

$$= \frac{\text{number of true genes in the identified bicluster}}{\text{number of genes in simulation setting}}$$

Discovered true rate of columns

$$= \frac{\text{number of true conditions in the identified bicluster}}{\text{number of conditions in simulation setting}}$$

The bigger the discovered true rate of genes (conditions) is, the better the identified bicluster.

Both criteria above evaluate only one direction of the identified bicluster. To assess the two-dimensional matrix, we combine the TDR and DTR as an overall measure of performance:

Overall true discovery rate of bicluster

$$= \sqrt{\text{TDR of genes} * \text{TDR of conditions}}$$

Overall discovered true rate of bicluster

$$= \sqrt{DTR \text{ of genes} * DTR \text{ of conditions}}$$



These measurements are next considered in the simulation studies.

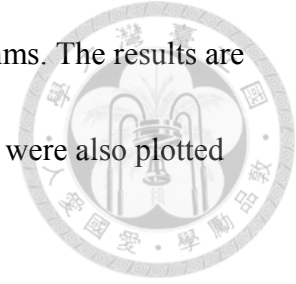
For example, if we considered a $S_{(30+70) \times (30+70)}$ matrix for biclustering, where the first 30 rows and first 30 columns were from the true bicluster $B_{50 \times 50}$ and the rest components were noise. If the BiCor identified a bicluster $B_{(25+2) \times (24+3)}$ containing 25 true genes and 24 true conditions, but 2 false positive genes and 3 false positive conditions. Then the criteria of performance can be calculated, as listed in Table 1. If BiCor identified a much smaller bicluster $B_{(10+0) \times (15+0)}$, then its TDR remained large but the DTR decreased reflecting the fact many genes and conditions have not been recovered. Figures 2-4 are TDR and DTR for three examples. Figure 2 demonstrates the case when TDR and DTR are small, Figure 3 for large TDR and small DTR, and Figure 4 for small TDR and large DTR.

Note that DTR is like the sensitivity and TDR the specificity. An alternative way to compare the performance of the algorithms is the sum of TDR and DTR. One should bear in mind however that the sum of TDR and DTR may disguise the original value of TDR or DTR.

Results

To compare with other existing methods, we consider CC (Cheng et al., 2000), BiMax (Prelic' et al., 2006), and ISA (Ihmels et al., 2002, 2004). Table 2 lists the

parameter values suggested by the authors to be used in the algorithms. The results are shown in Tables 3 and Tables 4. The rates in Tables 3 and Tables 4 were also plotted in Figure 5 and Figure 6.



In the left part of Table 3, under the first simulation setting of 30 true genes and 30 true conditions in $S_{100 \times 100}$ both the marginal true discovery rate for genes or conditions and the overall TDR for BiCor under four different sets of threshold values $\{(0.3, 0.5), (0.2, 0.5), (0.2, 0.4), (0.1, 0.3)\}$ remains close to 1, outperforming CC and ISA. Although BiCor and BiMax had similar performance in TDR, the computation time for BiCor was much less than BiMax. The latter took about 5 minutes for each iteration, while BiCor took only 4 seconds.

The marginal and overall discovered true rates are listed in the right part of Table 3. Such criteria evaluate how many genes or conditions were recovered among the original 30 genes and 30 conditions. It can be observed that the rates under BiCor were between 0.63 and 0.92, depending on the strictness of the threshold values. Less stringent values would lead to better performance. Under these criteria, BiCor performs the second best, next to CC, indicating that CC's method usually identifies a large bicluster, as compared with other algorithms.

In the second simulation we considered a smaller proportion of true bicluster. The second simulation setting considered only 15 true genes and 20 conditions from

the true bicluster $B_{50 \times 50}$, along with other noise components in the matrix $S_{100 \times 100}$.

The number of replications remained at 100. The top half of Table 4 contained the same numbers from the Table 3; while the lower half of Table 4 contained the results under the second simulation settings. It is obvious that, when the proportion of the true bicluster in $S_{100 \times 100}$ is smaller, the performance of BiCor becomes sensitive due to the fact that the identified bicluster becomes small, especially the marginal and overall DTR which used 15 and 20 in the denominator than 50 and 50 in the first setting. The other biclustering algorithms were not compared here because of their poorer performance than that in the first setting.

In addition, we investigated the sensitivity of the performance with respect to data-dependent thresholds. In Table 5, different percentiles were considered in both simulation settings. Although the marginal and overall TDR and DTR in Table 5 were not as good as those in Table 4 under the fixed thresholds, the rates remain satisfactory. The values were expressed in Figure 7.

Considering all the factors affecting the performance, we recommend Bicor over other biclustering algorithms because of its balance between two types of performance evaluation, less computational burden, and because of its robustness to the choice of thresholds.

Application



For real data analysis, we considered the gene expression data from *Arabidopsis thaliana*. *Arabidopsis thaliana* is a small flower with short life cycle of about 6 weeks from germination to mature seed. The genome of *Arabidopsis thaliana* is small, only approximately 135 megabase pairs (Mbp) in 5 chromosomes, and was the first to be sequenced in the year 2000. To understand plant traits, it is popular to consider *Arabidopsis thaliana* as an experimental material. In the following we introduce NASC's data and MVA and MEP pathway study as applications.

NASC's data

We used the real *Arabidopsis* data from NASC's International Affymetrix Service (<http://arabidopsis.info/affy/>). The data can be downloaded at <http://data.iplantcollaborative.org/quickshare/da175c84258a9cf3/Exp340.zip>. (Thilmon et al, 2006) The data contain 734 genes and 23 experiment conditions (each experiment condition were replicated 3 times, leading to a total of 69 experiment conditions). Details are shown in Table 8. As stated in the paper by Thilmon et al. (2006), “*Pseudomonas syringae* pv. tomato DC3000 (Pst) is a virulent pathogen, which causes disease on tomato and *Arabidopsis*. The type III secretion system (TTSS) plays a key role in pathogenesis by translocating virulence effectors from the bacteria into the plant host cell, while the phytotoxin coronatine (COR) contributes to

virulence and disease symptom development.” The goal of this study was to understand if both TTSS and COR are associated with the suppression of host basal defenses.



After performing the biclustering algorithm, we can see in Figure 8 that there are two distributions in correlation of genes and correlation of conditions. Therefore we used $\delta_{gene}=0.4$, $\delta_{condition}=0.6$ and $\delta_{gene}=0.4$, $\delta_{condition}=0.7$ in the BiCor for analysis.

