Journal of Bioinformatics and Computational Biology

SYMMETRIC AND ASYMMETRIC MULTI-MODALITY BICLUSTERING ANALYSIS FOR MICROARRAY DATA MATRIX

SUN-YUAN KUNG

Department of Electrical Engineering, Princeton University, Engineering Quad, Princeton, New Jersey 08544, USA
kung@princeton.edu

MAN-WAI MAK

Department of Electronic and Information Engineering The Hong Kong Polytechnic University, Hung Hom, Hong Kong
enmwmak@polyu.edu.hk

ILIAS TAGKOPOULOS

Department of Electrical Engineering, Princeton University, Engineering Quad, Princeton, New Jersey 08544, USA
iliast@princeton.edu

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

Machine learning techniques offer a viable approach to cluster discovery from microarray data, which involves identifying and classifying biologically relevant groups in genes and conditions. It has been recognized that genes (whether or not they belong to the same gene group) may be co-expressed via a variety of pathways. Therefore, they can be adequately described by a diversity of coherence models. In fact, it is known that a gene may participate in multiple pathways that may or may not be co-active under all conditions. It is therefore biologically meaningful to simultaneously divide genes into functional groups and conditions into co-active categories – leading to the so-called biclustering analysis. For this, we have proposed a comprehensive set of coherence models to cope with various plausible regulation processes. Furthermore, a multivariate biclustering analysis based on fusion of different coherence models appears to be promising because the expression level of genes from the same group may follow more than one coherence models. The simulation studies further confirm that the proposed framework enjoys the advantage of high prediction performance.

Keywords: Microarray; gene expression patterns; biclustering; computational bioinformatics; finite mixture models; machine learning.

1. Introduction

The genome is not just a collection of genes working in isolation, but rather it encompasses the global and highly coordinated control of information to carry out

a range of cellular functions. Any cellular activity requires elaborate patterns of gene interaction to marshal appropriate processes. In addition, the genome also incorporates information that controls when and where the parts of living organisms should be made. Therefore, it is imperative to conduct genome-wide studies so as to facilitate (1) effective identification of correlated genes and (2) better understanding of the mechanisms underlying gene transcription and regulation.

Expression of several thousands of genes can be measured simultaneously by DNA microarrays. Microarrays have been effectively used to classify clinical samples, to investigate the mechanism of drug action and to examine the effects of drugs on gene expression in various organisms.[1,3,8] The upside of microarrays is that gene expression analysis is computationally less demanding than sequencing. Furthermore, recent advances in machine learning tools for expression profiling have become more mature and cost effective. However, microarrays also have their own limitations. In particular, the data are very noisy and contain artifacts, making gene prediction very difficult. Moreover, the gene dimension of the data matrix is usually too large (causing large search space) while the condition dimension is too small (causing statistic error).

Suppose that a set of (independent or time-course) microarray experimental data is obtained. The data is often framed into a data matrix, which can be expressed as 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. 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. Similarly, the profile of each gene is described as an N-dimensional vector in which each element represents the expression level of the corresponding condition.

For the development of microarray data mining tools, a critical and common approach is *cluster analysis*, i.e., grouping genes or conditions that have comparable patterns of variation of expression levels. Three types of cluster analysis have been studied:

(1) Clustering of genes, where genes are divided into functionally similar or genotypically related categories.
(2) Clustering of conditions, where conditions are divided into co-active or phenotype-related groups.
(3) Biclustering analysis, for which the goal is to simultaneously divide genes into functional categories and conditions into co-active groups.

Several distinct and challenging properties differentiating biclustering from traditional clustering are highlighted here.

(1) Since biclustering involves simultaneous grouping of both genes and conditions, the corresponding coherence models become more complex. Moreover, the suitable models for a gene/condition group are usually unknown a priori. To tackle

this problem, we study a comprehensive list of coherence models to first assure a broad spectrum of representation of the gene expression data. In order to pinpoint the most suitable coherent model(s), we shall apply supervised machine learning techniques to guide our selection process.

(2) It has been observed that many genes are co-expressed via a diversity of coherence models. In other words, a gene may be co-expressed via more than one coherence models. Therefore, a multi-modality adaptive fusion network is adopted to improve the performance in the prediction phase.

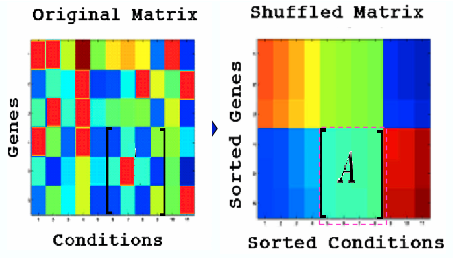
(3) Yet a third property pertaining to bicluster analysis – existence of overlapping between biclusters – is potentially advantageous in terms of simplification of the clustering procedure. Overlapping implies that genes with multiple functions may be simultaneously associated with more than one group. There is no loyalty issue, i.e., a gene (or condition) is no longer exclusively assigned to one cluster only. Such overlapping allows a gene or condition to be simultaneously associated with multiple biclusters. As there is no mutual dependence or conflict of interest among various biclusters, each of the biclusters can be independently searched. Thus, this paper adopts an independent bicluster searching strategy.

The biclustering analysis can be viewed as a special application of a general machine learning system. A machine learning system has in general three subsystems: (1) feature extraction, (2) cluster analysis, and (3) gene classification/prediction. More elaborately, the feature extraction subsystem is discussed in Section 2. Note that for biclustering analysis, the most crucial is not how to cluster data but how to find an appropriate way of looking at the data because the raw data may not be directly usable. This section proposes several preprocessing methods used for converting the raw data into a new representation that can better reflect the underlying coherence models. In Section 3, biclusters are identified via a supervised machine learning technique, i.e., a subset of genes from the targeted gene group are assumed to be known a priori. We used this small set of known genes to select a subset of co-active conditions. Then each gene can be represented by a feature vector, with the selected conditions as its elements. The feature vectors are then used as the basis for classification. Afterward, based on a (single) coherence model, the gene prediction can be conducted with performance recorded in terms of precision, sensitivity, and specificity. In Section 4, a multi-modality adaptive fusion network is proposed to further improve the prediction performance.

2. Biological Coherence Models for Microarray Data Matrix

2.1. *Basic Bicluster Criterion: Constant-Value Matrix Norm*

First of all, it is critical that we establish a precise and working definition on what constitutes a *bicluster* in the expression data. A bicluster is often based on a similarity function of the rows and columns in the expression matrix. Such a function is traditionally represented by the matrix norm of a submatrix of A. The most



(a) (b)

Fig. 1. Example illustrating an exhaustive approach to finding a constant-value submatrix. Ideally speaking, given a randomly ordered gene expression data matrix, one can try all the possible sorting orders on both genes and conditions. Hopefully, in one of such sorting results, we can find one most contiguous and at the same time flattest rectangular region. The corresponding gene and condition subsets can then be identified as a constant-value bicluster.

commonly used matrix norm is the Frobenius norm, denoted by $||A_s||_F$, which is defined as the square-root of the sum of squares of all elements in the submatrix A_s.

Hartigan proposed a constant-value matrix norm which has a very broad application spectrum.[4] In the definition, perfect biclusters are those with constant-value elements, denoted by c, in every matrix entry. The proximity of elements in a matrix depends on their deviation from an optimally chosen constant-value:

$$||A||_{\text{constant-value}} \equiv \min_c \left\| A - \begin{bmatrix} c & c & \dots & c \\ c & c & \dots & c \\ \vdots & \vdots & \vdots & \vdots \\ c & c & \dots & c \end{bmatrix} \right\|_F \tag{1}$$

where $||\cdot||_F$ denotes the Frobenius norm. The norm is smaller when the family is more similar. For example, a nearly perfect matrix is one such that

$$||A||_{\text{constant-value}} \approx 0.$$

Figure 1 shows an example in which a submatrix A can be identified as a constant-value matrix after shuffling the original matrix.

