Multi-Metric and Multi-Substructure Biclustering Analysis for Gene Expression Data

S.Y. Kung
Princeton University
kung@princeton.edu

Man-Wai Mak
The Hong Kong Polytechnic University
enmwmak@polyu.edu.hk

Ilias Tagkopoulos
Princeton University
iliast@princeton.edu

Abstract

A good number of biclustering algorithms have been proposed for grouping gene expression data. Many of them have adopted matrix norms to define the similarity score of a bicluster. We shall show that almost all matrix metrics can be converted into vector norms while preserving the rank equivalence. Vector norms provide a much more efficient vehicle for biclustering analysis and computation. The advantages are two folds: ease of analysis and saving of computation.

Most existing biclustering algorithms have also implicitly assumed the use of univariate (i.e., single metric) evaluation for identifying biclusters. Such an approach however overlooks the fundamental principle that genes (even though they may belong to the same gene group) (1) may be subdivided into different substructures; and (2) they may be co-expressed via a diversity of coherence models (a gene may participate in multiple pathways that may or may not be co-active under all conditions). The former leads to the adoption of a multi-substurcture analysis, while the latter to the multivariate analysis.

This paper will show that the proposed multivariate and multi-subscluster analysis is very effective in identifying and classifying biologically relevant groups in genes and conditions. For example, it has successfully yielded highly discriminant and accurate classification based on known ribosomal gene groups.

1 Introduction

Microarrays have been used to classify clinical samples, to investigate the mechanism of drug action, to examine the effects of drugs on gene expression in yeasts, and to identify and validate novel therapeutics for cancer patients [4, 2, 7]. The gene expression profile (mRNA), one of the molecular signatures (DNA, mRNA, and protein), is a snapshot of the malignant and proliferative mechanism behind cancers. Microarrays produce mass measurements of gene expression, but the tools to analyze the data are not well developed [5]. Because the number of dimensions in a microarray data set could reach from thousands to tens of thousands, the development of these analytical tools is crucial.

A gene expression data is an $M \times N$ matrix of real numbers: $A = [a_{ij}]$, where M is the number of genes and N the number of conditions. Each entry a_{ij} represents the logarithm of the relative abundance of the mRNA of the i^{th} gene under the j^{th} condition.[1]

The gene expression profile of each condition (sample) is described as an M-dimensional vector in which each element represents the expression level of one gene. The presence of well-separated sample groups implies that the representations of samples within the same group are close to each other in this gene expression space but distant from those of other samples. Thus, the representations of phenotype-related samples or condition form clusters. Similarly, the profile of each gene is described as an N-dimensional vector in which each element represents the expression level of one condition. Just like before, the genotypical related genes will form clusters.

1.1 Clustering and Biclustering

Cluster discovery detects previously unrecognized tumor subtypes [5]. Gene selection identifies the most relevant gene subset involving in the biological process that generates the patterns. Phenotype prediction assigns unknown tumor samples to known tumor classes [5].

It is biologically more meaningful to cluster both genes and samples in gene expression data. This leads to a notion of biclustering, first introduced by Hartigan (1972) [6] to describe simultaneous grouping of both row and column subsets in a data matrix. It involves grouping a subset of genes and a subset of conditions with a high similarity score. Biclustering was specialized for genomic grouping by a number of researchers. For examples, see [10, 11, 9, 3].

[1] Note that before data analysis can be performed, it is often necessary to engage a preprocessing step to properly fill in unknown entries in the matrix.

To this end, the similarity must reflect a measure of the coherence of the genes and conditions in the bicluster. The key properties that separate biclustering from clustering are (1) simultaneous clustering of both genes and conditions and (2) permitting to form overlapped grouping. The latter is due to the fact that genes with multiple functions may be simultaneously associated with more than one group. (Such overlapping allows a gene or condition to be simultaneously associated with multiple families.) Nevertheless, there is still intimate relationship between clustering and biclustering and their interplay is in general inevitable.

1.2 Biological Coherence Models

Two popular coherence models regulating the relative abundance of mRNA are additive and multiplicative coherence models. Their corresponding preprocessing processes are and normalization and standardization respectively. It has been long recognized that normalizing and standardizing either or both the rows (genes) and columns (conditions) could improve significantly the biclustering performance.

1. **Additive coherence model**

   A scaling relation between mRNA$_a$ and mRNA$_b$ is expressed as mRNA$_b = k($mRNA$_a)$, where k is a scaling factor. The logarithm transformation

   $$a = \log(\text{mRNA}_a) \qquad \text{and} \qquad b = \log(\text{mRNA}_b)$$

   allows conversion of multiplicative changes of the relative abundance into additive increments [3]: $b = k' + a$ where $k' \equiv \log(k)$. A "normalization" preprocessing step is often adopted to alleviate the uncertainty caused by the additive increments. Computationally, "normalization" is a process which subtracts the mean from each row (or column).

2. **Multiplicative coherence model**

   An exponential relation between mRNA$_a$ and mRNA$_b$ is expressed as mRNA$_b = ($mRNA$_a)^c$. Now the logarithm converts the exponential changes of the relative abundance into multiplicative factors, leading to a "multiplicative model" governing dependence between a and b: $b = c \times a$. A "standardization" preprocessing step can be adopted to counter the uncertainty incurred by the multiplicative increments. Computationally, "standardization" is a process which divides each row (or column) by its standard deviation.

1.3 Organization of the Paper

Section 2 reviews the plausible proximity metrics of biclusters in terms of the prevailing matrix norms. It then discusses various metrics created by a variety of combinations of preprocessing models for genes or conditions. Section 3 presents the computationally more appealing vector norms. It further establishes the rank-equivalence property between the matrix norms and their corresponding vector norms. Section 4 presents biclustering results based on various univariate analysis. Section 5 proposes a fusion scheme combining various metrics as there will be a need of multi-metric evaluation. The multivariate analysis offers a great potential for achieving high performance, which can be confirmed by SVM neural classification. This section shows successful biclustering results in terms of sensitivity-precision-specificity. Section 6 summarizes the novelty of the work and offers some research directions.

2 Biclustering Metrics: Matrix Norms

The first question to address is how to define a bicluster of expression data. The answer hinges upon a notion of proximity measurement which is often used to associate a new member (either a gene or a condition) into a group. The intra-family inhomogeneity of a bicluster is measured by its error residue, which may take various forms of definitions.

1. **Matrix norms:** Traditionally, such a function is represented by a matrix norm of a submatrix of A. The basis for biclustering is often a similarity function of the rows and columns in the expression matrix. The most commonly used matrix norm is the Frobenius norm, denoted by $||A||_F$, which is defined as the square-root of the sum of squares of all elements in the matrix. Unfortunately, the matrix norms are relatively complex as a computational tool for biclustering analysis.

2. **Vector norms:** In contrast, vector norms have very clear and simple physical meaning. The most common measure for the distance between two vectors, say $\mathbf{a}$ and $\mathbf{b}$, is the well-known Euclidean distance denoted by $||\mathbf{b} - \mathbf{a}||$.