In Table 6, we can see that CC identified a large bicluster because CC's original idea is to find any possible bicluster. BiMax and ISA identified the same condition (DC3000-10e6-24h) and 8 genes. These 8 genes were also identified by BiMax and were among the 43 genes found by ISA. Hence, we have more confidence that these 8 genes work together under the DC3000-10e6-24h condition. BiCor's bicluster is very different from that under BiMax and ISA. This is because BiCor uses correlation of gene expression data to bicluster, while BiMax and ISA use gene expression value to bicluster. As Table 6 shows, BiCor can identify one type of experiment condition alone with its other replications. The DTR of experiment conditions are 73.3% under BiCor(0.4,0.7) and 86.7%. under BiCor(0.4,0.6). Although DTR of experiment conditions under BiCor is smaller than BiMax and ISA, BiCor identified more experiment conditions than BiMax and ISA.

MVA and MEP pathway data



This study investigated two pathways, one was the mevalonate pathway (MVA) and the other was non-mevalonate pathway (MEP). The data were collected from the 118 GeneChip (Affymetrix) microarrays with 39 genes, where 20 of which were assigned to MVA and 19 to MEP, as shown in Figure 8 and Table 9. We use this gene network as a standard to compare the performance of bicluster algorithms. The data can be downloaded at

<http://genomebiology.com/content/supplementary/gb-2004-5-11-r92-s1.txt>. [Wille et al., 2004]

Based on the prior knowledge of the existence of MVA and MEP pathways in *Arabidopsis* data, we fitted a mixture model of correlations in the *Arabidopsis* data with two distributions (Figure 9). We also plot the histograms of correlations for the data under MVA and MEP pathways separately in Figures 10 and 11. We can see that the histograms of correlations of data under MVA and MEP pathways are really similar to the distributions in the mixture model. We use then considered

$\delta_{condition} = 0.8$ to perform BiCor.

Table 10 shows the bicluster results of four bicluster algorithms and two cluster results. BiCor performs the best in terms of TDR for each pathway, the identified biclusters by BiCor are all in their pathways. However, the DTR of pathways was not

large. When threshold of gene became strict, the performance of DTR of pathway got worse. CC performed the best in terms of DTR, almost 74% of genes in two pathways were found. In fact, the correlations under MVA and MEP pathways are similar, therefore it is not easy to have good bicluster results. In this case, it is a tradeoff between TDR and TDR, while one increases when the other decreases.

In cluster algorithms, we use Pearson's correlation distance as a measure of similarity to perform hierarchical clustering (HCL) and K-means. The TDR of pathway and DTR of pathway under K-means was approximately 82%, while that under HCL was only 49%. K-means performs the best here because the current application is interested in clustering, not bicluster.

Discussion



In this paper, we propose a new bicluster algorithm called Biclustering methods via correlation matrix (BiCor). This method uses two correlation matrices to cluster data such as gene expression levels. In the simulation studies, we showed that BiCor can successfully identify the true bicluster with large true discovery rate and discovered true rate. In addition, BiCor outperforms other existing algorithms like CC, BiMax and ISA. However, BiCor can identify one bicluster at one time; while others can find more than one bicluster. To identify more than one biclusters that are non-overlapping, we can use different pairs of δ_{gene} and $\delta_{condition}$, as stated at the end of Section Choice of Thresholds.

BiCor has three advantages. First, BiCor does not require the normalization step either in the levels of genes or conditions. Only the correlation among observations will be investigated. The origin data magnitude as well as the data information will not be lost. Second, the true discovery rate of bicluster in the simulation studies was larger than 90%, implying that BiCor has a low false positive rate. Third, BiCor takes correlation characteristic into consideration, while the CC, BiMax and ISA overlook this property.

Here in this research, we suggested three criteria to determine the threshold used in the algorithm. When no expert opinion is available, we recommend the second

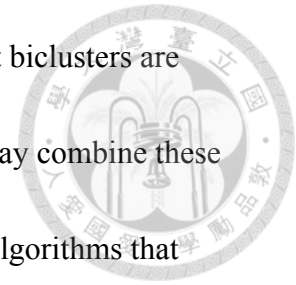
criterion. Thus choice is robust to the correlation pattern in the observed data and to the subject matter under study. Our simulation studies also suggested a satisfactory result when this criterion was considered.



Several issues remain for future studies. First, we plan to make the proposed BiCor algorithm publicly available by providing the code in R so that more people can use it. Second, the validation of true bicluster in real gene expression data analysis may be carried out with the Gene Ontology (GO) database. If considering GO as a standard, then the identified bicluster can be compared with results in GO. We will be working in this direction in the near future. Third, in the simulations, we have generated a square matrix of gene expression levels as the population matrix to start with the replications. This may not be realistic because the number of conditions in laboratory work is usually smaller. Future investigation may focus on such rectangular matrices.

There are two limitations in the BiCor algorithm. First, if the data contain more than the linear correlation, BiCor cannot bicluster well. This is because BiCor uses Pearson Correlation to find bicluster. A remedy can be the exchangeable correlation, first-order autoregressive correlation, unstructured correlation or user-specified correlation matrix, or the kernel methods for nonlinear correlations. In addition, when the data are binary, then other correlation measurements for categorical data should be

considered in BiCor. Second, when the correlations within different biclusters are similar, BiCor may not be able to separate them successfully and may combine these biclusters as a big one. As a modification, one may consider other algorithms that focus on features other than correlations. In conclusion, BiCor has good performance with TDR and moderate DTR; while CC has large DTR but small TDR. On future direction would be to combine CC and BiCor as two steps in bicluster analysis. We could use CC as the first step algorithm to find more correlated genes and conditions, and then use BiCor to check if these genes and conditions are correlated with each other.



References



1. Anindya Bhattacharya and Rajat K. De. 2009. Bi-correlation clustering algorithm for determining a set of co-regulated genes. *Bioinformatics*, 25:21,2795-2801
2. Yizong Cheng and George M. Church. 2000. Biclustering of Expression Data. In Book *Biclustering of Expression Data*. 93–103
3. Ihmels J and Friedlander G et al. 2002. Revealing modular organization in the yeast transcriptional network. *Nature Genetics*, 31, 370–377
4. Ihmels J and Sven Bergmann et al. 2004. Defining transcription modules using large-scale gene expression data. *Bioinformatics*, 20, 1993–2003
5. Li Li and Yang Guo et al. 2012. A comparison and evaluation of five biclustering algorithms by quantifying goodness of biclusters for gene expression data. *BioData Mining* 5:8
6. Sara C. Madeira and Arlindo L. Oliveira. 2004. Biclustering Algorithms for Biological Data Analysis: A Survey. *Ieee transactions on computational biology and bioinformatics*. 1:1
7. Amela Prelić and Stefan Bleuler et al. 2006. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics* 22:1122–1129