2.2. *Coherence Models and Preprocessing Methods*

For gene expression analysis, it is not only natural but also appealing to incorporate biologically relevant coherence models for both genes and conditions. However, in many practical situations, the constant-value matrix norm (Eq. 1) is simply not rich enough to handle the underlying complex biological process for co-regulated genes. For example, genes that have different expression levels but similar patterns may be co-regulated by a combination of transcription factors. Factors such as the transcription factor binding affinity, mRNA degradation rate, and transcription

initiation rate can produce displaced and scaled expression of gene products. There are also artifacts during the preparation of samples as well as systematic biases due to their heterogeneities (e.g., different patients).

In fact, several basic coherence models and their corresponding similarity measures for cluster discovery have already been proposed by microarray researchers.[16,12,2] In particular, two popular coherence models regulating the relative abundance of mRNA are *additive* and *multiplicative* coherence models and the corresponding preprocessing methods are normalization and standardization, respectively.[16,2] The raw data must be preprocessed to reflect the coherence models.

Let us further elaborate these two coherence models and their corresponding preprocessing methods.

(1) **Additive coherence model**: A scaling relation between mRNA_a and mRNA_b is expressed as $\text{mRNA}_b = \lambda(\text{mRNA}_a)$, where λ 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:[2] $b = \lambda' + a$ where $\lambda' \equiv \log(\lambda)$. There are several causes of the additive coherence model, e.g. (a) genes with different epigenetic modifications could be transcribed at different rates; (b) different topology of the promoter region could result in different efficiency of transcription of genes; and (c) different half lifetimes of the mRNAs could lead to different transcription rates.

Preprocessing for additive models: Two feature vectors **a** and **b** are said to belong to the same additive family if and only if they are equivalent except for a constant shift λ'. A "normalization" step is often adopted to alleviate the perturbation caused by the additive increments. Computationally, "normalization" is a process that subtracts each row (or column) by row (or column) mean, i.e., $(\mathbf{x} - \mu)$.

(2) **Multiplicative coherence model**: An exponential relation between mRNA_a and mRNA_b is expressed as $\text{mRNA}_b = (\text{mRNA}_a)^\gamma$. 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: $b = \gamma \times a$. Multiplicative coherence models can cope with the event of dissimilar transcription factor (TF) binding stoichiometry in a group. A simple example is when only a single TF molecule is needed to regulate a certain gene, but two or more TF molecules are required for the regulation of another. This model can also represent the situation when a modulator molecule(s) binds on multiple locations on the TF protein. The number of modulator molecules and the position they are bound affect regulation of target genes. In both cases, the mRNA concentrations of two co-regulated genes will exhibit a power law

Table 1. List of various coherence models for rows (genes) and columns (conditions). If the same preprocessing is applied to both rows and columns, it is referred to as a symmetrical coherence model. Otherwise, it is categorized into the asymmetrical coherence model. The entries in the table indicate the type of coherent model (far from being exclusive). For example, if normalization is applied to rows or columns (but not both), then the resulting clustering is equivalent to mean normalization. For Box (2,2), applying optimal normalization (c.f. Eq. 7) to both rows and columns will result in Cheng and Church's coherence model. The similar argument carries through to the remaining boxes. Here, the term "Generalized Z-norm" means that standardization is applied to rows and/or columns.

Coherence Model		Condition Coherence		
		No adjustment	Normalization	Standardization
Gene Coherence	No adjustment	Constant-value	Mean-normalization	Z-norm
	Normalization	Mean-normalization	Cheng and Church Type	Generalized Z-norm
	Standardization	Z-norm	Generalized Z-norm	Generalized Z-norm

dependency leading to a multiplicative model.[a] It is common to assume that the multiplicative variation is imposed on top of the additive variation. This leads to the "additive-multiplicative" coherence model: $b = \lambda' + \gamma \times a$.

Preprocessing for additive-multiplicative models: Two feature vectors **a** and **b** are said to be multiplicatively coherent if they are equivalent except for a constant scaling factor γ. Additive-multiplicative preprocessing is a "standardization" step that can be adopted to compensate for the additive as well as multiplicative variations. Computationally, "standardization" is a process that subtracts the expression levels by their mean and then divided by their variance, i.e., $((\mathbf{x} - \mu)/\sigma)$.

Normalization and standardization are especially important when the microarray contains a large assortment of random genes (or alternatively all genes in the species' genome), where we do not expect to see a large bias in each sample.[14] Figure 2 shows the effect of applying normalization preprocessing (additive coherence model) and standardization preprocessing (additive-multiplicative coherence model) on gene expression data.

2.3. *Comprehensive Coherence Models for Bicluster Discovery*

In order to explore a more comprehensive set of plausible coherence models, it is important that we provide a maximum flexibility in allowing all possible combinations of (row and column) preprocessing. This leads to two types of coherence models: symmetrical and asymmetrical models, as depicted in Table 1.

[a]For a detailed analysis of several coherence models and their biological derivation, see Tagkopoulos et al.[15]

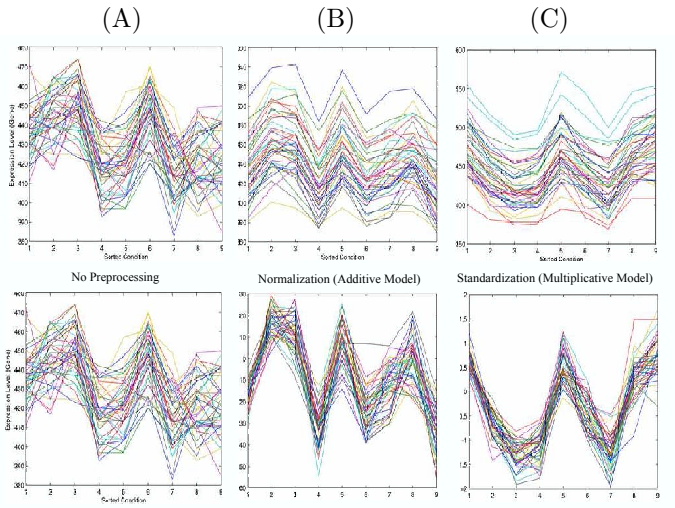


Fig. 2. Effect of applying preprocessing on raw expression data, including normalization (additive coherence model) and standardization preprocessing (multiplicative coherence model). In the graphs, each line represents the expression level of one gene across nine conditions in the yeast data set. Shown here are three time-course gene groups: A(upper-left), B(upper-center), and C(upper-right). By the naked eye, it would not be very persuasive to claim that Group B forms a tighter family than Group A, because the former exhibits a wider spread data structure. However, if we probe into the underlying data structure, it can then be revealed that Group B has a wider spread only because there is a large variation of the additive shifts. If such an additive variation can be properly compensated by additive preprocessing, cf. lower-center-box, the remaining spread becomes relative smaller than Group A. Similarly, Group C will be tightened up if the underlying multiplicative variation is also compensated in addition to additive normalization, cf. lower-right-box.