2.1 Matrix-Norm Metrics

The notion of biclusters in data matrices was first introduced by Hartigan [6]. The proposed constant-value matrix norm has a very broad application spectrum. After incorporation of some proper preprocessing models, the same matrix norm can also be adopted to measure the similarity between genes or conditions.

1. **Constant-value matrix norm**

   A special case for a perfect bicluster is one with constant value, denoted by c, in every matrix entry. If there is noise or perturbation, then it will cause a deviation represented by a constant-value residue norm [6]:

$$||A||_{constant-value} \equiv \min_{c} ||A - cE||_F$$

where E denotes an all-one matrix, i.e., $E \equiv [\,1\,1 \cdots\,1]^T \times [\,1\,1 \cdots\,1]$ and $||\cdot||_F$ denotes the Frobenius norm.

For gene expression analysis, it is not only natural but also appealing to incorporate biologically relevant coherence models. While conceptually simple, the constant-value based biclustering has a disadvantage that it ignores the biologically justifiable coherence models. This will definitely impose limitation on its classification capability. To overcome this weakness, preprocessing presents a very effective solution.

2. **Additive coherent matrix norm**

   Let us consider first the *normalization-type preprocessing* designed for additive coherent models. If preprocessing is applied to only the rows (or only the columns), then the mathematical operations are as follows:

$$A_{row-normalized} = A - \vec{\alpha}[\,1\,1 \cdots\,1] \quad (1)$$

$$A_{column-normalized} = A - [\,1\,1 \cdots\,1]^T \vec{\beta}^T \quad (2)$$

   where the elements of $\vec{\alpha}$ and $\vec{\beta}$ reflect the amount of adjustment in rows and columns, respectively. However, if both rows and columns are normalized, then we have

$$A_{both-normalized} = A - \vec{\alpha}[\,1\,1 \cdots\,1] - [\,1\,1 \cdots\,1]^T \vec{\beta}^T. \quad (3)$$

   Based on Eq. 3, if optimal normalization is applied, this effectively leads to Cheng and Church's residue given below:

$$||A||_{normal} \equiv \min_{\vec{\alpha},\vec{\beta}} ||A - \vec{\alpha}[\,1 \cdots\,1] - [\,1 \cdots\,1]^T \vec{\beta}^T||_F. \quad (4)$$

3. **Multiplicative coherent matrix norm**

   Other coherence models also have similar matrix-norm formulation. For example, to cope with a row (or column) multiplicative coherent model, we should adopt a *standardization-type preprocessing*.

2.2 Classification of Bilcustering Family

So far, the prevailing assumption is that genes (rows) and conditions (columns) must share the same coherence model. Such a symmetry assumption leads to the conclusion that the rows and columns must receive the same kind of preprocessing. However, such a symmetry property is not necessarily most appealing nor is it truly biologically justifiable.

In order to provide a more comprehensive platform for all plausible coherence models, it is important that we explore various combination of (row and column) preprocessing. This leads to two types of preprocessing models: symmetrical and asymmetrical models, as depicted in Table 1.

1. **Symmetrical preprocessing models**

   The symmetrical preprocessing models are found along the main diagonal boxes in Table 1, i.e., Boxes (1,1), (2,2) and (3,3). For example, for the center Box (2,2) in the table, "normalization" preprocessing is applied to both rows and columns. Therefore, it leads to the Cheng and Church model [3]. Moreover, in the right-lower box, the "normalization and standardization" preprocessing is applied to both rows and columns, leading to a Tavazoie-type test [11].[2]

2. **Asymmetrical preprocessing models**

   All the boxes, except those along the main diagonal, are the so-called asymmetrical preprocessing models. For example, the first sub-diagonal Box (2,1) is simply based on the traditional K-means clustering on rows. According to the simulation study in Section 4.2, c.f. Figure 4, the performance of some asymmetrical coherence models appear to be very promising. In fact, most of them outperform Box (2,2), i.e., Cheng and Church criteria.

To illustrate the operations of different preprocessing options, a numerical examples is provided in Table 2.

3 Biclustering Metrics: Vector Norms

3.1 Vector-Based Metrics

Assume that we are given two vectors **a** and **b**, representing two different genes, and each entry in the vector stands for one particular condition. Before we address the similarity of the two vectors, it is important to take into account the underlying biological coherence models. Fortunately, the comprehensive list of preprocessing options in Table 1 (e.g., normalization and/or standardization for genes and/or conditions) can again be directly applied to adopted to cope with the additive and/or multiplicative coherence models. After preprocessing, the similarity of the vectors can be measured by the traditional Euclidean distance.

1. **Distance for additive coherence models**

   The distance coping with additive coherence is denoted as $(\mathbf{a}, \mathbf{b})_{normal}$. Normalization is effective in

[2] For "normalization" preprocessing, the order of whether row before column or vice versa is immaterial. For "standardization" preprocessing, such order does make some difference. Throughout this paper, we assume that row-wise preprocessing precedes column-wise preprocessing.

Table 1. **List of various possibilities in combining the row (gene) and column (condition) preprocessing models. If the same preprocessing is applied to both rows and columns, it is referred to as a symmetrical preprocessing model. Otherwise, it is categorized into the asymmetrical models. The entries in the table indicate the equivalent type of clustering (far from being exclusive). For example, if preprocessing is applied to rows or columns (but not both), then the resulting clustering is equivalent to K-means. For Box (2,2), if optimal normalization (c.f. Eq. 4) is applied to both rows and columns, this effectively leads to Cheng and Church's clustering. The similar argument carries through to Boxes (2,3), (3,2), and (3,3). See Section 5.1.**

Preprocessing Models	No column-preprocessing	Normalization	Normalization&Standardization
No row-preprocessing	Constant-value	K-means	K-means
Normalization	K-means	C&C-type	Tavazoie-type
Normalization & Standardization	K-means	Tavazoie-type	Tavazoie-type

Table 2. **A numerical example to further elaborate the operations involved in preprocessing models listed in Table 1. The residual matrices after the completion of the corresponding preprocessing processes are listed in the table. For example, Box (2,1) is a result from Eq. (1) while Box (2,2) is from Eq. (3). Given the residual matrices, the final similarity measure can readily be derived either as the Frobenius norm of the residue matrices or via the vector norms in Section 3.**

Consider a gene expression matrix $A = \begin{bmatrix} 10 & 20 & 30 \\ 11 & 22 & 32 \\ 20 & 42 & 61 \end{bmatrix}$, with $\vec{\alpha} = \begin{bmatrix} 20 \\ 21.7 \\ 41 \end{bmatrix}$, $\vec{\beta} = \begin{bmatrix} 13.7 \\ 28 \\ 41 \end{bmatrix}$, and $\mu \approx 27.5$.

