A MACHINE LEARNING APPROACH TO DNA MICROARRAY BICLUSTERING ANALYSIS

S.Y. Kung

Dept. of Electrical Engineering
Princeton University
kung@princeton.edu

Man-Wai Mak

Dept. of Electronic and Information Engineering
The Hong Kong Polytechnic University
enmwmak@polyu.edu.hk

ABSTRACT

Based on well-established machine learning techniques and neural networks, several biclustering algorithms can be developed for DNA microarray analysis. It has been recognized that genes (even though they may belong to the same gene group) may be co-expressed via a diversity of coherence models. One convincing argument is that a gene may participate in multiple pathways that may or may not be co-active under all conditions. It is biologically more meaningful to cluster both genes and conditions in gene expression data – leading to the so-called biclustering analysis. In addition, we have developed a set of systematic preprocessing methods to effectively comply with various coherence models. This paper will show that the proposed framework enjoys a vital advantage of ease of visualization and analysis. Because a gene may follow more than one coherence models, a multivariate biclustering analysis based on fusion of scores derived from different preprocessing methods appears to be very promising. This is evidenced by our simulation study. In summary, this paper shows that machine learning techniques offers a viable approach to identifying and classifying biologically relevant groups in genes and conditions.

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 [2, 3, 4]. Gene expression data are arranged 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. It has been shown that clustering and/or biclustering analysis on microarray data is instrumental in extracting vital and biologically meaningful information from large gene expression data sets.

The notion of biclustering was first introduced by Hartigan [5] in 1972 to describe simultaneous grouping of both row and column subsets in a data matrix. It has been specialized for genomic grouping by a number of researchers [6, 7, 8, 9], in which biclustering involves grouping a subset of genes and a subset of conditions with a high similarity score. To this end, the similarity measure must reflect the coherence of the genes and conditions in the bicluster.

There is an intimate relationship between clustering and biclustering, and their interplay is in general inevitable. Nevertheless, several distinct and challenging properties differentiating biclustering from traditional clustering.

1. *Challenge:* In biclustering, the coherence models are usually unknown a priori.
   Solution: We propose to preprocess the gene expression data to support a comprehensive list of coherence models. We also applied supervised learning to determine the most suitable model or models, and then followed through with the training and prediction phases.

2. *Challenge:* Biclustering involves simultaneous clustering of both genes and conditions.
   Solution: We used a small set of known genes to select the most coherent set of conditions. Then, we considered the genes as multi-dimensional vectors (with the selected conditions as elements) and applied machine learning techniques to classify the genes.

3. *Challenge:* Existence of overlapping between biclusters 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 biclusters.
   Solution: We introduced a one-by-one (self-contained) bicluster forming strategy, i.e., biclusters are formed one at a time. This strategy allows a gene to be associated with more than one bicluster.

2. PREPROCESSING IN ACCORDANCE WITH BIOLOGICAL COHERENCE MODELS

2.1. Basic Bicluster Criterion: Constant-Value Matrix Norm

The first question to address is how to define a bicluster of expression data. Traditionally, such a function is represented by a matrix norm of a sub-matrix of A.[1] 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 sub-matrix.

Hartigan [5] proposed a constant-value matrix norm which has a very broad application spectrum. In the definition, a perfect bicluster is one with constant value, denoted by c, in every matrix en-

---

[1]To simplify notation, we will use A to denote both the whole matrix and its sub-matrices when no confusion occurs.

try. The proximity of elements in a matrix depends on the amount of deviation from a constant-value:

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

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.

2.2. Biological Coherence Models

The biological processes that govern the gene expression data should be taken into account in order to obtain a useful bicluster of gene expression data. Two popular coherence models regulating the relative abundance of mRNA are additive and multiplicative coherence models. Their corresponding preprocessing processes are normalization and standardization, respectively.

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

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

   allows conversion of multiplicative changes of the relative abundance into additive increments [9]: $b = k' + a$ where $k' \equiv \log(k)$.

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: $b = c \times a$.