(1) **Symmetrical coherence models**: A prevailing assumption in the literature is that genes (rows) and conditions (columns) must share the same coherence model. Such a symmetry assumption leads to a type of preprocessing method in which the rows and columns undergo the the same kind of preprocessing. The symmetrical preprocessing methods are shown on the main diagonal boxes in Table 1, i.e., Models (1,1), (2,2) and (3,3). For example, for the center box, i.e., Model (2,2), "normalization" preprocessing is applied to both rows and columns, just like the Cheng and Church model.[2] Moreover, in the right-lower box, i.e., Model (3,3), the "normalization and standardization" preprocessing is applied to both rows and columns, leading to a generalized Z-norm. However, There is no biological evidence that the genes and the condition have to follow the same coherence model.

(2) **Asymmetrical coherence models**: The above-mentioned symmetry property is, however, neither truly biologically justifiable nor necessarily yielding a more effective tool for bicluster analysis. Therefore, in Table 1, six additional asymmetric models are introduced: (1,2), (1,3), (2,1),(2,3), (3,1) and (3,2). Ac-

Table 2. A numerical example that further elaborates the preprocessing operations corresponding to the coherent models listed in Table 1. Three types of preprocessing operations are used: no adjustment, normalization (i.e., $\mathbf{x} - \mu$), and standardization (i.e., $(\mathbf{x} - \mu)/\sigma$). The preprocessed matrices after the completion of the corresponding preprocessing procedures are listed in the table. For example, Box (2,1) is a result from Eq. 2 while Box (2,2) is from Eq. 4. Given a preprocessed matrix, the final similarity measure can readily be obtained by using Eq. 1.

Consider a matrix $A = \begin{bmatrix} 10 & 20 & 30 \\ 11 & 22 & 32 \\ 20 & 42 & 61 \end{bmatrix}$, with $\vec{\alpha} = \begin{bmatrix} 20 \\ 21.7 \\ 41 \end{bmatrix}$ and $\vec{\beta} = \begin{bmatrix} 13.7 \\ 28 \\ 41 \end{bmatrix}$. Here, the amount of adjustment in rows and column ($\vec{\alpha}$ and $\vec{\beta}$) has been set to row means and column means, respectively.

Preprocessing Methods		Column Preprocessing: No adjustment	Column Preprocessing: Normalization	Column Preprocessing: Standardization
Row Preprocessing	No adjustment	$\begin{bmatrix} 10 & 20 & 30 \\ 11 & 22 & 32 \\ 20 & 42 & 61 \end{bmatrix}$	$\begin{bmatrix} -3.7 & -8.0 & -11.0 \\ -2.7 & -6.0 & -9.0 \\ 6.4 & 14 & 20 \end{bmatrix}$	$\begin{bmatrix} -0.8 & -0.8 & -0.8 \\ -0.6 & -0.6 & -0.6 \\ 1.4 & 1.4 & 1.4 \end{bmatrix}$
	Normalization	$\begin{bmatrix} -10 & 0 & 10 \\ -10.6 & 0.3 & 10.3 \\ -21 & 1 & 20 \end{bmatrix}$	$\begin{bmatrix} -23.7 & -28.0 & -31.0 \\ -24.3 & -27.7 & -30.7 \\ -34.7 & -27.0 & -21.0 \end{bmatrix}$	$\begin{bmatrix} 0.8 & -1.0 & -0.7 \\ 0.6 & -0.3 & -0.7 \\ -1.4 & 1.4 & 1.4 \end{bmatrix}$
	Standardization	$\begin{bmatrix} -1.2 & 0.0 & 1.2 \\ -1.2 & 0.0 & 1.2 \\ -1.25 & 0.05 & 1.2 \end{bmatrix}$	$\begin{bmatrix} 0.02 & -0.03 & 0.02 \\ 0.00 & 0.00 & 0.00 \\ 0.01 & 0.03 & -0.01 \end{bmatrix}$	$\begin{bmatrix} 1.3 & -1.3 & 1.3 \\ -0.3 & 0.4 & -0.2 \\ -1.0 & 1.0 & -1.1 \end{bmatrix}$

cording to the simulation study in Section 3.2 (Figure 3), the performance of the super-diagonal asymmetric model (1,2) in Table 1 appears to outperform the symmetric model (2,2), i.e., Cheng and Church model.

2.4. *Combine Preprocessing with Constant-value Norm*

To determine the closeness of a bicluster pertaining to a specific coherence model, there are two stages involved: (1) apply the preprocessing method corresponding to the coherence model and (2) apply the constant-value matrix norm (Eq. 1) to measure the similarity of the gene/condition family.

(1) **Preprocessing**: For example, if additive preprocessing is applied to only the rows (or the columns or both), then the mathematical operations are as follows:

$$A_{\text{row-normalized}} = A - \vec{\alpha}[\ 1\ 1 \cdots\ 1] \tag{2}$$

$$A_{\text{column-normalized}} = A - [\ 1\ 1 \cdots\ 1]^T \vec{\beta}^T \tag{3}$$

$$A_{\text{both-normalized}} = A - \vec{\alpha}[\ 1\ 1 \cdots\ 1] - [\ 1\ 1 \cdots\ 1]^T \vec{\beta}^T \tag{4}$$

where the elements of $\vec{\alpha}$ and $\vec{\beta}$ reflect the amount of adjustment in rows and columns, respectively.

To help illustrate the operations of various preprocessing methods, numerical examples are provided in Table 2. For each box in Table 2, row operations

precede column operations.[b] For example, the preprocessed matrix in Box (2,2) is obtained as follows. Before preprocessing, we have

$$A = \begin{bmatrix} 10 & 20 & 30 \\ 11 & 22 & 32 \\ 20 & 42 & 61 \end{bmatrix}.$$

After row normalization (Eq. 2), we have

$$A_{21} = A - \vec{\alpha}[1 \;\; 1 \;\; 1] = A - \begin{bmatrix} 20.0 \\ 21.7 \\ 41.0 \end{bmatrix} [1 \;\; 1 \;\; 1] = \begin{bmatrix} -10 & 0 & 10 \\ -10.6 & 0.3 & 10.3 \\ -21 & 1 & 20 \end{bmatrix}.$$

After row and column normalization (Eq. 4), we have

$$\begin{aligned} A_{22} &= A_{21} - [1 \;\; 1 \;\; 1]^T \vec{\beta}^T \\ &= \begin{bmatrix} -10 & 0 & 10 \\ -10.6 & 0.3 & 10.3 \\ -21 & 1 & 20 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} [13.7 \;\; 28.0 \;\; 41] = \begin{bmatrix} -23.7 & -28.0 & -31.0 \\ -24.3 & -27.7 & -30.7 \\ -34.7 & -27.0 & -21.0 \end{bmatrix}. \end{aligned}$$

For Box(2,3), the matrix after row normalization is identical to A_{21}, and the matrix after column normalization and standardization is

$$\begin{aligned} A_{23} &= \left(A_{21} - \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} [-13.89 \;\; 0.44 \;\; 13.44] \right) \begin{bmatrix} 1/5.04 & 0 & 0 \\ 0 & 1/0.42 & 0 \\ 0 & 0 & 1/4.64 \end{bmatrix} \\ &= \begin{bmatrix} 0.8 & -1.0 & -0.7 \\ 0.6 & -0.3 & -0.7 \\ -1.4 & 1.4 & 1.4 \end{bmatrix} \end{aligned}$$

where $[-13.89 \;\; 0.44 \;\; 13.44]^T$ and $\text{diag}\{5.04, 0.42, 4.64\}$ contain the column means and column standard derivations of A_{21}, respectively.