Preprocessing Models	No-preprocessing	Normalization	Normalization&Standardization
No-preprocessing	$\begin{bmatrix} -17.5 & -7.5 & 2.5 \\ -16.5 & -5.5 & 4.5 \\ 2.5 & 26.5 & 33.5 \end{bmatrix}$	$\begin{bmatrix} -3.7 & -8 & -11 \\ -2.7 & -6 & -9 \\ 6.4 & 14 & 20 \end{bmatrix}$	$\begin{bmatrix} -0.8 & -0.8 & -0.8 \\ -0.6 & -0.6 & -0.6 \\ 1.4 & 1.4 & 1.4 \end{bmatrix}$
Normalization	$\begin{bmatrix} -10 & 0 & 10 \\ -10.6 & 0.3 & 10.3 \\ -21 & 1 & 20 \end{bmatrix}$	$\begin{bmatrix} 4 & -.5 & -3.5 \\ 3 & -.1 & -3.1 \\ -7 & 0.5 & 6.5 \end{bmatrix}$	$\begin{bmatrix} 0.8 & -1.0 & -0.7 \\ 0.6 & -0.3 & -0.7 \\ -1.4 & 1.4 & 1.4 \end{bmatrix}$
Normalization&Standardization	$\begin{bmatrix} -1.2 & 0.0 & 1.2 \\ -1.2 & 0.0 & 1.2 \\ -1.25 & 0.05 & 1.2 \end{bmatrix}$	$\begin{bmatrix} .02 & -.03 & .02 \\ 0.0 & 0.0 & 0.0 \\ .01 & .03 & -.01 \end{bmatrix}$	$\begin{bmatrix} 1.3 & -1.3 & 1.3 \\ -0.3 & 0.4 & -0.2 \\ -1.0 & 1.0 & -1.1 \end{bmatrix}$

accounting for the additive coherence model. The purpose is to adjust gene levels relative to their average behavior.

Under the additive coherence model, the distance is adjusted by a minimizing parameter c such that

$$\min_{c} ||\mathbf{b} - \mathbf{a} - c[\,1\,1 \cdots\ 1]^T||.$$

This leads to the following minimum distance:

$$(\mathbf{a}, \mathbf{b})_{normal} = ||\bar{\mathbf{b}} - \bar{\mathbf{a}}||$$

where

$$\bar{\mathbf{a}} \leftarrow \mathbf{a} - \mu_{\mathbf{a}}[\,1\,1 \cdots\ 1]^T, \quad \bar{\mathbf{b}} \leftarrow \mathbf{b} - \mu_{\mathbf{b}}[\,1\,1 \cdots\ 1]^T \quad (5)$$

where $\mu_{\mathbf{a}}$ and $\mu_{\mathbf{b}}$ stand for the means of $\{a_i\}$ and $\{b_i\}$ respectively. In Section 3.2, it will be established that such a vector norm is rank-wise equivalent to the Cheng and Church's matrix-norm metric, cf. Eq. 4, i.e., the norm for the additive coherence model [3].

2. **Distance for additive and multiplicative coherence models**

Preprocessing with both normalization and standardization serves the purpose of adjusting gene levels relative to their average behavior and at the same time remove systematic biases in expression ratios. Mathematically,

$$\mathbf{a}^* \leftarrow \bar{\mathbf{a}}/\sigma_{\mathbf{a}}, \quad \mathbf{b}^* \leftarrow \bar{\mathbf{b}}/\sigma_{\mathbf{b}} \quad (6)$$

where $\sigma_{\mathbf{a}}$ and $\sigma_{\mathbf{b}}$ stand for the standard deviation of

$\{a_i\}$ and $\{b_i\}$ respectively. This leads to

$$(\mathbf{a}, \mathbf{b})_{n\&s} = ||\mathbf{b}^* - \mathbf{a}^*||.$$

In Section 3.2, it will be shown that such a norm exists and, moreover, it is rank-wise equivalent to the matrix norm adopted in Tavazoie's test (with row normalization/standardization) [11].

3.2 Relationship of Matrix and Vector Norms

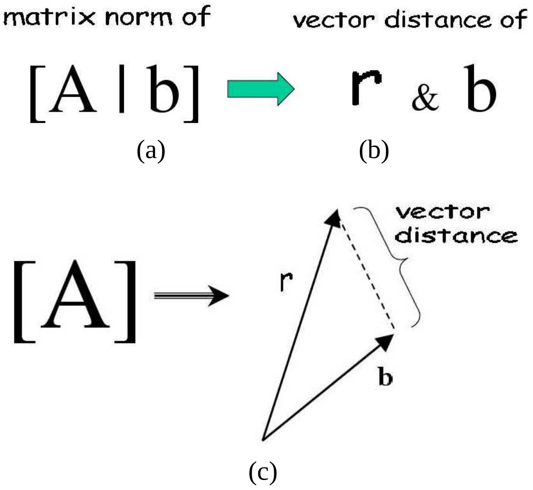


Figure 1. **This figure illustrates the** *vectorization process:* **(a) a matrix norm can be used to evaluate whether a candidate vector b should or should not be associated with A. (b) the reference matrix A is represented by a reference vector r. (c) A vector norm (e.g. the Euclidean norm can be used to measure the similarity between r and b, which will be rank equivalent to the similarity between A and b.**

Consider an $I \times J$ bicluster $A = \{a_{ij}; i = 1, \ldots, I \text{ and } j = 1, \ldots, J\}$. Assume that there are two column candidate vectors $\mathbf{b}$ and $\mathbf{b}'$ competing with each other to become an added member of bicluster A. The ranking between $\mathbf{b}$ and $\mathbf{b}'$ is traditionally dependent on the matrix norms such as $||[A|\mathbf{b}]||_{normal}$ and $||[A|\mathbf{b}']||_{normal}$. In order to simplify the analysis, we introduce a notion of vectorization, in which a "reference vector" and a "test vector" are defined, cf. Figure 1.

- If a row/column vector exists such that it can adequately represent the row/column properties of the current matrix, the vector will be called a "reference vector".
- Only one candidate row/column vector, namely "test vector", is evaluated in each step of the expansion of the matrix.

The usage of matrix norms and vector norms becomes interchangeable when the following conditions are met:

1. every matrix, say A, can be represented by a vector $\mathbf{r}$, $\mathbf{r} = f(A)$, and
2. there exists rank-equivalent (defined below) vector-norm corresponding to the targeted matrix-norm.

Definition 1 (Rank-Equivalence)

Two measurements (1) $\texttt{metric}_M$ *(a matrix norm) and (2)* $\texttt{metric}_V$ *(a vector norm) are said to be rank-equivalent if* $\texttt{metric}_M(A, \mathbf{b}) > \texttt{metric}_M(A, \mathbf{b}')$ *implies that* $\texttt{metric}_V(\mathbf{r}, \mathbf{b}) > \texttt{metric}_V(\mathbf{r}, \mathbf{b}')$*, and vice versa, where* $\mathbf{r} = f(A)$ *is the vector representation of the reference matrix* A*.*

The equivalence between several key matrix and vector norms (symbolically denoted by "$\Leftrightarrow$") is established in the following theorem.

