Hierarchical Multi-Bottleneck Classification Method And Its Application to DNA Microarray Expression Data

Xuejian Xiong[†] Wong Weng Fai[†] Hsu Wen-jing [§]

[†] Singapore-MIT Alliance
c/o National University of Singapore
3 Science Drive 2, Singapore 117543
(smaxx,wongwf)@nus.edu.sg

[§] Singapore-MIT Alliance
c/o School of Computer Engineering
Nanyang Technological University

Abstract

The recent development of DNA microarray technology is creating a wealth of gene expression data. Typically these datasets have high dimensionality and a lot of varieties. Analysis of DNA microarray expression data is a fast growing research area that interfaces various disciplines such as biology, biochemistry, computer science and statistics. It is concluded that clustering and classification techniques can be successfully employed to group genes based on the similarity of their expression patterns. In this paper, a hierarchical multi-bottleneck classification method is proposed, and it is applied to classify a publicly available gene microarray expression data of budding yeast Saccharomyces cerevisiae.

1. Introduction

DNA microarrays offer the ability to measure the levels of expression of thousands of genes simultaneously. These arrays consist of large numbers of specific oligonucleotides or DNA sequences, each corresponding to a different gene, affixed to a solid surface at very precise location. When an array chip is hybridized to labelled DNA derived from a sample, it yields simultaneous measurements of the mRNA levels in the sample for each gene represented on the chip. Since mRNA levels are expected to correlate roughly with the levels of their translation products, the active molecules of interest, array results can be used as a crude approximation to the protein content and thus the ‘state’ of the sample[10].

DNA microarrays provide a global view of gene expression which can be analyzed by a number of methods. For example, clustering can be performed to identify genes which are regulated in a similar manner under many different environmental conditions, and to predict the unknown functions of genes based upon the known functions of other gens in the same cluster[15, 10, 5, 23, 6]. Microarray data can also be used to infer regulatory pathways at the level of transcription. Toward that aim, Bayesian networks have recently been constructed to explain the probabilistic relationships among the expression of different genes[7]. DNA microarrays can also be used to characterize the cellular differences between different tissue types, such as between normal cells and cancer cells at different stages of tumor progression, or between cancers with different responses to treatment, or between control cells and cells treated with a particular drug. In this area, support vector machines (SVMs)[9] and Bayes techniques[2, 1, 10] have been used.

Classification on the basis of microarray data presents several algorithmic challenges. For example, the data often contain ‘technical’ noise that can be introduced at a number of different stages, such as production of the DNA microarray, preparation of the samples, and signal analysis and extraction of the hybridization results[10]. In addition, they also have ‘biological’ noise which come from non-uniform genetic backgrounds of the samples, or from the impurity or misclassification of samples. Furthermore, microarray expression data contain an overwhelming number of attributes relative to the number of training samples. The combined effect of large numbers of irrelevant genes could potentially drown out the contributions of the relevant ones[10]. To deal with this kind of machine learning problem, the use of SVMs has been suggested[14]. Brown *et. al.*[5, 4] used the SVMs method to analyze the microarray gene expression data of *Saccharomyces cerevisiae*. Lin *et. al.*[11] analyzed the ability of SVM to discriminate ribosomal protein coding genes from all other gens of known function based on their codon composition in *Saccharomyces cerevisiae*. Note that most of SVMs applications belongs to the binary classification problem.

In this paper, a multi-bottleneck concept is proposed for classifying multiple gene classes. Each available class, which has high dimensionality and/or small samples, is transferred into an auxiliary space first. In this new space, an information bottleneck represents the characteristics of the class in the original space. As an 'optimal' bottleneck of the class, it is independent to other bottlenecks corresponding to other classes. In other words, by finding the bottlenecks, overlapped gene classes can be separated well. Based on this concept, a hierarchical multi-bottleneck classification (HMBC) method is presented to find the multiple 'optimal' bottlenecks simultaneously.

The organization of this paper is as follows. The multi-bottleneck concept is proposed in section 2. In section 3, the way to find the 'optimal' multi-bottlenecks is presented, and the procedure of the HMBC method is shown in section 4. The classification of DNA microarray expression data of *Saccharomyces cerevisiae* is reported in section 5. Finally, in section 6, conclusions are given.