(2) **Apply the constant-value matrix norm after preprocessing operations**: Once preprocessing corresponding to a specific coherence model is applied, the constant-value matrix norm can again be used to compute the closeness of the bicluster. For example, based on Eq. 4, Cheng and Church's residue can be computed as:

$$||A||_{\text{C\&C}} = ||A_{22}||_{\text{constant-value}} \tag{5}$$

$$= \left\| \begin{bmatrix} -23.7 & -28.0 & -31.0 \\ -24.3 & -27.7 & -30.7 \\ -34.7 & -27.0 & -21.0 \end{bmatrix} - \begin{bmatrix} -27.6 & -27.6 & -27.6 \\ -27.6 & -27.6 & -27.6 \\ -27.6 & -27.6 & -27.6 \end{bmatrix} \right\|_{\text{F}} \tag{6}$$

[b]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.

$$= \left\| \begin{bmatrix} 3.9 & -0.4 & -3.4 \\ 3.3 & -0.1 & -3.1 \\ -7.1 & 0.6 & 6.6 \end{bmatrix} \right\|_{\mathrm{F}} = 11.88.$$

If optimal normalization is applied to both columns and row as below,

$$||A||_{\mathrm{C\&C}} \equiv \min_{\vec{\alpha},\vec{\beta}} ||A - \vec{\alpha}[\ 1 \cdots 1] - [\ 1 \cdots 1]^T \vec{\beta}^T||_F. \tag{7}$$

It can be shown that (details omitted here) this leads to Cheng and Church's residue[c]

3. Bicluster Analysis Based on Single Coherence Model

Section 3.1 discusses how to discover bicluster in the supervised training phase, while Section 3.2 addresses the performance analysis in the gene prediction phase.

3.1. *Supervised Biclustering Scheme*

We adopted a supervised biclustering strategy proposed recently for condition selection.[11] There are two reasons to pursue a supervised biclustering scheme. One is due to the complexity of coherence models for bicluster discovery, i.e., the number of plausible coherence models for the bicluster analysis is far more than traditional cluster analysis. A more important justification is that there are plenty of prior information on some gene groups (e.g., ribosomal of yeast) ready to be utilized to guide the bicluster discovery.

Our procedure begins with a core set of vectors that are a priori known to be from the same gene group; 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 group. The process continues until all candidate vectors are properly evaluated and those closest vectors are admitted to the group. The bicluster ultimately formed will depend on an optimal tradeoff between a maximum size (in terms of the number of genes/conditions) and the closeness of intra-group genes/conditions.

Without loss of generality, we use an exemplar case to describe the general procedure. More precisely, we shall focus our discussion on the identification of

[c]We note that the optimal solution is not unique. One possible optimal solution is given below:

$$\vec{\alpha}_{\mathrm{opt}} = \frac{1}{J}\sum\nolimits_{j=1}^{J} \vec{a}_{\cdot j} - \frac{1}{2}\mu \quad \text{and} \quad \vec{\beta}^T_{\mathrm{opt}} = \frac{1}{I}\sum\nolimits_{i=1}^{I} \vec{a}_{i\cdot} - \frac{1}{2}\mu$$

where μ is the global mean of A (i.e., $\mu = \frac{1}{IJ}\sum_{i=1}^{I}\sum_{j=1}^{J} a_{ij}$), $\vec{a}_{\cdot j}$ and $\vec{a}_{i\cdot}$ are the j-th column and i-th row of A, respectively. Substituting the above into Eq. 7 results in

$$||A||_{\mathrm{C\&C}} = \frac{1}{IJ}\sum\nolimits_{i=1}^{I}\sum\nolimits_{j=1}^{J}\left(a_{ij} - \frac{1}{J}\sum\nolimits_{j'=1}^{J} a_{ij'} - \frac{1}{I}\sum\nolimits_{i'=1}^{I} a_{i'j} + \mu\right)^2,$$

which is the equation used by Cheng and Church.

ribosomal biclusters in the yeast data set.[5] To facilitate the supervised training process, we shall assume that 80 of the true ribosomal genes are provided. In other words, they are known a priori to belong to the *positive* category.

(1) **Condition initialization**: Based on the set of all (80) known ribosomal genes, we search the best condition pair (out of a total of $C_2^{17} = 136$ pairs) with the shortest distance. The names of ribosomal genes can be found in a yeast genome website.[7] Note that among the ribosomal genes specified in this website, 121 of them match the genes in the data set we obtained from Cheng and Church's website.[5] Among these 121 ribosomal genes, we randomly selected 80 to be the positive training data.
(2) **Condition selection**: Starting from the two best conditions, given a coherent model, we grow the condition group according to the corresponding closeness metric. We continue the growing until the closeness metric reaches a predefined threshold or the number of conditions reaches a predefined maximum value. Naturally, we wish to select a column such that it bears the strongest resemblance with the current condition group, i.e., it incurs a minimum increase in the closeness metric. For simplicity, we choose to fix the number of selected conditions to be 9 (out of the total of 17) conditions in the yeast data. After this condition selection process, we will have eighty 9-dimensional ribosomal gene vectors $\mathbf{x}_t^{(p)}$ for training the gene predictor (see Section 3.2).

After identifying a bicluster, one can proceed with the discovery of another gene bicluster. Before the search of the next gene group, there are two ways to handle those genes identified to be positive in the first gene group. If a gene has to be exclusively assigned to one and only one cluster, then the positive genes just selected must be first exempted from the gene pool before the next search commences. This precautionary step effectively prevents the same gene to be selected by another gene group.

On the contrary, we assume in this paper that there can be overlapping between different gene groups. More precisely, we allow a gene (or condition) to be simultaneously associated with multiple groups. Consequently, the positive genes can now remain in the gene set used in the next search. In other words, instead of determining one (and only one) most relevant gene group for a targeted gene, the focus is to determine whether a gene should be admitted to any gene group. Note that such *independent* bicluster searching strategy, forming one-group at a time, was proposed by Mirkin (1996),[13] which starts with a single cell in the matrix and gradually expands it to reach a maximal constant bicluster.

3.2. *Gene Prediction Based on Single Coherence Model*

To build classifiers for gene prediction, it is necessary to convert the feature vectors into scores. This can be achieved by using Fisher Discriminant Analysis (FDA) in which the positive (e.g., ribosomal) and negative (e.g., non-ribosomal) gene vec-

tors are projected onto a direction parallel to the vectors that connect the means of positive and negative gene vectors.[9] More precisely, let us denote the positive and negative gene vectors as $\mathbf{x}_t^{(p)}$ and $\mathbf{x}_t^{(n)}$, respectively.[d] The projection vector is obtained by $\mathbf{w} = (\bar{\mathbf{x}}^{(n)} - \bar{\mathbf{x}}^{(p)})/\|\bar{\mathbf{x}}^{(n)} - \bar{\mathbf{x}}^{(p)}\|$ where

$$\bar{\mathbf{x}}^{(p)} = \frac{1}{T_p}\sum\nolimits_{t=1}^{T_p} \mathbf{x}_t^{(p)} \quad \text{and} \quad \bar{\mathbf{x}}^{(n)} = \frac{1}{T_n}\sum\nolimits_{t=1}^{T_n} \mathbf{x}_t^{(n)}, \tag{8}$$

where T_p and T_n are the numbers of positive and negative training genes, respectively. Given a test gene vector $\mathbf{y}_t$, its FDA-projected value is

$$s(t) = \mathbf{y}_t^T \mathbf{w}, \tag{9}$$

where T represents transpose.