Theorem 1 (Rank-equivalence of matrix and vector norms)

The vectorization processes of all of the following matrix norms lead to their corresponding **rank-equivalent** *vector norms. Such rank equivalence is denoted by "$\Leftrightarrow$".*

1. **Constant-value matrix norm:**

   $$||[A|\mathbf{b}]||_{constant-value} \Leftrightarrow ||[\mu, \mu, ..., \mu]^T - \mathbf{b}||$$

   where μ is the mean of all elements in A.

2. **Additive coherent model:**

   $$||[A|\mathbf{b}]||_{normal} \Leftrightarrow (\mathbf{r}, \mathbf{b})_{normal}$$

   where $\mathbf{r} = f(A)$ and $r_i \equiv \frac{1}{J}\sum_{j=1}^{J} a_{ij}$.

3. **Additive and multiplicative coherent model:**

   $$||[A|\mathbf{b}]||_{n\&s} \Leftrightarrow (\mathbf{r}, \mathbf{b})_{n\&s}$$

The proof is given in Appendix A.

3.3 Advantages of Vectorization

If the equivalence conditions are met, the matrix proximity metric can now be equivalently expressed as the distance between two vectors, i.e., the reference vector $\mathbf{r} = f(A)$ and the test vector. This process is called vectorization. There are many advantages of vectorization: (1) it leads to a substantial computational saving; and (2) the notion of vector distance considerably simplifies the analysis and facilitates visualization of gene/condition patterns.

3.3.1 Computational saving

Note that the complexity associated with different metrics can be very different. In fact, the computation burden for a matrix norm versus a vector norm can be drastically different even though they could be rank equivalent. More specifically, the formula in Eq. 18 has a complexity of $O(I)$. Therefore, by using the vector metrics, the evaluation of all the candidates would amount to a total time $O(NI)$, where N is the column size of the entire expression matrix. This represents a significant computational saving compared with the computation time of $O(NIJ)$ required by a nonrecursive method to compute the (equivalent) matrix norm.

Note that by swapping "row" and "column", the same recursive scheme can be used to perform row-wise expansion. By the same recursive scheme, the expansion of a matrix by one row would amount to a much reduced time $O(MJ)$, where M is the row size of the expression matrix.[3]

3.3.2 Visualization facilitated by vectorization

As evident in Figure 2, vectorization allows the separation between negative and positive patterns to become directly visualizable. This is illustrated by the FDA-like visualization of the 9-dimensional vectors corresponding to the 9 conditions selected in the biclustering process. This shows that univariate analysis will not yield as a good separation as multivariate analysis, pointing to the adoption of multi-metric fusion classifiers.

4 Univariate Analysis of Expression Data

The prevailing trend of existing biclustering algorithms is to adopt a univariate (i.e., single metric) for evaluating new members of a bicluster. The one-by-one biclustering scheme described in the following subsection adopts such an approach.

4.1 One-by-One Biclustering Scheme

A bicluster can be effectively formed in a step-by-step basis. In such a scheme, two strategies (expansion and contraction) may be adopted to identify each gene group (or condition group):

1. The contraction strategy starts with a bicluster of exceeding size, then from which one vector is pruned at a time, until all dissimilar vectors are eliminated from the group. For greater details, see [3].

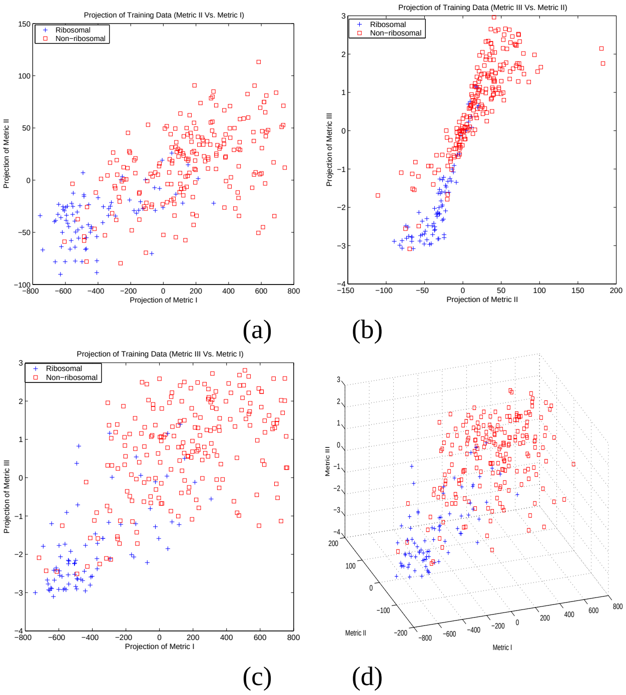


Figure 2. Vectorization allows an FDA-like visualization of vector-norm distances between genes under various coherence models. Here "+" represents positive (ribosomal) genes while square represents negative (non-ribosomal) genes. (a) additive-coherent row-preprocessing vs. no-preprocessing; (b) additive-multiplicative-coherent row-preprocessing vs. additive-coherent row-preprocessing; (c) additive-multiplicative coherent row-preprocessing vs. no-preprocessing; and (d) combined view of using all the three preprocessing schemes.

2. The expansion strategy, on the other hand, begins with a core set of vectors, then similar vectors are admitted to the group in a one-by-one basis. A proper criterion for expansion has to be designed so that it will first admit the candidate gene (or condition) that bears closest resemblance with the current subgroup. The process continues until all candidate vectors receive a proper evaluation and most (if not all) similar vectors are admitted to the group. The bicluster ultimately formed will depend on an intimate tradeoff between a maximum size (in terms of the number of genes/conditions) and a closest intra-group proximity.

Therefor, we advocate a "one-by-one" supervised clustering strategy for several motivations. First, plenty of prior information on known gene groups should be fully utilized to guide the grouping of genes. Furthermore, overlapping of groups allows a gene (or condition) to be simultaneously associated with multiple groups. Consequently, the focus

[3] Furthermore, the same recursive scheme can be used to perform row/column deletion from a matrix with the same cost-effectiveness.

is to determine whether a gene should be admitted by a gene group, instead of having to select the single best group (among multiple choices) to host the gene under consideration. This one-by-one grouping strategy, forming one-group at a time, was proposed by Mirkin (1996) [10], which starts with a single cell in the matrix and gradually expands it to reach a maximal constant bicluster.

There are two stages in the supervised training strategy: (1) training phase and (2) classification phase.

4.1.1 Training Phase

We start with the given knowledge of a certain gene group, and eventually derive a reference vector for the group. We consider the given gene group as positive (in-group) training data and the rest of the genes as negative (off-group) training data. The training procedure is as follows:[4]

1. Condition Initialization

   Based on the set of all (80) known ribosomal genes, we search the best condition pair (out of a total of $C_2^{17} - 17 = 119$ pairs) with the shortest distance. The names of ribosomal genes can be found in [13].

2. Condition Selection

   Starting from the two best conditions, we grow the condition group via the corresponding coherence model and stop the growing until the distance metric reaches a threshold. In this work, we found 9 conditions from the 17 conditions in the yeast data.