2. The Multi-Bottleneck Concept

Tishby *et. al* in [8, 18] first proposed the information bottleneck (IB) concept. The aim of the IB is to squeeze the information that a random variable $\mathcal{X}$ provides about another variable $\mathcal{Y}$ through a "bottleneck" formed by a limited set of codewords $\tilde{\mathcal{X}}$. Then the problem can be formalized as that of finding a short code for $\mathcal{X}$ that preserves the maximum information about $\mathcal{Y}$. Note that the idea behind the IB is to find the connection between $\mathcal{X}$ and $\mathcal{Y}$, which often cannot be obtained directly in an explicit way. Therefore, an auxiliary variable $\tilde{\mathcal{X}}$, the information bottleneck between $\mathcal{X}$ and $\mathcal{Y}$, is required. The relationships among $\mathcal{X}$, $\tilde{\mathcal{X}}$, and $\mathcal{Y}$ are shown in Figure 1. It can be seen that the introduction of the $\tilde{\mathcal{X}}$ can avoid the difficulty of finding the implicative relationship between $\mathcal{X}$ and $\mathcal{Y}$. The IB methods have been mainly applied in text and document classification[17, 3, 19, 16].

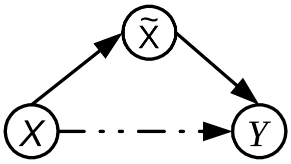

Figure 1: **The relationships among the variable $\mathcal{X}$, the information bottleneck $\tilde{\mathcal{X}}$, and the variable $\mathcal{Y}$.**

In the DNA microarray expression data classification, the relationships among different classes are very complex and implicative, and there are noise in each class. Inspired by the idea of IB, a multi-bottleneck concept is proposed where each class has its own bottleneck, i.e. there are multiple bottlenecks corresponding to multiple classes. Bottlenecks in the IB method are used to reflect the relationship between two relevant variables. In our proposed multi-bottleneck method, however, multiple bottlenecks are selected to be independent of each other. In other words, overlapped classes can be separated well via their bottlenecks in a auxiliary space. For example, two classes **L** and **K** have two bottlenecks $\mathcal{T}^l$ and $\mathcal{T}^k$, respectively. Their relationships are illustrated in Figure 2. It can be seen that there is no connection between two bottlenecks $\mathcal{T}^l$ and $\mathcal{T}^k$.

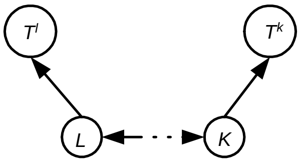

Figure 2: **The relationships among the class L, its information bottleneck $\mathcal{T}^l$, the class K, and its information bottleneck $\mathcal{T}^k$.**

Similar to the IB, the bottlenecks are the abstract representations of the respective classes. Therefore, the multi-bottleneck problem can be formalized as that of finding a representation of a class $\mathcal{X}$ that is totally different from the representation of the other class $\mathcal{Y}$. As a result, the mutual information between $\mathcal{T}^l$ and $\mathcal{T}^k$ should be minimal, while the mutual information between **L** and $\mathcal{T}^l$ and that between **K** and $\mathcal{T}^k$ should be maximal. The objective function for selecting optimal bottlenecks, $\mathcal{T}^l$ and $\mathcal{T}^k$, is defined as,

$$\mathcal{J}[\mathcal{T}^l;\mathcal{T}^k] = I(\mathcal{T}^l;\mathcal{T}^k) - \beta_1 I(\mathbf{L};\mathcal{T}^l) - \beta_2 I(\mathbf{K};\mathcal{T}^k) \qquad (1)$$

where β_1 and β_2 are two Lagrange multipliers.

Because a bottleneck is the abstract representation of its corresponding class, a supervised and parallel scheme is adopt to generate it. For a class with some available sample data, or samples in short, the bottleneck can by obtained by studying samples using machine learning techniques. Multiple bottlenecks are generated individually and parallelly, and they can be seen as the 'suboptimal'/initial bottlenecks. This is because the overlap among classes are not considered during this generation of bottlenecks.

Here, the unsupervised fuzzy clustering (UFC) algorithm[20, 21] is applied to classes to extract their bottlenecks, respectively. Because there is no prior knowledge on the inherent structure/distribution of any class in gene expression data, other well-known clustering algorithms, for example, K-means or fuzzy c-means methods, are not suitable. The UFC can automatically provide not only optimal clusters, but also the optimal number of clusters in each class. As a result, each bottleneck consists of a set of fuzzy clusters. Note that the number of clusters in different classes can be different due to the independent generation scheme.

Because the UFC is an optimal clustering algorithm, the last two terms in equation (1), $I(\mathbf{L};\mathcal{T}^l)$ and $I(\mathbf{K};\mathcal{T}^k)$, are known and thus constant. Therefore, the objective function of the multi-bottleneck method is rewritten as,

$$\mathcal{J}[\mathcal{T}^l;\mathcal{T}^k] = I(\mathcal{T}^l;\mathcal{T}^k) \tag{2}$$

By equation (2), it can be seen that the connections among classes are involved. The optimal bottlenecks can be obtained by minimizing this function based on the suboptimal ones.