Let the distribution of the FDA-projected values $s(t)$'s corresponding to ribosomal and non-ribosomal be $p(s(t)|\Lambda^{(p)})$ and $p(s(t)|\Lambda^{(n)})$, respectively. A test gene t is classified as ribosomal if

$$\log p(s(t)|\Lambda^{(p)}) > \log p(s(t)|\Lambda^{(n)}); \tag{10}$$

otherwise it will be classified as non-ribosomal. By counting the number of misclassified test genes, we can compute the precision, sensitivity, and specificity corresponding to a single point on the ROC curve.[e] These performance measures are defined as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} = \frac{1}{1 + \text{FP}/\text{TP}} \tag{11}$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} = \frac{1}{1 + \text{FN}/\text{TP}} \tag{12}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{FP} + \text{TN}} = \frac{1}{1 + \text{FP}/\text{TN}} \tag{13}$$

where TP, TN, FP, and FN are the numbers of true-positives, true-negatives, false-positives, and false-negatives, respectively. Verbally speaking, precision is the proportion of predicted positive (i.e. ribosomal) genes that are indeed positive, sensitivity refers to the ability to correctly predict positive (i.e. ribosomal) genes, and specificity refers to the ability to correctly predict negative (i.e. non-ribosomal) genes. For example, perfect precision implies that every predicted positive genes is indeed positive, perfect sensitive implies that every positive gene (in the testing pool) must be predicted positive, and perfect specificity implies that every negative gene (in the testing pool) must be predicted negative.

[d]The derivation is simplified by assuming that the variances of the positive and negative clusters are the same, i.e., no weighting is applied to the vectors.
[e]ROC stands for receiver operating characteristic that displays the trade-off of two of these measurements over their entire range.

To produce the entire spectrum of sensitivity-precision-specificity, a disparity between the positive and negative log-likelihoods is introduced, i.e, a test gene t is classified as ribosomal if

$$\log p(s(t)|\Lambda^{(p)}) > \log p(s(t)|\Lambda^{(n)}) + \alpha, \tag{14}$$

where $\alpha > 0$ (respectively $\alpha < 0$) if a higher specificity (respectively sensitivity) is desired. Note that because $s(t)$'s are scalers, the ROC can also be obtained by sweeping a decision threshold ζ from the minimum to the maximum value of the test scores $s(t)$'s in the following decision rule:

$$\text{If } s(t) \begin{cases} < \zeta & t \text{ is ribosomal} \\ \geq \zeta & t \text{ is not ribosomal.} \end{cases} \tag{15}$$

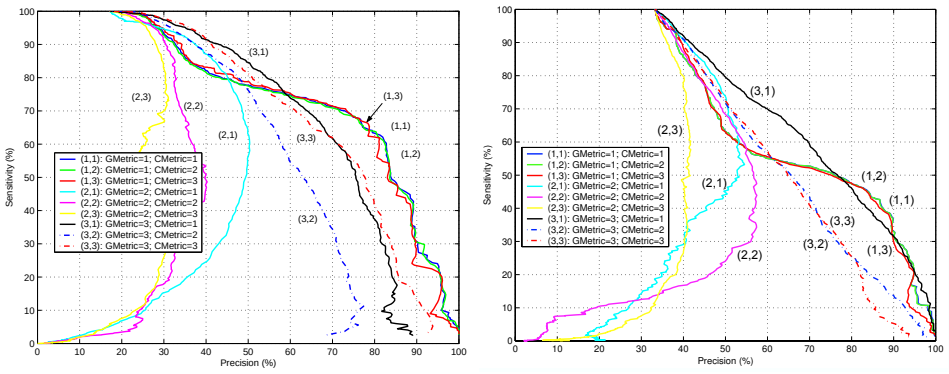
Figure 3 shows the sensitivity against precision (ROC) corresponding to nine different preprocessing methods for the detection of ribosomal genes. Note that Models (1,1), (1,2), and (1,3) have better performance in high precision region, while Model (3,1) performs better in low precision region. This provides very crucial information for the fusion strategy proposed in the next section.

Although it is true that when FP increases FN will decreases, when the change in TP overshadows the change in FP, both FP/TP and FN/TP could decrease, as illustrated in Figure 4. Therefore, when TP rapidly increases (decreases), both precision and sensitivity may be simultaneously compromised (enhanced). For example, for Models (2,1), (2,2), and (2,3), there is a region in the ROCs at which both sensitivity and precision are increasing (c.f. the lower-left region of Figure 3). Figure 4 also shows that the minimum of FP/TP (solid line) may not be equal to zero. This result together with Eq. 11 explain why in some cases the precision can never get higher than an upper limit.

4. Multi-Modaltiy Fusion: Combining Multiple Coherence Models

It is known that there may exist multiple sub-structure within the same gene group and the genes may participate in multiple pathways. Therefore, multi-modality fusion may be used as an effective tool for expansion (and possibly consolidation) of biclusters discovered via single-modality methods. As will be explained momentarily, there exists plenty of biological evidences to support the multi-modality nature of gene groups. In contrast, such an evidence is still very much lacking to support the application of multi-modality fusion to the expansion of condition groups. Therefore, in this section, we will only focus on the expansion of biclusters along the gene dimension, while leaving the condition dimension unchanged.

Figure 5 illustrates two possible fusion architectures. Figure 5(a) shows a direct fusion scheme, where the gene vectors $\mathbf{x}_t$ obtained by two preprocessing methods (Box (1,3) and Box (3,1) in Table 1) are concatenated to form an expanded vector. In the fusion layer, the long vector is processed by a supervised classifier (e.g., decision-based neural networks (DBNN)).[10]



(a) (b)

Fig. 3. Sensitivity against precision of nine different combinations of preprocessing methods for the conditions and genes. (a) Prediction of ribosomal. (b) Prediction of molecular activity genes. In the legend, "GMetric=m; CMetric=n" means that coherence models m and n were assigned to the genes and conditions, respectively. In other words, it corresponds to Box (m,n) in Table 1.

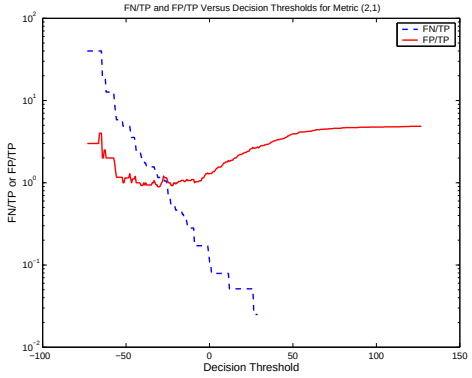
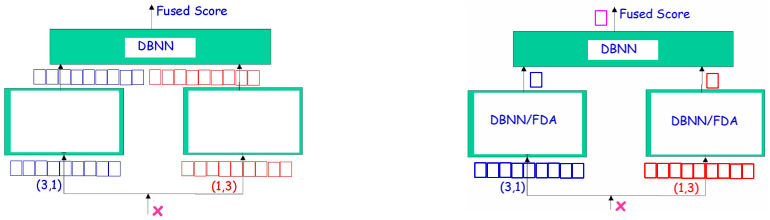


Fig. 4. False-negative/True-positive (dotted line) and False-positive/True-positive (solid line) versus decision thresholds for Metric (2,1) in Figure 3(a). Note that for thresholds less than −25, both FN/TP and FP/TP are decreasing, causing both sensitivity and precision to be simultaneously compromised or enhanced (c.f. the lower-left region of Figure 3).

Alternatively, an indirect fusion scheme is shown in Figure 5(b), where each feature vector is either compressed into a scalar (or low dimensional) feature or is represented simply by a local score. In the fusion layer, a supervised classifier can be adopted to combine the local features or scores. More specifically, we propose a Mixture-of-Expert (MOE) architecture in which each local expert computes a local score based on a single preprocessing method. Thereafter, the DBNN can be used to fuse the scores derived from various preprocessing methods to reach a final decision.[10]



(a) (b)

Fig. 5. Two possible Fusion Architecture. (a) Direct Fusion: multiple features are concatenated to form an expanded feature vector. (b) Indirect Fusion: each feature vector is compressed into a scalar (or low dimensional) feature or represented by a local score.