3. Gene Initialization

   Based on the 9 conditions, we search from the set of all (80) known ribosomal genes in the training set the best pair (out of a total of $C_2^{80} = 3080$ pairs) that gives the the shortest distance. At this point in the training procedure, our bicluster has a size of 2×9.

4. Gene Selection

   Starting from the 2×9 bicluster, we grow the bicluster (in gene dimension) by selecting a gene that is closest to the bicluster from all of the remaining genes ($= 2884 -$ *No. of genes in the bicluster*). The selected gene is then packed to the bicluster to form one with an additional row. The process is repeated until the distance between the selected genes and the current bicluter is smaller than a threshold.

Note that the searching phase involves searching for a row/column to be added to the current bicluster. We wish to select a row/column such that it bears the strongest resemblance with the current bicluster, i.e., it incurs a minimum increase in error residue. According to the theorem in Section 3.2, the search of candidate **b** can be replaced by a vector distance formulation $||\mathbf{b} - \mathbf{r}||$, where $\mathbf{r} = f(A)$ is the vector representation of A.

4.1.2 Classification/prediction phase

Test data are divided into two groups: in-group and off-group. The former includes the ribosomal genes in the bicluster plus those not in the bicluster, whereas the latter includes all non-ribosomal genes within or not within the bicluster. For those genes (ribosomal or not) that are in the bicluster, their corresponding residuals have been computed in the training phase and as a result do not need to be computed again. However, it is necessary to compute the residuals for those genes that are not in the bicluster. This can be achieved by packing those genes one-by-one to a base bicluster formed by the ribosomal genes found in the training phase. None of the genes used in the classification/prediction phase was used in the training phase.

Once we have the residuals of the in-group and off-group genes, we can classify the genes (g) by comparing their residual ($r(g)$) with a threshold (ζ), as follows:

$$\text{If } r(g) \begin{cases} < \zeta & g \text{ is ribosomal} \\ \geq \zeta & g \text{ is not ribosomal} \end{cases} \tag{7}$$

These binary decisions give us the number of false positives and the number of false negatives, from which we can compute the sensitivity, precision, and specificity.

4.2 Performance of Univariate Proximity Metrics

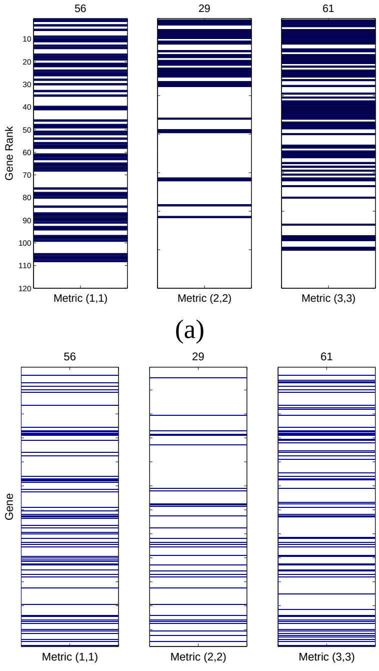
Figure 3 illustrates the ribosomal genes found in the biclusters based on various vector metrics: from left to right (1) constant-value model, (2) additive coherent model, and (3) additive-and-multiplicative coherent model. A bar represents the corresponding gene is ribosome (i.e., a true positive), whereas a white bar represents the corresponding gene is not ribosome (i.e., a false positive). The numbers on top of the diagrams are the numbers of genes in the biclusters that are in fact ribosome.

Figure 4 shows the sensitivity against precision of nine different combinations of preprocessing schemes for the conditions and genes. Evidently, different combinations lead to different gene classification/prediction performance.

4.3 Univariate versus Multivariate Biclustering

As evidenced by Figure 3, a gene group (blue bars) found by one metric can be substantially different from those

[4] For presentation simplicity, we assume that the training procedure is applied to the yeast data downloadable from [1]. However, the procedure is also applicable to other microarray data.



(a)

(b)

Figure 3. **Diagrams showing the positive and negative training genes in the biclusters successfully found by using various metrics: (1) constant-value metric, (2) additive coherent model, and (3) additive-and-multiplicative coherent model. (a) The bar-code is ordered in terms of ranking by various metrics. The numbers on top of the diagrams are the numbers of genes in the biclusters that are in fact ribosome. The veritcal axis is the "Gene Rank". A genes with lower ranks give smaller MSR than genes with higher ranks. (b) Now the vertical axis is the index corresponding to the gene name instead of its ranking. The figure suggests that the three metric provide complementary information, making them ideal candidates for multi-metric fusion.**

found by others. Figure 6 also shows that a single measurement leading to the discovery of one co-expressed group can obscure the finding of another differently co-expressed similarity groups. In Figure 2, when the positive and negative genes are projected onto one metric axis only, the amount of overlapping between the positive and negative genes becomes larger. Therefore, it will be difficult for a single metric to adequately explain the mutual regulation process between the genes.

The above results suggest that applying a single metric will fall short from an acceptable classification accuracy including sensitivity and specificity. Therefore, we advocate a new approach making use of combined metrics, which is the main theme of the subsequent section.

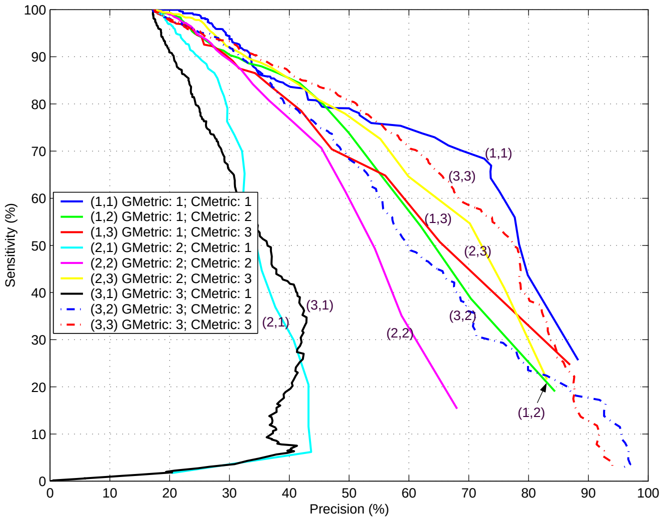


Figure 4. **For ribosomal genes, this illustrates the sensitivity against precision of nine different combinations of preprocessing processes for the conditions and genes. In the legend, "GMetric: m CMetric: n" mean that Metrics m and n were applied to the genes and conditions, respectively. In other words, it corresponds to Box (m,n) in Table 1.**

5 Multivariate Analysis

5.1 Sub-structural Study: Single Centroid Versus Multiple Centroids

Traditionally, it is assumed that a gene group can in general be adequately represented by a single substructure. However, it has been observed that there exists multiple substructures within the same (say, ribosomal gene) group. For example, as shown in Figure 6, the ribosomal gene group appear to contain at least two (if not more) substructures. Such an observation lead to two ideas:

1. When there are multiple substructures, the Gaussian Mixture Model (GMM) may be adopted to effectively model the subcluster structure. In other words, decision making may need to rely on more than one thresholds.