- 
8. Udi Ben Porat and Ophir Bleiberg. 2006. Analysis of Biological Networks: Network Modules – Clustering and Biclustering. Lecturer: Roded Sharan
 9. Anja Wille and Philip Zimmermann et al. 2004. Sparse graphical Gaussian modeling of the isoprenoid gene network in *Arabidopsis thaliana*. *Genome Biology* 2004, 5:R92
 10. Amos Tanay and Roded Sharan et al. 2002. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18 Suppl. 1. S136–S144
 11. Thilmony R and Underwood W et al. 2006. Genome-wide transcriptional analysis of the *Arabidopsis thaliana* interaction with the plant pathogen *Pseudomonas syringae* pv. tomato DC3000 and the human pathogen *Escherichia coli* O157:H7. *The Plant Journal* 46, 34–53
 12. <https://www.arabidopsis.org/>

Table 1 Examples of performance evaluation

Performance	BiCor $B_{(25+2)*(24+3)}$	BiCor $B_{(10+0)*(15+0)}$
True discovery rate of row	$25/(25+2)=0.93$	$10/(10+0)=1.00$
True discovery rate of column	$24/(24+3)=0.89$	$15/(15+0)=1.00$
Overall true discovery rate	$\sqrt{0.93 * 0.89}=0.91$	$\sqrt{1.00 * 1.00}=1.00$
Discovered true rate of row	$25/30=0.83$	$10/30=0.33$
Discovered true rate of column	$24/30=0.80$	$15/30=0.50$
Overall discovered true rate	$\sqrt{0.83 * 0.80}=0.81$	$\sqrt{0.33 * 0.50}=0.41$

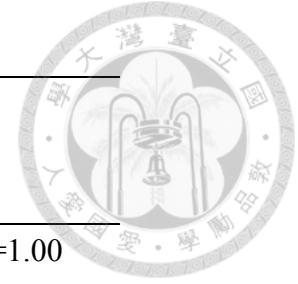




Table 2 Parameter settings for the four bicluster algorithms.

Algorithm	Default Parameter Settings	Changed values
Biclustering method via correlation matrix (BiCor)	$cor_g = 0.3, cor_c = 0.5$	$cor_g = 0.3, cor_c = 0.5$
CC	$\delta = 0.5, \alpha = 0.12$	$\delta = 0.5, \alpha = 0.12$
BiMax	Normalize genes and conditions~N(0,1) Discretize (to binary values) by percentage=10	Normalize genes and conditions~N(0,1) Discretize (to binary values) by percentage=30
ISA	$t_g = 2.0, t_c = 2.0, nr. seeds = 13$	$t_g = 1.0, t_c = 1.0, nr. seeds = 1000$

BiCor: $cor_g: \min Corr(g_i, g_i) \geq \delta$; $cor_c: \min Corr(c_j, c_j) \geq \delta$

CC: δ : the maximum acceptable mean squared residue score. $H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{iJ} - a_{iJ} + a_{IJ})^2 \leq \delta$.

α : a threshold for multiple node deletion.

BiMax: Discretize (to binary values) by percentage=30

ISA: nr. seeds: set seed for random number generator, $t_g = |g_{i,j} - g_{i,j}| \leq \delta$ $t_c = |c_{i,j} - c_{i,j}| \leq \delta$



Table 3 Performance comparison of the four bicluster algorithms.

Algorithm (mean time/each permute)	True discovery rate of rows (mean, se)	True discovery rate of columns (mean, se)	Overall true discovery rate of bicluster (mean, se)	Discovered true rate of rows (mean, se)	Discovered true rate of columns (mean, se)	Overall discovered true rate of bicluster (mean, se)
BiCor (0.3,0.5) (≤ 0.4 secs)	1.00 (<0.01)	1.00 (<0.01)	1.00 (<0.01)	0.72 (0.02)	0.63 (0.02)	0.67 (0.02)
BiCor (0.2,0.5) (≤ 0.4 secs)	1.00 (<0.01)	1.00 (<0.01)	1.00 (<0.01)	0.78 (0.01)	0.64 (0.02)	0.71 (0.02)
BiCor (0.2,0.4) (≤ 0.4 secs)	1.00 (<0.01)	1.00 (<0.01)	1.00 (<0.01)	0.83 (0.02)	0.79 (0.02)	0.81 (0.02)
BiCor (0.1,0.3) (≤ 0.4 secs)	0.99 (<0.01)	1.00 (<0.01)	0.99 (<0.01)	0.87 (0.02)	0.87 (0.02)	0.87 (0.01)
BiCor (90th,90th) (≤ 0.4 secs)	0.99 (<0.01)	1.00 (<0.01)	0.99 (<0.01)	0.85(0.01)	0.92(0.01)	0.88(0.01)
CC (≤ 4 secs)	0.73(0.02)	0.60(0.02)	0.66 (0.02)	0.93(0.05)	0.92(0.02)	0.92 (0.03)
BiMax (≤ 5 mins)	1.00 (<0.01)	0.99(<0.01)	0.99 (0.01)	0.28(0.01)	0.30(0.02)	0.29 (<0.01)
ISA (≤ 5 secs)	1.00 (<0.01)	0.63(<0.01)	0.79 (<0.01)	0.68(0.02)	0.72(0.02)	0.70 (0.01)

se: standard error

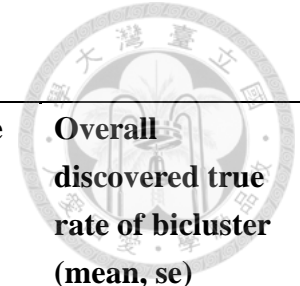


Table 4 Different simulation settings for BiCor algorithm.

Algorithm (mean time/each permute)	True discovery rate of rows (mean, se)	True discovery rate of columns (mean, se)	Overall true discovery rate of bicluster (mean, se)	Discovered true rate of rows (mean, se)	Discovered true rate of columns (mean, se)	Overall discovered true rate of bicluster (mean, se)
$S_{(30+70)*(30+70)}$						
BiCor (0.3,0.5) (≤ 0.4 secs)	1.00 (<0.01)	1.00 (<0.01)	1.00 (<0.01)	0.72 (0.02)	0.63 (0.02)	0.69 (0.02)
BiCor (0.2,0.5) (≤ 0.4 secs)	1.00 (<0.01)	1.00 (<0.01)	1.00 (<0.01)	0.78 (0.01)	0.64 (0.02)	0.71 (0.02)
BiCor (0.2,0.4) (≤ 0.4 secs)	1.00 (<0.01)	1.00 (<0.01)	1.00 (<0.01)	0.83 (0.02)	0.79 (0.02)	0.81 (0.02)
$S_{(15+85)*(20+80)}$						
BiCor (0.3,0.5) (≤ 0.4 secs)	0.91 (0.02)	0.94 (0.02)	0.94 (0.02)	0.55 (0.02)	0.39 (0.02)	0.49 (0.02)
BiCor (0.2,0.5) (≤ 0.4 secs)	0.88 (0.02)	0.93 (0.02)	0.93 (0.02)	0.59 (0.02)	0.40 (0.02)	0.51 (0.02)
BiCor (0.2,0.4) (≤ 0.4 secs)	0.89 (0.02)	0.94 (0.02)	0.93 (0.02)	0.61 (0.02)	0.48 (0.02)	0.56 (0.02)

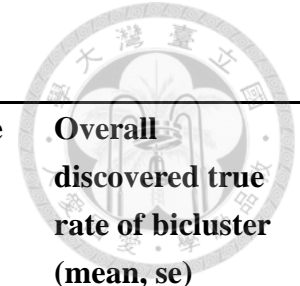


Table 5 Performance of BiCor under different data-dependent thresholds.