4.1. *Biological Support for Multi-Modality*

It is well known that genes may participate in more than one pathway; as a result, their expression profiles may be better explained by using multiple models. This fact holds even for genes that are members of a strongly correlated group, such as the ribosomal gene group. As an example, the products of genes YBR191W and YIL052C take part not only in protein biosynthesis but also in filamentous growth, a process by which an organism grows in a threadlike, filamentous shape. Therefore, it is not expected or necessary that a single model will be able to describe the behavior of correlated genes in all conditions. This calls for a fusion strategy that combines features produced by different preprocessing methods in order to improve classification and prediction performance.

In addition, it has been observed that there exist multiple substructures with different behaviors within the same (say, ribosomal gene) group. For example, as shown in Figure 6, the ribosomal gene group appears to contain at least two substructures. The existence of multiple substructures could result in a case where using one model would reveal a subgroup with high precision but not high sensitivity. This substructure, however, may be better characterized by another coherence model.

4.2. *Fusion of Coherence Models*

Hard-Switching and Consistent Fusion. A modest fusion objective is to deliver a *consistent fusion* result[9], which is at least as good as any of the single model in the entire sensitivity/specificity region. As long as the sources are complementary to each other (w.r.t. the ROC), consistent fusion is always possible and it will yield improvement as long as certain statistical conditions are met. Such a goal can be achieved by a hard-switching fusion scheme, described below:

(1) Determine the crossover point of two coherence models in Figure 7. Denote the sensitivity and specificity at the crossover point as $S_{crossover}$ and $P_{crossover}$,

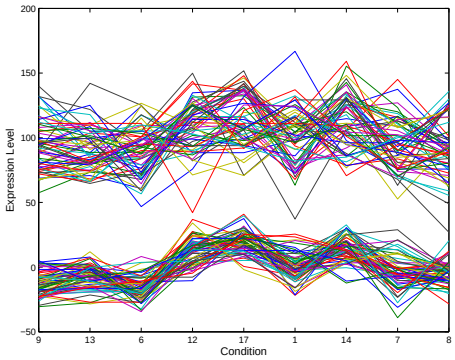


Fig. 6. This figure illustrates the existence of multiple substructures in the ribosomal gene group under the additive coherence model. Note that, to facilitate a better display of the difference of the substructures, an offset has been artificially added to one of the substructures so that the two substructures will no longer overlap.

respectively. For example, in Figure 7, the crossover of Model (1,1) and Model (3,3) is at the point $S_{crossover} = 75\%$ and $P_{crossover} = 87\%$.

(2) In order to guarantee a sensitivity to remain higher than or equal to $S_{crossover} = 75\%$, the decision boundary pertaining to Model (1,1) of the left expert in Figure 7(a) should be adopted. On the other hand, to assure a specificity no poorer than $P_{crossover} = 87\%$, we should switch to Model (3,3).

The decision boundaries for such a fusion scheme are illustrated in Figure 7(b). This scheme may assure that a lower bound performance of any consistent fusion is as good as the better of the two modalities. The consistence, however, requires some conditions to be met. More elaborately, suppose that a set of data is reserved in addition to the testing data set. Such data set is usually termed held-out data set. Let us assume that the fusion result is obtained based on the held-out data set only. Under such experimental procedure, a consistent performance can continue to hold up only under the additional assumption that the held-out data set shares the same statistics as the testing set.

Linear Fusion. Mathematically, denote the fusion score as Z, $Z = \alpha X + \beta Y$, where X and Y are input scores. In the hard-switching scheme, we have either $\alpha = 1,\ \beta = 0$ or $\alpha = 0,\ \beta = 1$. In contrast, one may adopt a linear soft fusion scheme based on a new fusion score $Z' = \alpha' X + \beta' Y$, where $\alpha' + \beta' = 1$. In many cases, such a soft fusion scheme can lead to better-than-lower-bound performance. The optimal values of α' and β' can better be derived via prominent machine learning techniques, such as Fisher classifiers and support vector machines (SVMs).[17] Unfortunately, it is known that linear classifiers often have limited discriminating power.

Nonlinear Fusion. In order to attain most flexible decision boundaries, one has

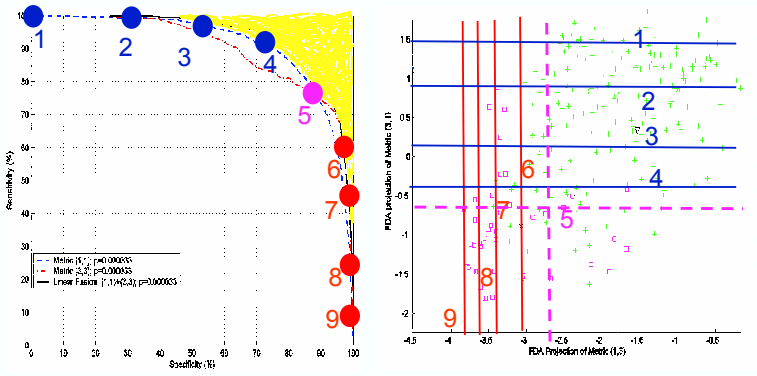


Fig. 7. Diagram illustrating the concept of consistent fusion. Note that the (horizontal) decision boundaries based on Model (1,1) – boundaries #1, #2, #3, and #4 – have relatively higher sensitivity, while the (vertical) decision boundaries based on Model (3,3) – boundaries #6, #7, #8, and #9 – have a relatively higher specificity. Therefore, the boundaries are switched from horizontal to vertical around the crossover point. At the crossover point, the boundary (#5) can be either horizontal or vertical as they deliver exactly the same sensitivity-specificity performance. In short, boundaries #1 to #5 are for high sensitivity while boundaries #5 to #9 are meant for high specificity.

to opt for nonlinear fusion schemes, which can be implemented by neural networks such as SVMs or decision based neural networks (DBNNs).[10] For the DBNN fusion classifier, there are two processing phases (c.f. Figure 5(b)): (1) a local expert (lower layer) uses a Gaussian mixture model to represent the patterns of the positive (or negative) class; (2) a "gating agent" (upper layer) can then be adopted to fuse the local scores and reach a Bayesian optimal decision.

4.3. *Adjustment of Decision Boundary for ROC Evaluation*

Without loss of generality, let us consider the indirect fusion of scores derived from two preprocessing methods. The ROC can be obtained by extending the idea introduced in Section 3.2 to multi-dimensional cases. More specifically, $s(t)$ in Eq. 14 become two-dimensional vectors $\mathbf{s}(t) = [s_1(t)\ s_2(t)]^T$ comprising of FDA-projected scores derived from two preprocessing methods (see Eq. 9), and $\Lambda^{(p)} = \{\pi_i^{(p)}, \mu_i^{(p)}, \Sigma_i^{(p)}\}_{i=1}^{K^{(p)}}$ and $\Lambda^{(n)} = \{\pi_i^{(n)}, \mu_i^{(n)}, \Sigma_i^{(n)}\}_{i=1}^{K^{(n)}}$ are 2-D Gaussian mixture models representing the positive and negative genes, respectively. By simply counting the number of test vectors $\mathbf{s}(t)$ falling on the wrong side of the decision boundary, we can estimate the precision, specificity, and sensitivity corresponding to a single point on the ROC curve. The entire spectrum of sensitivity-precision-specificity and their corresponding decision boundaries (see Figure 8) can then be obtained by adjusting the value of α in Eq. 14.