3. The Semi-Parametric Mixture Identification for Bottlenecks

To obtain the 'optimal' bottlenecks, each bottleneck is represented by a mixture. The components of the mixture are corresponding to the clusters in the class. The number of clusters in the class and the parameters of each cluster, such as cluster centers, size, dispersion, etc., are available by using the UFC. Therefore, the parameters of the components, for example, centers, variance, etc., are known. Subsequently, the coefficients of the components of the mixture are set as the weights of the corresponding clusters. Hence, in the following, the term of "coefficient" will be replaced with "weight".

For a mixture, each component of the mixture can be described by a basis kernel function, or kernel in short. The kernel corresponds to a cluster in the class. Let $\{\mathbf{P}^l_i, i = 1,2,\cdots,c^l\}$ be a set of c^l clusters in the class $\mathbf{L}$, and samples $x \in \mathcal{X}$. The prior probability density distribution (pdf) of the bottleneck $\mathcal{T}^l$, i.e. $\mathbf{p}(x|\mathcal{T}^l)$, can be represented as a mixture with a set of kernels. The basis kernel function, $\phi^l(x|i)$, is the distribution function of the cluster $\mathbf{P}^l_i$ in the class $\mathbf{L}$. Therefore,

$$\mathbf{p}(x|\mathcal{T}^l) = \sum_{i=1}^{c^l} w^l_i \phi^l(x|i), \tag{3}$$

where w^l_i is the weight of the i-th kernel in the bottleneck $\mathbf{T}^l$. There is the requirement as follows,

$$\begin{aligned} w^l_i &\in [0,1] \qquad \forall \quad l,i \\ \sum_i w^l_i &= 1 \qquad \forall \quad l \end{aligned} \tag{4}$$

w^l_i can be reasonably fixed as the ratio of the dispersion of $\mathbf{P}^l_i$ over the class $\mathbf{L}$.

Therefore, the bottleneck-conditional prior distributions of all C classes can be written as a linear combination of the kernels with the weight matrix, i.e.

$$\mathbf{P} = \mathbf{W\Phi} \tag{5}$$

where $\mathbf{P} = \{\mathbf{p}(x|\mathcal{T}^l)\}$ is the set of class prior probabilities, $\mathbf{\Phi} = \{\phi^l(x|i)\}$ is the set of basis kernels, and $\mathbf{W} = \{w^l_i\}$ is a weight matrix. $\mathbf{W}$ is not necessary to be square because the numbers of clusters c^l in different classes are different. In addition, the basis kernels in a mixture can be in different forms, such as, Gaussian, Detric, etc. The kernels in different mixtures can also be in different forms.

Due to the parallel generation of initial bottlenecks, the relationships among classes are ignored. Obviously, the overlaps among classes make the initial bottlenecks are not optimal. The optimal ones can separate the corresponding classes completely. To reduce the effect of the overlap between two classes, the weights of clusters in overlapped area should be minimized. This problem can be dealt with by finding the optimal weights for the mixtures. In other words, this is a semi-parametric mixture identification problem. Most of the existing mixture identification methods, for example, the expectation maximization (EM) algorithm or the reversible jump Markov Chain Monte Carlo (RJMCMC) method[13], aim to find the optimal partitions and to estimate the parameters of mixtures simultaneously. They are an overkill for solving the overlap problem here. This is because only the weights of the mixtures are needed to be identified in this semi-parametric mixture identification problem.

The objective function in equation (2) can then be rewritten as,

$$\mathcal{J}[\mathcal{T}^l;\mathcal{T}^k] = I(\mathbf{p}(x|\mathcal{T}^l);\mathbf{p}(x|\mathcal{T}^k)) \tag{6}$$

where both $\mathbf{p}(x|\mathcal{T}^l)$ and $\mathbf{p}(x|\mathcal{T}^k)$ are semi-parametric mixtures. Note that this equation is only for two bottlenecks. To consider the total overlaps among all classes in $\mathcal{C}$, the objective function is defined as:

$$\mathcal{J} \triangleq \sum_{l=1}^{C} P(\mathbf{L}) \left(I[\mathbf{p}(x|\mathcal{T}^l);\mathbf{p}(x)] \right) \tag{7}$$

It is shown that $\mathcal{J}$ is the mutual information between $\mathcal{X}$ and $\mathcal{C}$, $I(\mathcal{X};\mathcal{C})$. We therefore seek the bottlenecks for which $\mathcal{J}$ is minimal.

Because each bottleneck is formed by a semi-parametric mixture in equation (3), whose kernels are fixed but whose weight matrix, $\mathbf{W}$, is adaptive so as to minimize the overall mutual information of classes at the bottleneck level. Therefore, the objective function $\mathcal{J}$ should be minimized, i.e.

$$\min_{\{w_i^l\}} \{\mathcal{J}(w_i^l)\} \tag{8}$$

subject to the constraints in equation (4). The gradient of the equation (8) with respect to each weight, w_i^l, can be obtained straightforwardly. Then, the optimization of $\{w_i^l\}$ for each mixture can be realized via any optimization method, such as, steepest descent method, Broyden-Fletcher-Goldfarb-Shanno (BFGS) variable metric method[12], etc.