2. Vectorization may be a useful tool to enhance our understanding on the substructures and their impact on the various metrics.

One centroid would be sufficient to serve as the reference vectors if there is only a single substructure in the gene group of interest. Consider Cheng and Church's additive coherence model (with row normalization), the centroid is represented by one (and only one) reference vector, say

$\bar{\mathbf{r}}$. The single substructure assumption allows us to assume that most candidate vectors $\bar{\mathbf{b}}$ are centered around $\bar{\mathbf{r}}$, thus they can be expressed as $\bar{\mathbf{r}}$ corrupted by additive noise, i.e., $\bar{\mathbf{b}} = \bar{\mathbf{r}} + \mathbf{n}$:

$$(\mathbf{r}, \mathbf{b})_{normal} = ||\bar{\mathbf{b}} - \bar{\mathbf{r}}|| = ||\mathbf{n}|| = \sigma^2.$$

Without loss of generality, assume that there are two centroids, c.f. Figure 5, and they are represented by two reference vectors: $\bar{\mathbf{r}}$ and $\bar{\mathbf{r}}'$ with variances σ and σ', respectively. A candidate vector $\bar{\mathbf{b}}$ may be expressed as either $\bar{\mathbf{r}}$ (the primary centroid) or $\bar{\mathbf{r}}'$ (the secondary centroid), corrupted by additive noise:

$$\bar{\mathbf{b}} = \begin{cases} \bar{\mathbf{r}} + \mathbf{n} & \text{if it belongs to the primary substructure} \\ \bar{\mathbf{r}}' + \mathbf{n}' & \text{if it belongs to the secondary substructure} \end{cases} \tag{8}$$

Then the distance to the primary centroid will be (1)

$$(\mathbf{r}, \mathbf{b})_{normal} = ||\bar{\mathbf{b}} - \bar{\mathbf{r}}|| = ||\mathbf{n}|| = \sigma \tag{9}$$

if $\mathbf{b}$ belongs to the primary substructure; or (2)

$$\begin{aligned} (\mathbf{r}, \mathbf{b})_{normal} &= ||\bar{\mathbf{b}} - \bar{\mathbf{r}}|| = ||\bar{\mathbf{r}}' - \bar{\mathbf{r}} + \mathbf{n}'|| \\ &= \sqrt{(||\bar{\mathbf{r}}' - \bar{\mathbf{r}}||^2 + \sigma'^2)} = \sigma^* \end{aligned} \tag{10}$$

if $\mathbf{b}$ belongs to the secondary substructure.

It is therefore expected that the distance values of the gene group will fall into two regions, with the mean distance of one region approximately equal to σ while the other σ^*.

5.2 SVM-MOE Fusion Classifiers

There are plenty of supervised algorithms available for building classifiers that can combine the scores from different metrics, notably are SVM, multilayer perceptrons, and decision-based learning rules, see e.g., [8]. To design a more flexible classifier architecture, we propose an MOE architecture in which each local expert computes a local score based on a single metric. Thereafter, the (fuzzy) SVM [8] is used to fuse different scores to produce the final confidence. In a hierarchical MOE, a multi-subcluster structure could also be accommodated by each local expert module (see Figure 7).

The multivariate training strategy follows that of univariate evaluation. We advocate the adoption of both positive (in-group) and negative (off-group) training data. The data provide us two (or more) different metrics to be fused by the SVM classifier.

The choice of measurements for multivariate evaluation must be very different from that used for univariate evaluation. The selection criteria for the metrics adopted for the fusion network include:

1. The metric must by itself (i.e. univariate) deliver a sound performance.

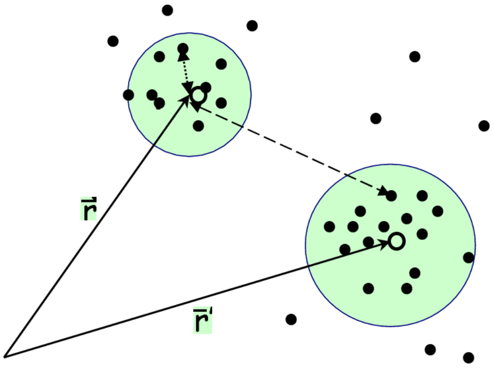


Figure 5. Analysis on multiple substructures: Vector representation of two reference vectors corresponding to two distinct substructures in the same gene group. Depicted here are the two distances from the candidate vectors (shown as solid circles) to the primary reference vector $\bar{\mathbf{r}}$. The shorter distance (the dotted line) suggests that the candidate is from the primary subcluster, cf. Eq. 9. The longer distance (the dashed line) indicates that the candidate is from the secondary subcluster, cf. Eq. 10. The theoretical prediction of two substructures is supported by Figure 10.

2. The metrics to be selected for fusion classifier must offer complementary information.

Theoretically speaking, constant-value metric, Box(1,1) in Table 1, and normalized-standardized metric, Box(3,3) in Table 1, would form a good team for fusion because they offer complementary information to each other. This is due to the fact that Box(1,1) uses least pre-processing (in fact none at all) while Box(3,3) has most preprocessing. The sensitivity-precision curves shown in Figure 8 also support the idea of teaming up the constant-value metric with the additive-multiplicative metrics for fusion purpose. The advantage of such selection of fusion metrics is shown in Figure 9(a). Figure 9(b) further shows that such fusion yields improvement in terms of sensitivity-specificity curves.

5.3 Performance of Multi-metric Fusion Schemes

Figure 10(a) illustrates the decision boundary created by an SVM and the training data. Figure 10(b) shows the distribution of test data with respect to the decision boundary obtained. Note that training data set and test data set are mutually exclusive.

Let us now take a closer look at the performance of the fuzzy-SVM (FSVM) in terms of its ability in classifying and/or discovering ribosomal genes. Also reported are cross-validation accuracies in terms of sensitivity, precision,

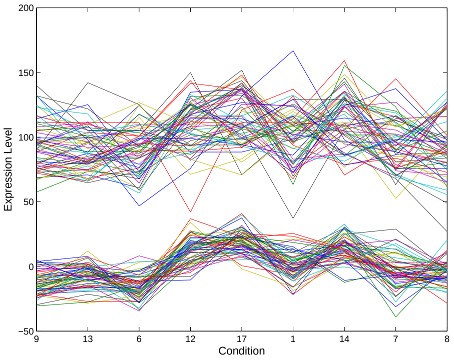


Figure 6. This figure illustrates the existence of substructures in the ribosomal gene group. The lower substructure will be a good choice for the primary reference vector, while the upper for the secondary reference vector. In viewing this figure, please keep in mind that an offset has been artificially added to the second substructure so that the difference of the waveform structures can be better displayed.

and specificity. The experiments are based on the yeast's microarray data set [1]. Figure 8 illustrates the sensitivity-precision curve based on 50 random simulations, to assure the statistically significance of the results. The sensitivity-precision curve corresponding to the fusion scheme clearly outperforms any univariate evaluations, including additive-coherent and additive-multiplicative-coherent metrics.