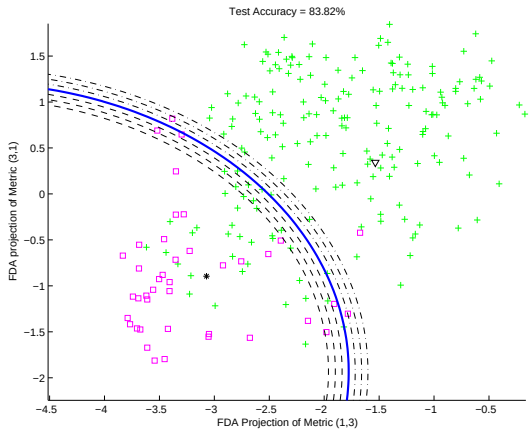


Fig. 8. Illustration of the use of 41 positive test patterns (ribosomal, represented by pink "□") and 200 test patterns (non-ribosomal, represented by green "+") from the yeast data set. The decision boundary is produced by a DBNN classifier trained by 80 positive training data and 200 negative training data. The dashed and dashed-dotted curves represents the decision boundaries for decreasing and increasing decision thresholds α, respectively. The solid-blue line represents the decision boundary when $\alpha = 0$ in Eq. 14.

4.4. *Which Models to Fuse*

It is important to have a good criterion to select proper features to fuse. First of all, the individual features must by themselves deliver a sound performance. Moreover, they must offer complementary information. On one hand, fusion of features which are positively correlated can assure improvement by fusion. On the other hand, features with strong resemblance may prove to be too redundant to produce a useful fusion. Fortunately, ROCs provide a very clear indication on when and which features are most advantageous to fuse. The ROC curves as depicted in Figure 3 show the sensitivity against precision for all the nine coherence models. Indeed, such ROCs offer an effective tool for selecting candidate models for fusion. For example, Model (3,1) has a relatively higher sensitivity in the low-precision region but a relatively lower sensitivity in the high-precision region. In contrast, Model (1,3) has just the opposite performance. In this case, these two models are truly complementary to each other and can serve as ideal fusion candidates as far as sensitivity-precision ROC is concerned.

4.5. *Gene Prediction Based on Multi-Modality Fusion*

Figure 8 illustrates the test data and the decision boundary created by a DBNN for the classification of ribosomal and non-ribosomal genes in the yeast's microarray data set.[5] It was found that a single elliptical basis-function per class is adequate—this sufficiently simplifies the training phase.

Let us now take a closer look at the cross-validation accuracies in terms of sensi-

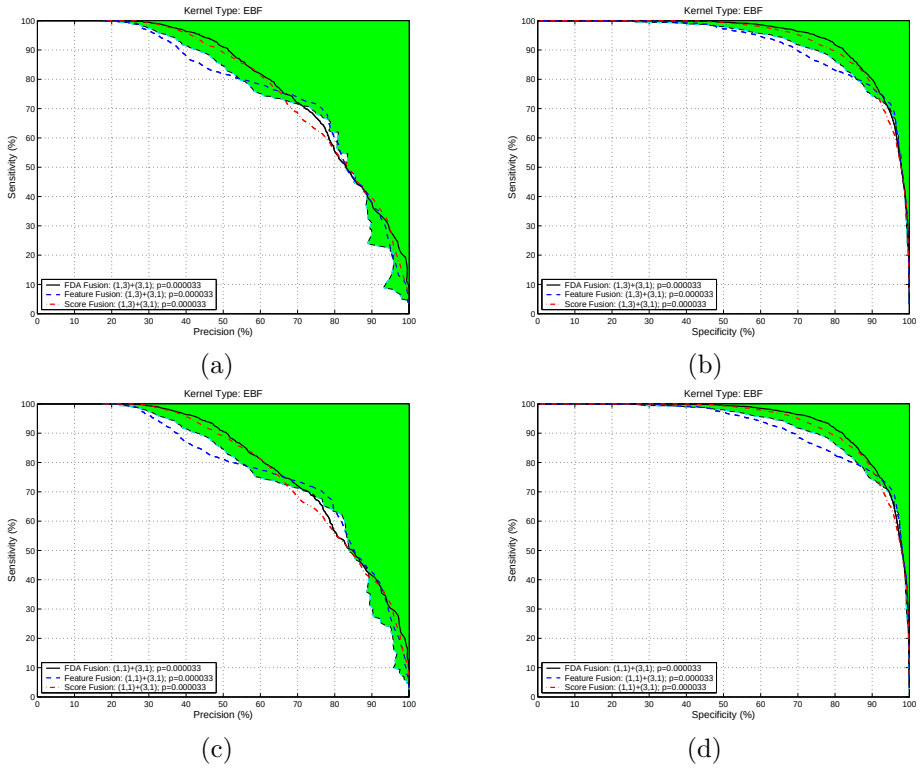


Fig. 9. The sensitivity-precision-specificity results of ribosomal detection supporting the idea of teaming up the constant-value coherence model with the additive-multiplicative coherent model for fusion purposes. (a) and (b) fusion of Models (1,3) and (3,1); (c) and (d) fusion of Models (1,1) and (3,1). In the legend, (m, n) means that coherence models m and n were assigned to genes and conditions, respectively. The green area represents the region of consistent performance.

tivity, precision, and specificity. Figure 9 illustrates the sensitivity-precision curves and sensitivity-specificity curves based on various fusion models: (1) direct (feature) fusion and (2) indirect (FDA or score) fusion. To assure statistically significance, each curve is based on 50 simulations, each with a different set of training genes. Evidently, the fusion of FDA-projected scores attains the highest performance, which is followed by the fusion of DBNN scores. The results show that feature fusion is consistently inferior to other fusion approaches. This may be attributed to the large feature dimension after feature concatenation. Although the DBNN in feature fusion considers all features (18 dimensions in this case), it may have difficulty in modelling the distribution of 18-dimensional vectors, given the limited amount of positive training vectors. On the other hand, the score fusion and FDA fusion consider the feature vectors (9 dimensions) derived from individual preprocessing methods independently, which helps alleviate the difficulty encountered by feature fusion.

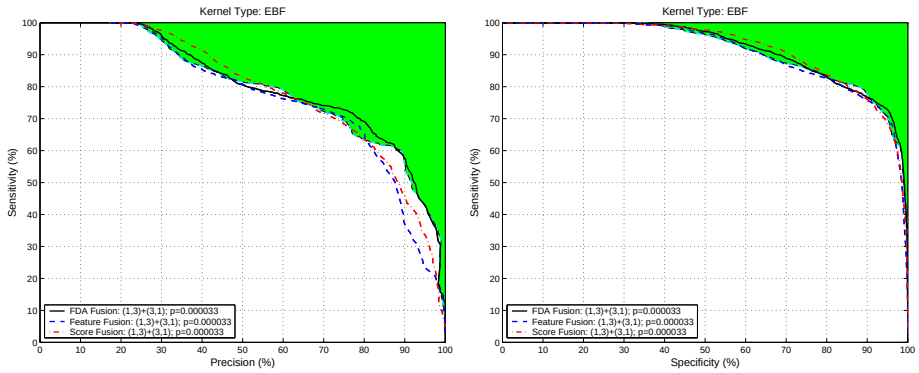


Fig. 10. The performance of ribosomal detection using all of the 17 conditions in the yeast data set and the fusion of Models (1,3) and (3,1). (a) Sensitivity vs. precision. (b) Sensitivity vs. specificity. The green area represents the region of consistent performance.