Algorithm (mean time/each permute)	True discovery rate of rows (mean, se)	True discovery rate of columns (mean, se)	Overall true discovery rate of bicluster (mean, se)	Discovered true rate of rows (mean, se)	Discovered true rate of columns (mean, se)	Overall discovered true rate of bicluster (mean, se)
$S_{(30+70) \times (30+70)}$						
BiCor (60th, 60th) (≤ 0.3 secs)	0.96(<0.01)	0.97(<0.01)	0.97(<0.01)	0.88(0.01)	0.94(<0.01)	0.91(0.01)
BiCor (70th, 70th) (≤ 0.3 secs)	0.98(<0.01)	0.98(<0.01)	0.98(<0.01)	0.88(0.01)	0.94(<0.01)	0.91(0.01)
BiCor (80th, 80th) (≤ 0.3 secs)	0.99(<0.01)	0.99(<0.01)	0.99(<0.01)	0.87(0.01)	0.93(<0.01)	0.90(0.01)
$S_{(15+85) \times (20+80)}$						
BiCor (75th, 70th) (≤ 0.4 secs)	0.81(0.01)	0.88(0.01)	0.85(0.02)	0.70(0.02)	0.68(0.02)	0.71(0.02)
BiCor (85th, 85th) (≤ 0.4 secs)	0.85(0.01)	0.90(0.01)	0.88(0.02)	0.68(0.02)	0.66(0.02)	0.68(0.02)

80th)
(≤ 0.4 secs)
BiCor (95th,
90th)
(≤ 0.4 secs)

0.89(0.01)

0.92(0.01)

0.91(0.02)

0.66(0.02)

0.63(0.02)





Table 6 Bicluster result of four bicluster algorithms

Bicluster algorithm	Number of genes	Id of experiment conditions	Discovered true rate of experiment conditions
BiCor (0.4,0.7)	32	16 18 (Cor-hrpS-5x10e7-10h) , 24 (hrpAfliC-10e8-7h), 25 26 27 (hrpA-10e8-7h), 34 35 36 (E.coli-0157-H7-10e8-7h), 37 39 (E.coli-TUV86-2-fliC-10e8-7h)	$\frac{11}{5 * 3} = 73.3\%$
BiCor (0.4,0.6)	44	16 18 (Cor-hrpS-5x10e7-10h) , 23 24 (hrpAfliC-10e8-7h), 25 26 27 (hrpA-10e8-7h), 34 35 36 (E.coli-0157-H7-10e8-7h), 37 38 39 (E.coli-TUV86-2-fliC-10e8-7h)	$\frac{13}{5 * 3} = 87.7\%$
CC	155	23 experiment conditions are included	$\frac{39}{13 * 3} = 100.0\%$
BiMax	8	7 8 9 (DC3000-10e6-24h)	$\frac{3}{1 * 3} = 100.0\%$

ISA

43

7 8 9 (DC3000-10e6-24h)

$\frac{3}{3} = 100.0\%$

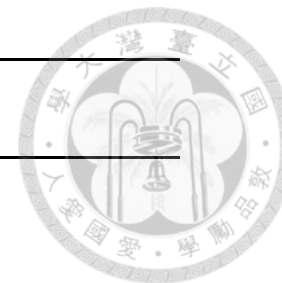


Table 7 Parameter settings of 3 bicluster algorithms in *Arabidopsis* data

Algorithm	Parameter Settings
CC	$\delta = 0.5, \alpha = 0.12$
BiMax	Normalize genes and conditions $\sim N(0,1)$ Discretize to binary values by =2
ISA	$t_g = 2.0, t_c = 2.0, \text{nr. seeds} = 1000$

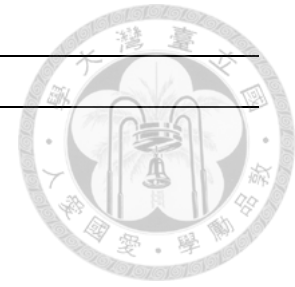


Table 8 23 experiment conditions of *Arabidopsis* data

ID	Pathogen-cfu/ml	Collected time
1-3	Cor-10e6	24h
4-6	Cor-hrpS-10e6	24h
7-9	DC3000-10e6	24h
10-12	Mock-Inoculum	24h
13-15	Cor-5x10e7	10h
16-18	Cor-hrpS-5x10e7	10h
19-21	Mock-Inoculum	10h
22-24	hrpAflC-10e8	7h
25-27	hrpA-10e8	7h
28-30	DC3000-10e8	7h
31-33	Mock-Inoculum	7h
34-36	E.coli-0157-H7-10e8	7h
37-39	E.coli-TUV86-2-fliC-10e8	7h

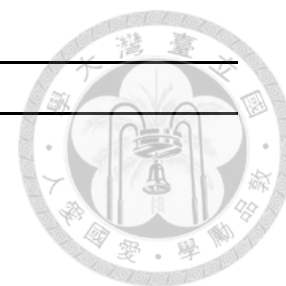
Collected time: collected sample time after giving pathogen-cfu/ml

Arabidopsis data from NASC's International Affymetrix Service

734 genes and 23 experiment conditions (each experiment conditions replicate 3 times, total 69 experiment conditions).