The color version of these figures and the Matlab programs that generate the results can be found in the web page [12] accompanying this paper.

6 Conclusion

In conclusion, let us summarize some novel ideas, which in the authors' opinion are quite distinct from the previous approaches, introduced by the paper:

1. A notion of vectorization allows the conversion of matrix-norm metrics into rank equivalent vector-norm metrics, leading to efficient gene/condition addition-deletion algorithms for expanding and/or contracting biclusters. It simplifies analysis and can potentially achieve computational saving. The notion of rank equivalence also allows a consolidation of many rank-equivalent metrics.

2. A novel multi-metric and multi-substructure fusion classifier is proposed under a mixture-of-experts architecture. This leads to a different criterion for the selection of optimal metrics to be fused. The performance improvement over univariate evaluation appears to be very promising.

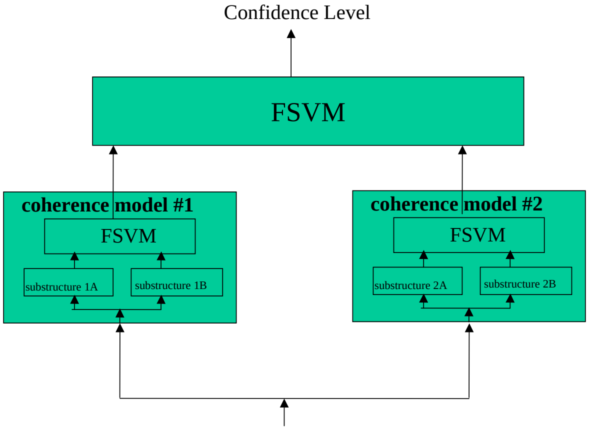


Figure 7. An MOE architecture with each local expert computing a score based on a single metric. Then different scores are combined by a fuzzy-SVM (FSVM) [8] to produce the final confidence level. In a hierarchical MOE, each local expert module could also accommodate a multi-subcluster structure.

7 Acknowledgements

The authors thank Mr. Chad Myers and Prof. Shang-Hung Lai of the Princeton University, for invaluable insights. We also appreciate the availability of microarray data published in the web site [1]. This work was supported in part by the Burroughs Wellcome Fund Fellowship and by the RGC of Hong Kong, Grant No. PolyU 5214/04E.

Appendix A: Proof of Rank Equivalence

1. The equivalence between the constant-value matrix norm and constant-value vector norm can be easily verified.

   Representing A by $\mathbf{r} = f(A)$:

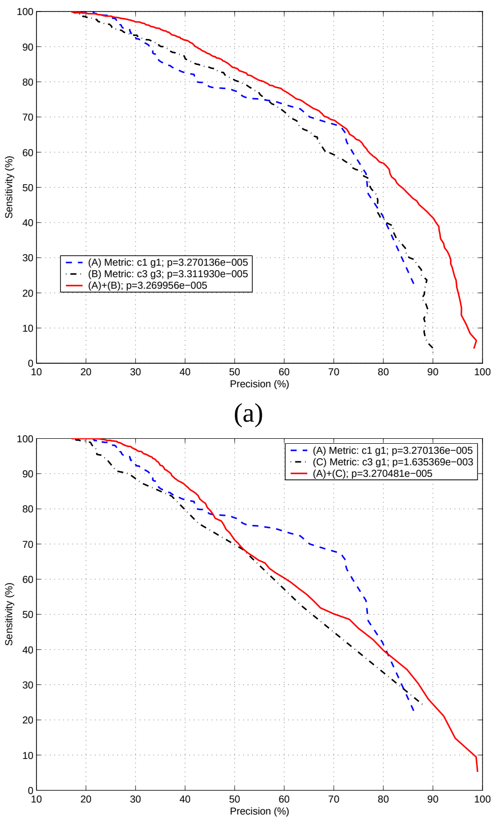
   $$\mathbf{r} = [\mu, \mu, \ldots, \mu]^T,$$

   where μ denotes the mean of all elements in A. Note that when J is a large number, then the mean of all elements in $[A|\mathbf{b}]$ is approximately equal to μ. Then we have

   $$\begin{aligned}&\min_c ||[A|\mathbf{b}] - c[E|[1, 1, \ldots, 1]^T]||_F \\ &\qquad \approx ||[A - \mu E]||_F + ||\mathbf{b} - [\mu, \mu, \ldots, \mu]^T||. \quad (11)\end{aligned}$$

   Noting that the first term does not depend on the candidate vector, only the second term has an effect on the ranking, therefore, the constant-value matrix norm is rank equivalent to the vector norm:

   $$||\mathbf{b} - [\mu, \mu, ..., \mu]^T||.$$



(a)

(b)

Figure 8. Sensitivity versus precision curves for (a) fusion of constant-value metric and additive-multiplicative metric and (b) fusion of additive metric with combination of additive-multiplicative metric and constant-value metric. In the legend, cngm means that Metrics m and n were applied to genes and conditions, respectively.

2. Without loss of generality, assuming the candidate vector is the $(J+1)^{th}$ (column) vector:

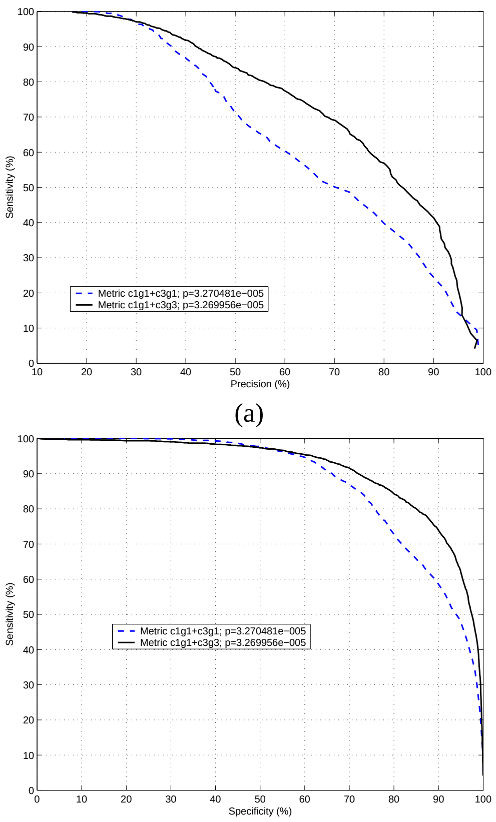
$$b_i \quad = \quad a_{i,J+1}. \tag{12}$$

The vector norm is denoted as $(\mathbf{r}, \mathbf{b})_{normal}$. The matrix norm is denoted as $r(I, J+1) = ||[\mathbf{A}|\mathbf{b}]||_{normal}$. The residue matrix norm $r(I, J+1)$, useful for measuring the homogeniety of $[A|\mathbf{b}]$, is rank-equivalent to the Euclidean distance

$$||\bar{\mathbf{r}} - \bar{\mathbf{b}}||.$$

where

$$\begin{aligned} \bar{r}_i &= \alpha_i(J) - \mu(I,J) && (13) \\ \bar{b}_i &= a_{i,J+1} - \beta_{J+1}(I) && (14) \end{aligned}$$



(a)

(b)

Figure 9. (a) The sensitivity-precision results supporting the idea of teaming up the constant-value metric with the additive-multiplicative-coherent metrics for fusion purposes. (b) Fusion performance in terms of sensitivity and specificity. In the legend, cngm means that Metrics m and n were applied to genes and conditions, respectively.