2.3. Biologically-driven Preprocessing Models

For gene expression analysis, it is not only natural but also appealing to incorporate biologically relevant coherence models. Preprocessing is an effective means of achieving this goal. It has long been recognized that normalizing and/or standardizing rows (genes) and/or columns (conditions) could improve significantly the biclustering performance.

1. **Preprocessing for additive coherence models:** A "normalization" step is often adopted to alleviate the uncertainty caused by the additive increments. Computationally, "standardization" is a process that subtract each row (or column) by row (or column) mean.

2. **Preprocessing for multiplicative coherence models:** Multiplicative coherent preprocessing is a "standardization" step that can be adopted to counter the uncertainty incurred by the multiplicative coherent model. Computationally, "standardization" is a process that divides each row (or column) by its standard deviation.

2.4. Comprehensive Preprocessing Models

In order to provide a more comprehensive platform for all plausible coherence models, it is important that we explore various combinations 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., 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 [9]. 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 Tavazoie-type test [7].[2]

2. **Asymmetrical preprocessing 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 the conclusion that the rows and columns must adopt the same kind of preprocessing. However, such a symmetry property is neither truly biologically justifiable nor is it necessarily most effective. For example, the sub-diagonal box Model (2,1) in Table 1 is simply based on the traditional K-means clustering on rows. According to the simulation study in Section 3.2, c.f. Figure 2, the performance of some asymmetrical coherence models appear to be very promising. In fact, most of them outperform Model (2,2), i.e., Cheng and Church criteria.

To help clarify the definitions/operations of various preprocessing models, numerical examples are explicitly provided in Table 2.

2.5. Combine Preprocessing with Constant-value Norm

After some proper preprocessing, the same constant-value matrix norm can again be adopted to measure the similarity between genes or conditions. There are two stages involved:

1. **Preprocessing before matrix norm**

   There are two types of preprocessing: additive coherent preprocessing and multiplicative preprocessing. Computationally, "normalization" is a process that subtracts the mean from each row (or column). If preprocessing is applied to only the rows (or only the columns or both), then the mathematical operations are as follows:

   $$
   \begin{aligned}
   A_{row-normalized} &= A - \vec{\alpha}[\,1\,1\cdots\,1] & (1)\\
   A_{column-normalized} &= A - [\,1\,1\cdots\,1]^T \vec{\beta}^T & (2)\\
   A_{both-normalized} &= A - \vec{\alpha}[\,1\,1\cdots\,1] \\
   &\quad - [\,1\,1\cdots\,1]^T \vec{\beta}^T & (3)
   \end{aligned}
   $$

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

2. **Constant-value matrix norm**

   Once preprocessing corresponding to the biologically justifiable coherence model is applied, the constant-value matrix norm becomes suitable to measure the proximity of the genes/conditions under that specific coherence assumption.

---

[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), applying optimal normalization (c.f. Eq. 4) to both rows and columns will result in Cheng and Church's clustering. The similar argument carries through to Boxes (2,3), (3,2), and (3,3). See Section 2.4.

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 that further elaborates the operations involved in the 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 (see [10]).

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	-17.5	-7.5	2.5	-3.7	-8	-11	-0.8	-0.8	-0.8
	-16.5	-5.5	4.5	-2.7	-6	-9	-0.6	-0.6	-0.6
	2.5	26.5	33.5	6.4	14	20	1.4	1.4	1.4
Normalization	-10	0	10	4	$-.5$	-3.5	0.8	-1.0	-0.7
	-10.6	0.3	10.3	3	$-.1$	-3.1	0.6	-0.3	-0.7
	-21	1	20	-7	0.5	6.5	-1.4	1.4	1.4
Normalization&Standardization	-1.2	0.0	1.2	$.02$	$-.03$	$.02$	1.3	-1.3	1.3
	-1.2	0.0	1.2	0.0	0.0	0.0	-0.3	0.4	-0.2
	-1.25	0.05	1.2	$.01$	$.03$	$-.01$	-1.0	1.0	-1.1

For example, based on Eq. 3, if optimal normalization is applied, this effectively leads to Cheng and Church's residue:

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

2.6. Visualization After Preprocessing

The preprocessing also provides a valuable vehicle for the visualization of gene expression data because the preprocessed vectors can be easily projected onto a low dimension subspace. In particular, we may consider the expression level of each gene across D conditions as D-dimensional vectors and then apply a supervised-learning Fisher Discriminant Analysis (FDA) to project the positive (e.g., ribosomal) and negative (e.g., non-ribosomal) gene vectors to a direction parallel to the vectors that connect the means of positive and negative gene vectors. More precisely, let us denote the positive and negative gene vectors as $\mathbf{x}_t^{(p)}$ and $\mathbf{x}_t^{(n)}$, respectively.[3] 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

$$\bar{\mathbf{x}}^{(p)} = \frac{1}{T_p} \sum_{t=1}^{T_p} \mathbf{x}_t^{(p)} \quad \text{and} \quad \bar{\mathbf{x}}^{(n)} = \frac{1}{T_n} \sum_{t=1}^{T_n} \mathbf{x}_t^{(n)}, \quad (5)$$

---

[3]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.

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

$$s(t) = \mathbf{y}_t^T \mathbf{w}, \quad (6)$$

where T represents transpose. As evident in Figure 1, vectorization allows the separation between positive and negative 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. The figure also suggests that univariate analysis will not yield as a good discrimination as multivariate analysis, pointing to the adoption of multi-metric fusion classifiers.

3. BICLUSTER EVALUATION BASED ON SINGLE COHERENCE MODEL

3.1. One-by-One Biclustering Scheme

We adopted a "one-by-one" supervised biclustering strategy proposed recently [10] for condition selection. Without loss of generality, we describe below the procedure for the ribosomal gene group in the yeast data set [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
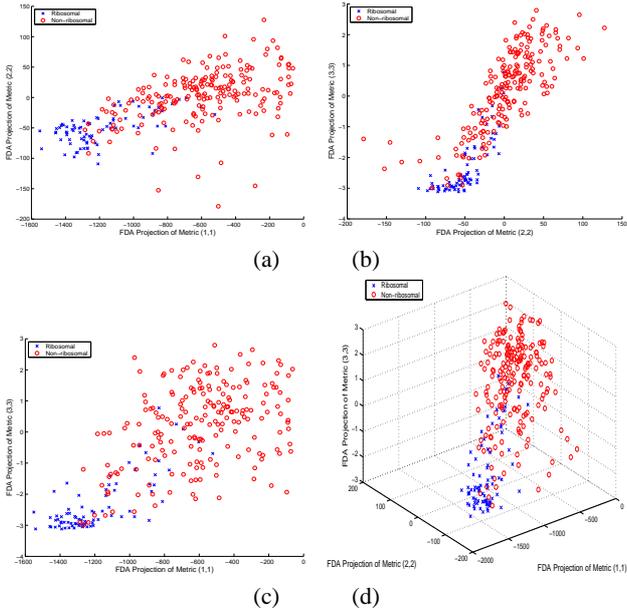
Fig. 1. Vectorization allows an FDA-like visualization of vector-norm distances between genes under various coherence models. Here blue "+" represent positive (ribosomal) genes while red circles represent negative (non-ribosomal) genes. (a) additive-coherent preprocessing vs. no-preprocessing; (b) additive-multiplicative-coherent preprocessing vs. additive-coherent preprocessing; (c) additive-multiplicative coherent preprocessing vs. no-preprocessing; and (d) combined view of using all the three preprocessing schemes.

of ribosomal genes can be found in [11].

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. 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 error residue. 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.

Once the conditions are selected, Eq. 6 can be used to project the test genes onto the FDA-subspace. A threshold ζ can then be chosen to classify a test gene t as either ribosomal or non-ribosomal:

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

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. A specific value of threshold ζ gives us one point on the ROC curve. To produce the entire spectrum of sensitivity-precision-specificity for the ROC, we may sweep the value of ζ from the minimum to the maximum value of the test scores $s(t)$, i.e., $\zeta = \min_t\{s(t)\}, \ldots, \max_t\{s(t)\}$.