Table 9 Genes coding for enzymes in the two isoprenoid pathways

Name	AGI number	Pathway
AACT1	At5g47720	MVA
AACT2	At5g48230	MVA
CMK	At2g26930	MEP
DPPS1	At2g23410	MVA
DPPS2	At5g58770	MVA
DPPS3	At5g58780	MVA
DXPS1	At3g21500	MEP
DXPS2	At4g15560	MEP
DXPS3	At5g11380	MEP
DXR	At5g62790	MEP
FPSP1	At4g17190	MVA
FPSP2	At5g47770	MVA
GGPPS1	At1g49530	MVA
GGPPS2	At2g18620	MEP
GGPPS3	At2g18640	MVA
GGPPS4	At2g23800	MVA
GGPPS5	At3g14510	MVA
GGPPS6	At3g14530	MEP
GGPPS8	At3g20160	MVA
GGPPS9	At3g29430	MVA
GGPPS10	At3g32040	MEP
GGPPS11	At4g36810	MEP
GGPPS12	At4g38460	MEP
GPPS	At2g34630	MEP
HDR	At4g34350	MEP
HDS	At5g60600	MEP
HMGR1	At1g76490	MVA
HMGR2	At2g17370	MVA
HMGS	At4g11820	MVA
IPPI1	At3g02780	MEP
IPPI2	At5g16440	MVA
MCT	At2g02500	MEP
MECPS	At1g63970	MEP
MK	At5g27450	MVA
MPDC1	At2g38700	MVA
MPDC2	At3g54250	MVA



PPDS1	At1g17050	MEP
PPDS2	At1g78510	MEP
UPPS1	At2g17570	MVA

MVA: Mevalonate pathway; MEP: Non-mevalonate pathway



Table 10 Bicluster result of four bicluster algorithms and two cluster results

Bicluster algorithm	True discovery rate of pathway	Discovered true rate of pathway
BiCor (0.1,0.8)	1	0.45
BiCor (0.2,0.8)	1	0.45
BiCor (0.3,0.8)	1	0.35
CC	0.63	0.74
BiMax	0.63	0.25
ISA	0.92	0.55
Cluster algorithm		
HCL	0.49	0.49
K means	0.83	0.82

Table 11 Parameter settings of 5 bicluster algorithms in *Arabidopsis* data

Algorithm	Parameter Settings
CC	$\delta = 0.5, \alpha = 1.2$
BiMax	Normalize genes and conditions $\sim N(0,1)$ Discretize to binary values by $=1$
ISA	$t_g = 1.0, t_c = 1.0, \text{nr. seeds} = 10000$
HCL	Number of cluster=2, single linkage, Pearson's correlation distance
K means	Number of cluster=2, Pearson's correlation distance

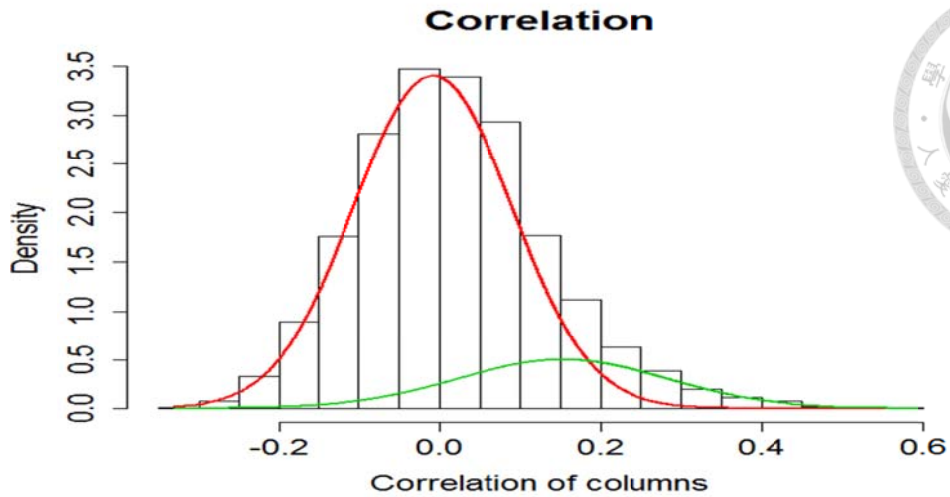


Figure 1 An example of a two-component mixture model for correlations.

Performance and comparison:

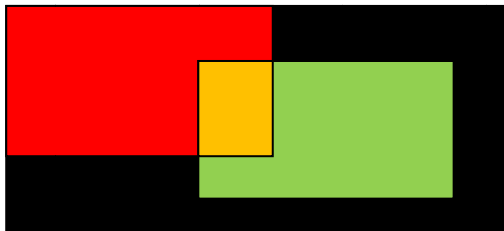


Figure 2 Low true discovery rate and low discovered true rate.

Red: Setting true bicluster; Green: Identified bicluster; Orange: Overlapping region