Consider an $I \times J$ bicluster $A = \{a_{ij}; i = 1, \ldots, I \text{ and } j = 1, \ldots, J\}$. We define the mean squared residual (MSR) as

$$r(I,J) = \frac{1}{IJ} \sum_{i=1}^{I} \sum_{j=1}^{J} \left(a_{ij} - \alpha_i(J) - \beta_j(I) + \mu(I,J)\right)^2, \tag{15}$$

where $\alpha_i(J) = \frac{1}{J}\sum_{j=1}^{J} a_{ij}$ are the row means, $\beta_j(I) = \frac{1}{I}\sum_{i=1}^{I} a_{ij}$ are the column means, and $\mu(I,J) = \frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J} a_{ij}$ is the overall mean of the bicluster A.

We have the following recursive relationships

$$\begin{aligned} \alpha_i(J+1) &= \alpha_i(J) + \frac{a_{i,J+1} - \alpha_i(J)}{J+1} && (16) \\ \mu(I,J+1) &= \mu(I,J) + \frac{\beta_{J+1}(I) - \mu(I,J)}{J+1} \end{aligned}$$

$$r(I, J+1) = \frac{1}{I(J+1)} \left\{ IJr(I,J) + \frac{J}{J+1} \sum_{i=1}^{I} \tilde{g}_i^2 \right\} \quad (17)$$

where

$$\tilde{g}_i = \beta_{J+1}(I) - \mu(I,J) - a_{i,J+1} + \alpha_i(J) = \bar{r}_i - \bar{b}_i \quad (18)$$

The last equality is due to Eq. 13 and Eq. 14. Eq. 17- Eq. 14 directly verify that $r(I, J+1) < r'(I, J+1)$ if and only if $(\mathbf{r}, \mathbf{b})_{normal} < (\mathbf{r}, \mathbf{b}')_{normal}$. Thus the proof. □

References

[1] arep.med.harvard.edu/biclustering.

[2] M. Bittner, et al. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, 406(3):536–540, Aug. 2000.

[3] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proc. Eighth Int'l Conf. Intelligent Systems for Molecular Biology (ISMB)*, volume 8, pages 93–103, 2000.

[4] D. J. Duggan, M. L. Bittner, Y. Chen, P. Meltzer, and J. M. Trent. Expression profiling using cDNA microarrays. *Nature Genetics*, 21:10–14, Jan. 1999.

[5] T. R. Golub, D. K. Slonim, C. H. P. Tamayo, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, Oct. 1999.

[6] J. Hartigan. Direct clustering of a data matrix. *J. Am. Statistical Assoc. (JASA)*, 67(337):123–129, 1972.

[7] D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: A survey. *IEEE Transactions On Knowledge And Data Engineering*, 16(11):1370–1386, 2004.

[8] S. Y. Kung, M. W. Mak, and S. H. Lin. *Biometric Authentication: A Machine Learning Approach*. Prentice Hall, Upper Saddle River, New Jersey, 2005.

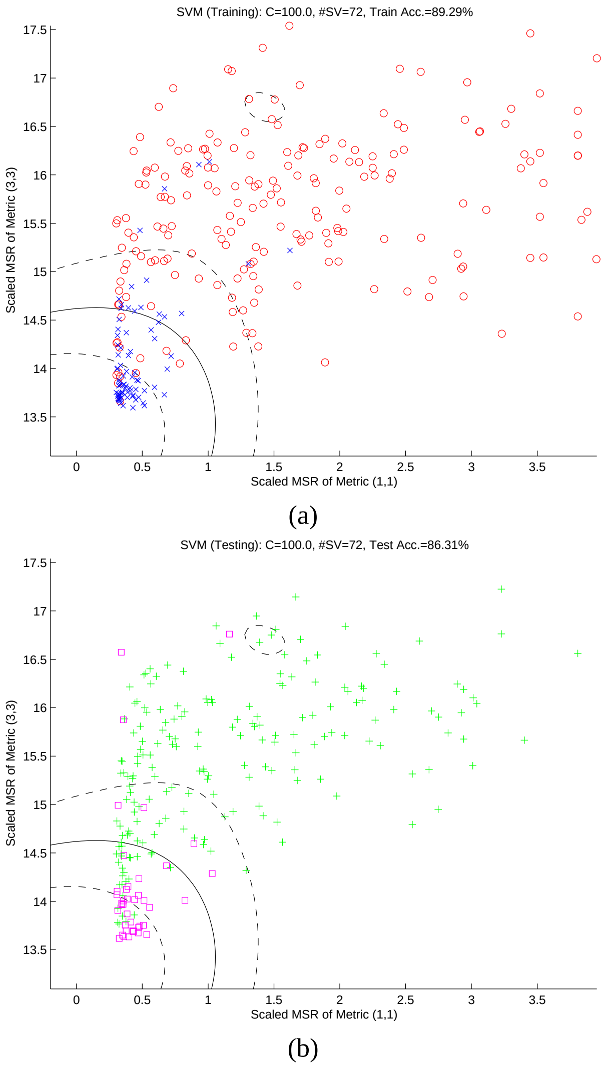
[9] L. Lazzeroni and A. B. Owen. Plaid models for gene expression data. Technical report, March, 2000, www-stat.stanford.edu/ owen/reports/plaid.pdf.

[10] B. Mirkin. *Math. Classification and Clustering*, chapter Nonconvex Optimization and its Applications. Kluwer Academic Publishers, 1996.

[11] S. Tavazoie, D. Hughes, M. Campbell, R. Cho, and G. Church. Systematic determination of genetic network architecture. *Nature Genetics*, pages 281–285, 1999.

[12] www.eie.polyu.edu.hk/ mwmak/microarray.htm.

[13] www.yeastgenome.org.



(a)

(b)

Figure 10. **Illustration of the use of 41 positive test patterns (ribosomal) and 200 test patterns (non-ribosomal) from the yeast data set. (a) The decision boundary is produced by an SVM classifier trained by 80 positive training data and 200 negative training data. Light blue crosses "X" represent positive training patterns. Light blue circles represent negative training patterns. (b) The prediction performance based on 41 positive test data and 200 negative test data. Light violet squares represent positive test patterns. Light green plus "+" represents negative test patterns. Note that the testing data and training data used here are mutually exclusive. The decision boundary is established by the training data only.**