3.2. Performance of Univariate Evaluation

Figure 2 shows the sensitivity against precision (ROC) of nine different preprocessing models 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.
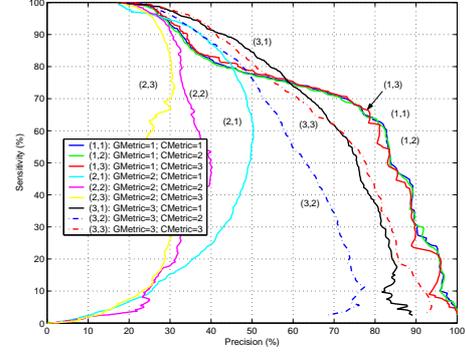

Fig. 2. Sensitivity against precision of nine different combinations of preprocessing processes for the conditions and genes. In the legend, "GMetric=m; CMetric=n" means that Metrics m and n were applied to the genes and conditions, respectively. In other words, it corresponds to Box (m,n) in Table 1.

4. FUSION OF MULTIPLE COHERENCE MODELS

There is plenty of biological evidence to support a theory that genes in the same group may be co-expressed under different models. For example, it is well-known that a gene may participate in multiple pathways that may be co-active under a subset of selected conditions. This calls for a fusion strategy that combines features produced by different preprocessing models to improve classification and prediction performance.
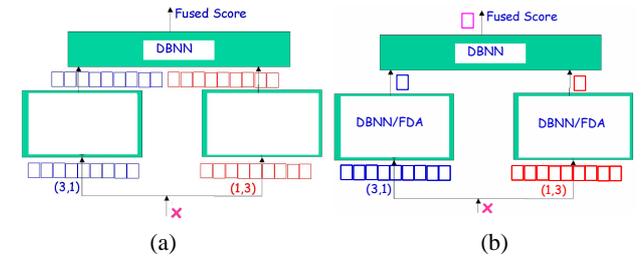

Fig. 3. Two possible Fusion Architecture. (a) Direct Fusion: multiple features are concatenated to form an expanded feature vector. (b) Indirect Fusion: each feature vector is compressed into a scalar (or low dimensional) feature or represented by a local score.

4.1. Fusion Classifiers

Figure 3 illustrates two possible fusion architectures. Figure 3 (a) shows a direct fusion scheme, where the original feature vectors 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) [12]).

Alternatively, an indirect fusion scheme is shown in Figure 3(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 processing model. Thereafter, the DBNN [12] can be used to fuse scores from various processing models to reach a final decision.

4.2. Adjustment of Decision Boundary for ROC Evaluation

Without loss of generality, let us consider the indirect fusion of two preprocessing models. Theoretically, a neutral decision boundary is the one that gives identical log-likelihood for positive and negative classes, More specifically, we have $\log p(\mathbf{s}(t)|\Lambda^{(p)}) = \log p(\mathbf{s}(t)|\Lambda^{(n)})$, where $\mathbf{s}(t) = [s_1(t) \quad s_2(t)]^T$ are 2-D vectors comprising of FDA-projected scores corresponding to the two preprocessing models (see Eq. 6) 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 classes, respectively. By counting the number of test vectors $\mathbf{s}(t)$ falling on the wrong side of the decision boundary, we can compute the precision, specificity, and sensitivity corresponding to a single point on the ROC curve. To produce the entire spectrum of sensitivity-precision-specificity, a disparity between the positive and negative log-likelihoods is introduced, i.e,

$$\log p(\mathbf{s}(t)|\Lambda^{(p)}) = \log p(\mathbf{s}(t)|\Lambda^{(n)}) + \alpha \qquad (8)$$

where $\alpha > 0$ (resp. $\alpha < 0$) if a higher specificity (resp. sensitivity) is desired. By adjusting α accordingly, the decision boundaries shown in Figure 4 can be obtained.