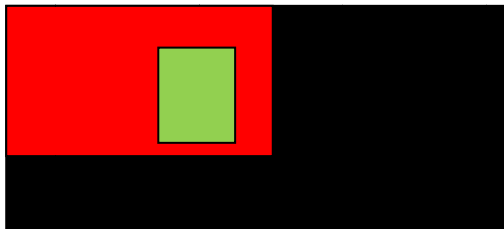


Figure 3 High true discovery rate and low discovered true rate



Figure 4 Low true discovery rate and high discovered true rate

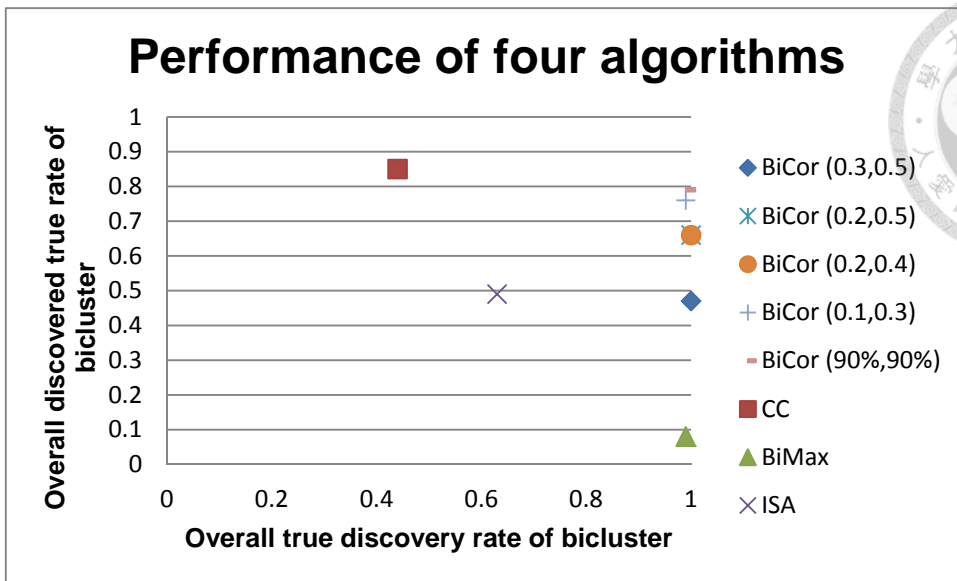


Figure 5 Performance of four algorithms

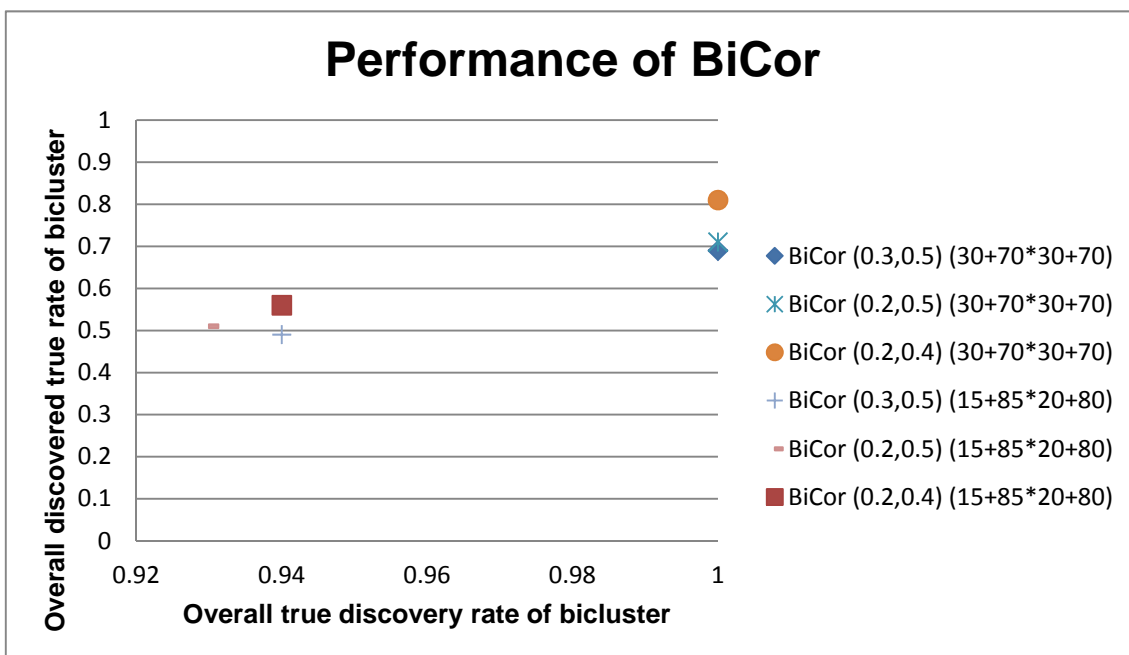


Figure 6 Performance of BiCor with fixed thresholds

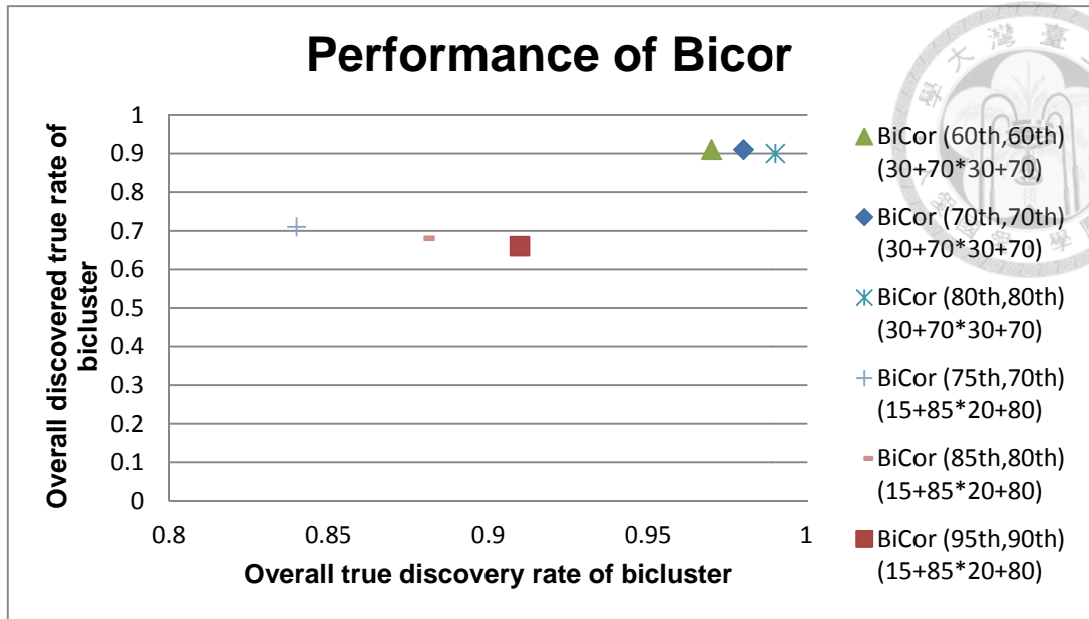


Figure 7 Performance of BiCor with fuzzy threshold

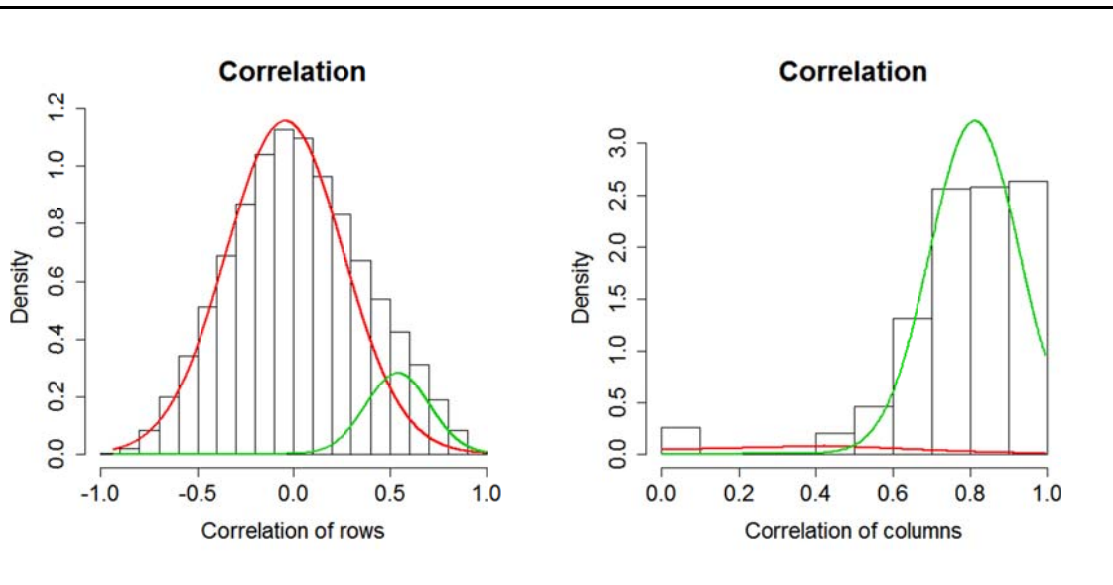