It is of interest to see the performance when all of the 17 conditions are used for the FDA projection. Figure 10 shows the ROC of different types of fusion. A comparison between Figure 10(a) and Figure 9 suggests that when we aim at achieving high precision, we may opt for using more conditions. On the other hand, if achieving high sensitivity is the goal, we may instead prefer a selected subset of conditions. It is not uncommon to have different performance requirements in different applications, i.e., some applications may need high precision and some others high sensitivity. Our results suggest that the number of conditions is an important factor that determines the ROC performance. Therefore, the number of conditions can be optimized to best serve the purpose of a particular application.

Based on the above gene prediction method, the multi-modality fusion approach can also be used to expand the row size of a bicluster. For example, the row size of the ribosomal bicluster was 80 genes in the training set. After gene prediction, the row size was expanded to $(80 + \mathrm{TP} + \mathrm{FP})$, where TP and FP are the numbers of true-positives and false-positives in the prediction, respectively. Figure 11 shows the histogram of the expanded row size when the precision is set to be around 70%. The expansion has to be reduced (or increased) if higher precision (or sensitivity) is desired.

Relatively speaking, the ribosomal group is considered by us as an "easy case", while the molecular activity group, for example, is considered as a "hard case". (The latter is however by far not the hardest case we are aware of.) Figure 12 illustrates the performance of detecting the molecular activity gene group. Unlike the detection of ribosomal, fusion could not achieve dramatic improvement for the molecular activity group; however, fusion does improve the detection performance at the region of high precision.

The Matlab programs that produce the results in this paper can be found in the supplementary website.[6]

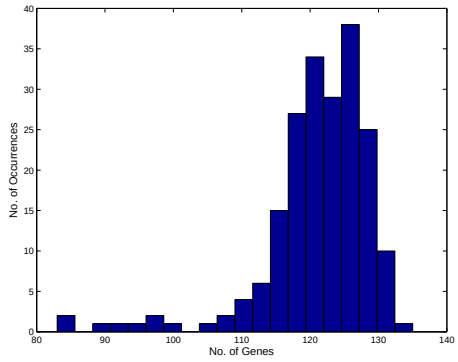


Fig. 11. Histogram showing the row size of biclusters after fusing the scores of Models (1,3) and (3,1). The histogram is produced by running the fusion algorithm 200 times, each with a different set of training and test genes. For each run, the numbers of true-positives and false-positives are obtained by setting the decision threshold such that precision is closest to 70%.

5. Conclusion

In conclusion, preprocessing models (both symmetric and asymmetric) are found to be useful for coping with various genotypical co-expression coherence models. The approach considerably simplifies bicluster analysis. A multi-modality fusion classifier based on a DBNN mixture-of-experts architecture is proposed. The proposed coherence models and fusion methods have successfully achieved highly discriminant and accurate classification of known ribosomal gene group (easy case) and molecular activity gene group (hard case).

6. Acknowledgment

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 Cheng and Church's web site.[5] This work was supported in part by the Burroughs Wellcome Fund Fellowship and by the Research Grant Council of Hong Kong SAR, Grant No. PolyU 5214/04E.

References

1. M. Bittner, et al. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*, 406(3):536–540, Aug. 2000.
2. 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.
3. 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.
4. J.A. Hartigan. Direct clustering of a data matrix. *J. Am. Statistical Assoc. (JASA)*, 67(337):123–129, 1972.
5. http://arep.med.harvard.edu/biclustering.

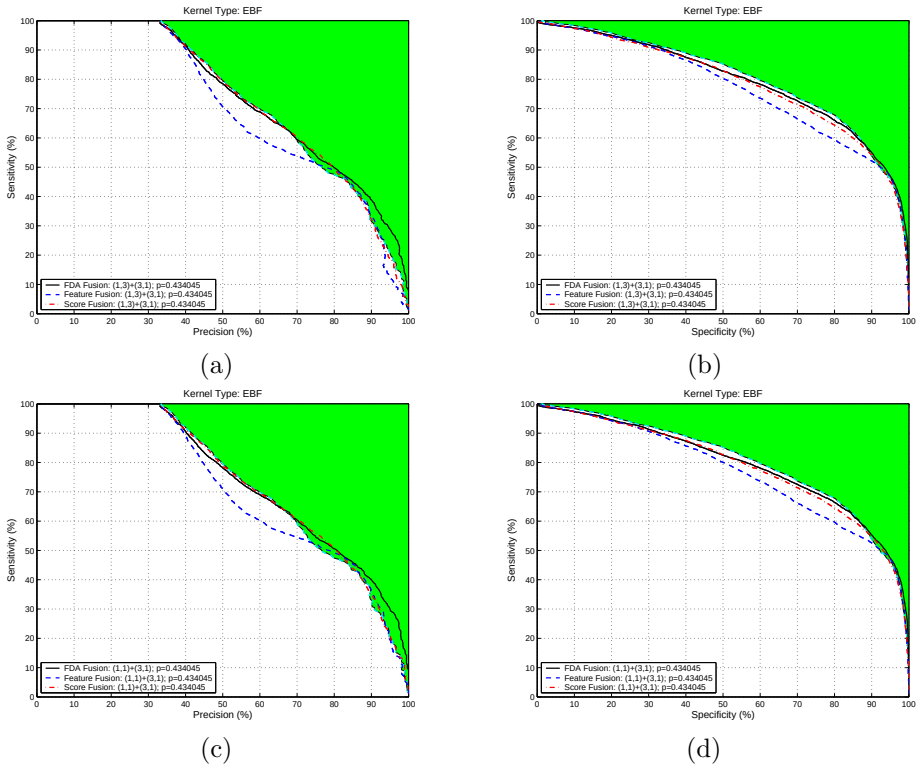


Fig. 12. The sensitivity-precision-specificity results of the molecular activity gene group. (a) and (b) fusion of Models (1,3) and (3,1); (c) and (d) fusion of Models (1,1) and (3,1). Refer to Figure 9 for the meaning of legends.

6. http://www.eie.polyu.edu.hk/∼mwmak/microarray.htm.
7. http://www.yeastgenome.org.
8. 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.
9. S. Y. Kung and M. W. Mak. A machine learning approach to DNA microarray biclustering analysis. In *2005 IEEE International Workshop on Machine Learning for Signal Processing*, Mystic, Connecticut, Sept. 2005.
10. S. Y. Kung, M. W. Mak, and S. H. Lin. *Biometric Authentication: A Machine Learning Approach*. Prentice Hall, Upper Saddle River, New Jersey, 2005.
11. S. Y. Kung, M. W. Mak, and I. Tagkopoulos. Multi-metric and multi-substructure biclustering analysis for gene expression data. In *IEEE Computational Systems Bioinformatics Conference*, Stanford University, California, Aug. 2005.
12. L. Lazzeroni and A. B. Owen. Plaid models for gene expression data. Technical report, Stanford University, March, 2000, www-stat.stanford.edu/∼owen/reports/plaid.pdf.
13. B. Mirkin. *Math. Classification and Clustering*, chapter Nonconvex Optimization and its Applications. Kluwer Academic Publishers, 1996.
14. J. Quackenbush. Microarray data normalization and transformation. *Nature Genetics Supplement*, 32:496–501, 2002.

15. I. Tagkopoulos, N. Slavov, and S.Y. Kung. Multi-class biclustering and classification based on modeling of gene regulatory networks. In *Proceedings, IEEE Symposium on Bioengineering and Bioinformatics, BIBE2005*, Minneapolis, Minnesota, 2005.
16. S. Tavazoie, D. Hughes, M.J. Campbell, R.J. Cho, and G.M. Church. Systematic determination of genetic network architecture. *Nature Genetics*, pages 281–285, 1999.
17. V. N. Vapnik. *Statistical Learning Theory.* Wiley, New York, 1998.