4.3. Which Models to Fuse

The selection criteria for the metrics adopted for the fusion network include (1) The preprocessing model must by itself (i.e., univariate) deliver a sound performance; and (2) the models to be selected for fusion must offer complementary information.

Let us now explore further on how to determine which models can offer more complementary information. The ROC curves as depicted in Figure 2 show the sensitivity against precision for the nine preprocessing models. Indeed, such ROCs offer an effective tool for selecting candidate models for fusion. For examples, if Model A (see e.g. 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 B has just the opposite performance (see e.g., Model (1,3)). In this case, the two ROC curves will have a crossover point. More precisely, these two models are truly complementary to each other and can serve as ideal fusion candidates.

4.4. Consistent Fusion

A modest fusion objective is to deliver a *consistent fusion* result, which is by definition at least as good as any of the single models in the entire specificity/precision region. Such a goal can be readily achieved by the following fusion scheme.

1. Determine the crossover point of curve A and curve B. Denote the sensitivity and precision at the crossover point as $s_{crossover}$ and $p_{crossover}$, respectively. For example, in

Figure 2, the crossover of Model (3,1) and Model (1,3) is at the point $s_{crossover} = 76\%$ and $p_{crossover} = 58\%$.

2. If we want to guarantee a sensitivity to remain higher than or equal to $s_{crossover} = 76\%$, then the decision boundary pertaining to Model A should be adopted, i.e., Model (3,1) of the left expert in Figure 3(b). On the other hand, if we want to be sure of a precision no poorer than $p_{crossover} = 58\%$, then we should adopt Model B, i.e., Model (1,3).

This scheme will yield basically a lower bound performance of any consistent fusion. Usually, a better than lower bound performance is expected if a soft fusion is adopted. This is discussed in the next section.

4.5. Performance of Multi-metric Fusion Schemes

Figure 4 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 [1]. It was found that a single elliptical cluster 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 sensitivity, precision, and specificity. Figure 5 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 preprocess models independently, which helps alleviate the difficulty encountered by feature fusion.

Figure 6 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 detection performance at the region of high precision. The Matlab programs that produce the results can be found in [13].

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-metric fusion classifier based on a DBNN mixture-of-experts architecture is proposed. The proposed preprocessing and fusion method has successfully yielded highly discriminant and accurate classification based on known ribosomal gene group.

6. REFERENCES

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

[2] D. J. Duggan, M. L. Bittner, Y. Chen, P. Meltzer, and J. M. Trent, "Expression profiling using cDNA microarrays," *Nature Genetics*, vol. 21, pp. 10–14, Jan. 1999.
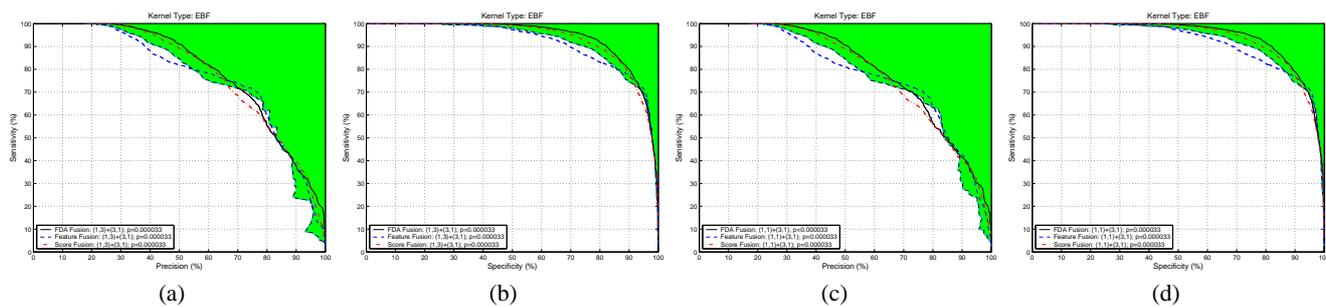
(a)　　　　　(b)　　　　　(c)　　　　　(d)