Figure 8 Mixture model of correlation of rows and columns in *Arabidopsis* data

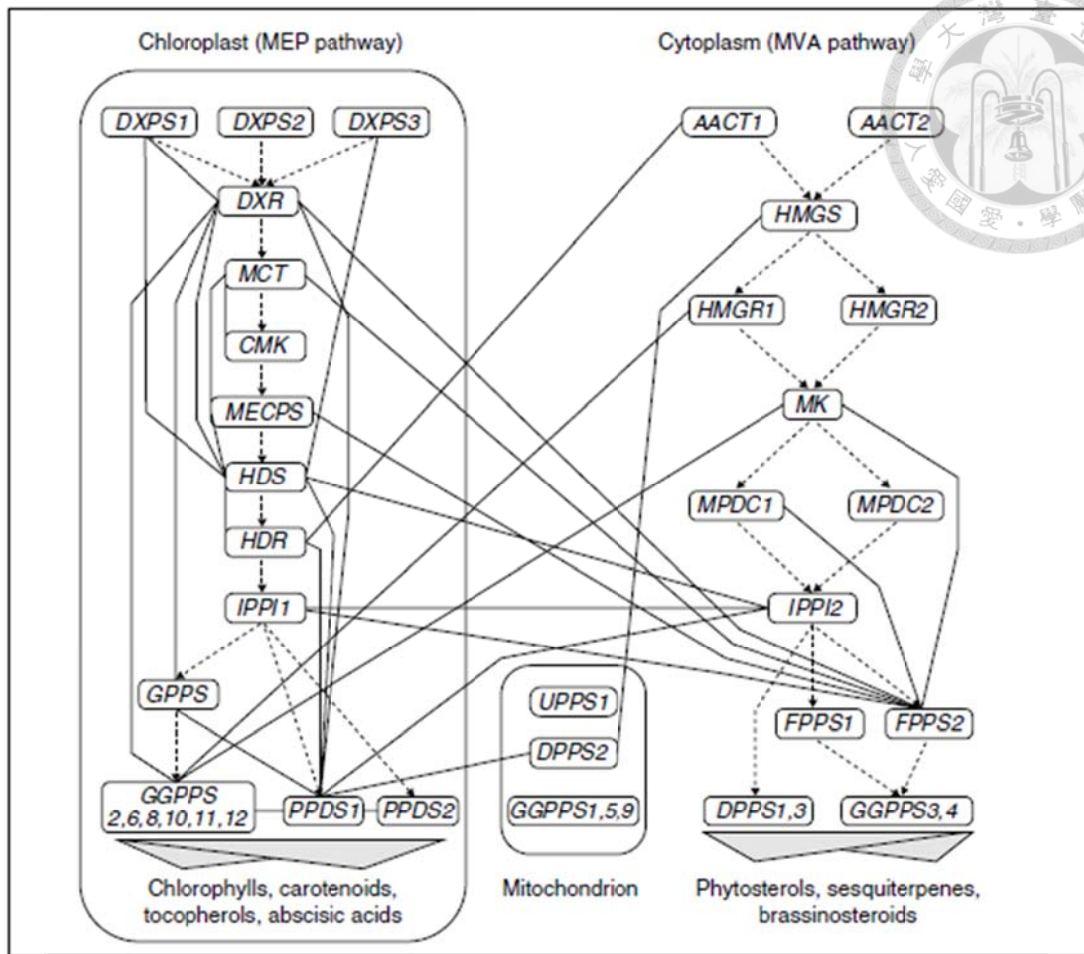
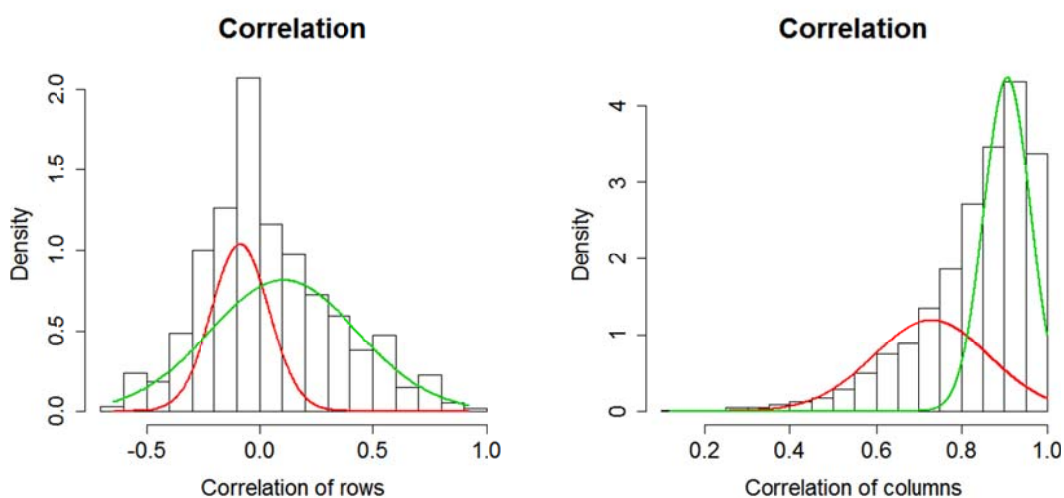


Figure 9 Chloroplast (MEP pathway) and cytoplasm (MVA pathway). Taken from Wille et al. (2004) [5:R92].

Figure 9 Mixture model of correlation of rows and columns in *Arabidopsis* data



Red: MEP pathway; Green: MVA pathway

Figure 10 Correlation of 20 genes and 118 experiments of MVA pathway in *Arabidopsis* data

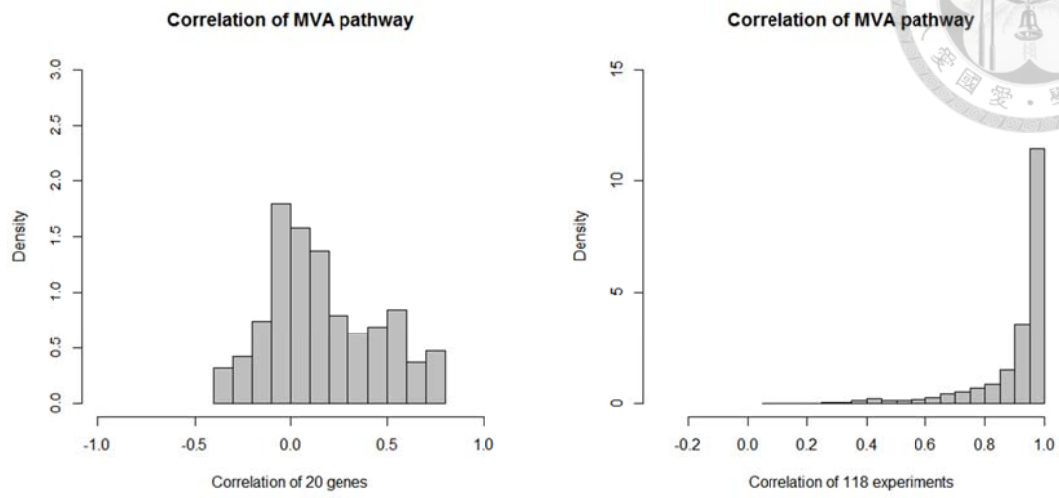
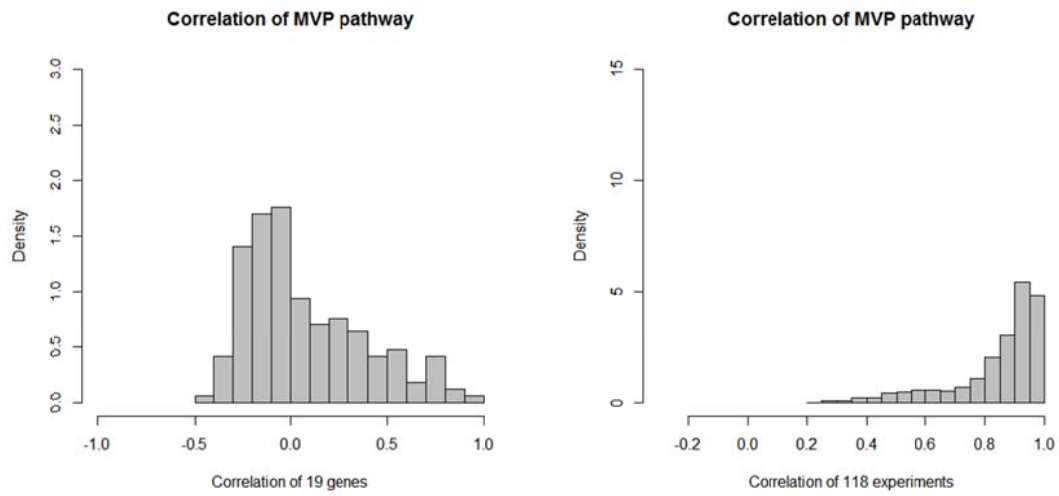


Figure 11 Correlation of 19 genes and 118 experiments of MEP pathway in *Arabidopsis* data



Appendix



Code in R

Appendix 1 Generate $A_{(50+150)*(50+150)}$ process

```
mydim=50 #定義要生出的有相關的矩陣的維度，其實不一定要方正矩陣
temp=matrix(rep(0.5),mydim,mydim) #先定義變異數矩陣
diag(temp)=rep(1,mydim) #定義變異數矩陣對角線為1
mymatrix=matrix(ncol=mydim,nrow=mydim) #定義要被cluster的矩陣
mymatrix[1,]=rmvnorm(1, mean=rep(0,mydim), temp) #先生出第一個row
mymu=1*mymatrix[1,] # new mean vector for conditional pdf 後面直行的mean
跟第一列有關
temp=matrix( rep(0.5) ,mydim,mydim) # new var-var for conditional pdf
diag(temp)=rep(0.75,mydim)
for (i in 2:mydim){ mymatrix[i,]=rmvnorm(1, mean=mymu, temp) } # 生剩下的
columns,現在有50*50矩陣

y1<-rnorm(n=50*150, m=0, sd=1) #生不相關的 50*150 個資料
x1<-array(y1, dim=c(50,150)) #變成 50*150 矩陣
x<-cbind(mymatrix,x1) #得到 50* ( 50+150 ) 矩陣

y2<-rnorm(n=150*200, m=0, sd=1) #生不相關的 150*200 個資料
x2<-array(y2, dim=c(150,200)) #變成 150*200 矩陣
xfinal<-rbind(x,x2) #得到 ( 50+150 ) * ( 50+150 ) 矩陣
```

Appendix 2 Random generate $(a+b)*(c+d)$ matrix

```
random_matrix<-function(a,b,c,d){
  true_sample <- data2[sample(1:50, a,replace=FALSE),]
  sample <- data2[sample(51:200, b,replace=FALSE),]
  sample <- rbind(true_sample,sample)
  sample <- t(sample)

  true_sample <- sample[sample(1:50, c,replace=FALSE),]
  sample1 <- sample[sample(51:200, d,replace=FALSE),]
  sample_final <- rbind(true_sample,sample1)
  sample_final <- t(sample_final)
}
```



Appendix 3 Biclustering methods via correlation matrix (BiCor)

```
Bicluster_cor_algorithm<-function(delta_gene,delta_condition,condition_threshold,ge
ne_threshold,sample ){
  X_temp <- sample
  aa <- TRUE
  i <- 1
  while(aa){
    cat("i=",i,'\n')
    cor_col <- abs(cor(X_temp))
    whichmin <- which(cor_col == min(cor_col), arr.ind = TRUE)
    if(sum(whichmin)==0){break("stop,no bicluster")}
    del_col <-
which.min(c(sum(cor_col[whichmin[1,1],]),sum(cor_col[whichmin[1,2],])))
    if(any(cor_col<delta_condition )){X_temp <-
X_temp[,-whichmin[del_col,1]]}
    if(dim(X_temp)[2]<3){
      OUT <- NA
      show(X_temp);break("stop")
    }

    y_temp <- t(X_temp)
    cor_row <- abs(cor(y_temp))
    whichmin <- which(cor_row == min(cor_row), arr.ind = TRUE)
    del_row <-
hich.min(c(sum(cor_row[whichmin[1,1],]),sum(cor_row[whichmin[1,2],])))
  }
}
```

```

if(any(cor_row<delta_gene )){y_temp <- y_temp[,-whichmin[del_row,1]]}
if(dim(y_temp)[2]<3){
OUT <- NA
show(y_temp);break("stop")
}
X_temp <- t(y_temp)

aa <- (any(abs(cor(y_temp))<delta_gene) |
any(abs(cor(X_temp))<delta_condition ))
if(aa==TRUE){
delta_condition <- min(delta_condition + .01,condition_threshold)
delta_gene <- min(delta_gene + .01,gene_threshold)
i <- i+1
if(i > 10000) {stop("Too many iter!")}
}else if(aa==FALSE){OUT <- X_temp}
}
list(OUT,cor(t(t(OUT))),cor(t(OUT)),i)
}

```