Fig. 5. The sensitivity-precision-specificity results of ribosomal detection supporting the idea of teaming up the constant-value metric with the additive-multiplicative-coherent metrics for fusion purposes. (a) and (b) fusion of Metrics (1,3) and (3,1). (c) and (d) fusion of Metrics (1,1) and (3,1). In the legend, (m, n) means that Metrics m and n were applied to genes and conditions, respectively. The green area represents the region of consistent performance.
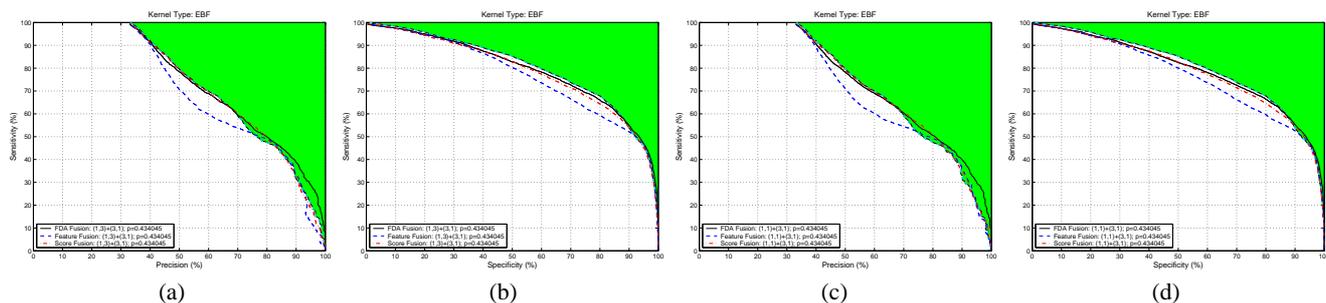

(a)　　　　　(b)　　　　　(c)　　　　　(d)

Fig. 6. The sensitivity-precision-specificity results of the molecular activity gene group. (a) and (b) fusion of Metrics (1,3) and (3,1). (c) and (d) fusion of Metrics (1,1) and (3,1). Refer to Figure 5 for the meaning of legends.
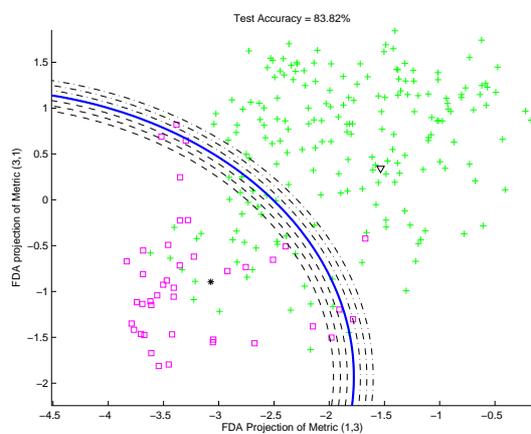

Fig. 4. Illustration of the use of 41 positive test patterns (ribosomal, represented by pink "□") and 200 test patterns (non-ribosomal, represented by green "+") from the yeast data set. The decision boundary is produced by a DBNN classifier trained by 80 positive training data and 200 negative training 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. 8.

[3] M. Bittner, et al., "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 3, pp. 536–540, Aug. 2000.

[4] D. Jiang, C. Tang, and A. Zhang, "Cluster analysis for gene expression data: A survey," *IEEE Transactions On Knowledge And Data Engineering*, vol. 16, no. 11, pp. 1370–1386, 2004.

[5] J.A. Hartigan, "Direct clustering of a data matrix," *J. Am. Statistical Assoc. (JASA)*, vol. 67, no. 337, pp. 123–129, 1972.

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

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

[8] L. Lazzeroni and A. B. Owen, "Plaid models for gene expression data," Tech. Rep., March, 2000, www-stat.stanford.edu/~owen/reports/plaid.pdf.

[9] Y. Cheng and G. M. Church, "Biclustering of expression data," in *Proc. Eighth Int'l Conf. Intelligent Systems for Molecular Biology (ISMB)*, 2000, vol. 8, pp. 93–103.

[10] 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.

[11] www.yeastgenome.org.

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

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