4. The Procedure of the HMBC Method

The hierarchical multi-bottleneck classification (HMBC) method is proposed based on above discussions. There are three stages in the HMBC.

- In the first stage, the initial bottlenecks are generated from corresponding classes by using the UFC via a parallel way. Thus, each bottleneck is represented by a semi-parametric mixture.
- In the second stage, the minimum-mutual-information approach is used to obtain the optimal weight matrix, $\mathbf{W} = \{w_i^l\}$, of bottlenecks.
- In the third stage, the "optimal" bottleneck of the class is determined. The bottleneck still consists of a set of clusters, with a set of corresponding optimal weights.

Figure 3 illustrates the procedure of using the HMBC method to two classes, **L** and **K**.

For categorizing an unknown data to a class, a new distance, from the image-class matching distance[22, 20], is used.

$$D(x_q, \mathbf{L}) = \frac{\sum_{j=1}^{c^l} \varpi_{qj} dist(\mathbf{x}_q, \mathbf{P}_j^l)}{\sum_{j=1}^{c^l} \varpi_{qj}} \tag{9}$$

where, $dist(\mathbf{x}_q, \mathbf{P}_j^l)$ stands for the ground distance between the gene vector $\mathbf{x}_q$ and the cluster $\mathbf{P}_j^l$ in the class **L**. Normally, the Euclidian distance is adopt to be the ground distance. ϖ_{qj} is the another optimal weight on $dist(\mathbf{f}_q, \mathbf{P}_j^l)$ that minimizes the $D(x_q, \mathbf{L})$ subject to the following constraints:

$$\begin{aligned} \varpi_{qj} &\geq 0 \\ \sum_{j=1}^{c^l} \varpi_{qj} \leq 1; &\quad \varpi_{qj} \leq w_j^l \\ \sum_{j=1}^{c^l} \varpi_{qj} &= \min(1, \sum_{j=1}^{c^l} w_j^l) \end{aligned} \tag{10}$$

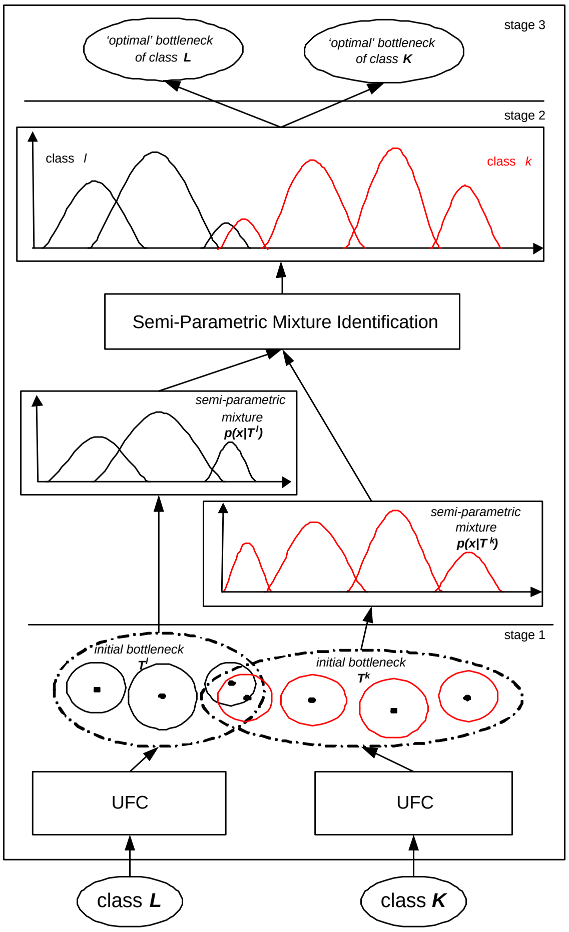
stage 3
'optimal' bottleneck of class L
'optimal' bottleneck of class K
stage 2
class l
class k
Semi-Parametric Mixture Identification
semi-parametric mixture p(x|T^l)
semi-parametric mixture p(x|T^k)
stage 1
initial bottleneck T^l
initial bottleneck T^k
UFC
UFC
class L
class K

Figure 3: **Illustration of the procedure of the HMBC method to two classes L and K.**

where, w_j^l is the obtained optimal weight of the cluster $\mathbf{P}_j^l$ in the class **L**.

5. The Classification of DNA Microarray Expression Data of *Saccharomyces Cerevisiae*

Saccharomyces cerevisiae is an especially favorable organism in which to conduct a systematic investigation of gene expression. The genes are easy to recognize in the genome sequence. The regulatory elements are generally compact and close to the transcription units, and much is already known about its genetic regulatory mechanisms. The expression data of *Saccharomyces cerevisiae* used in this paper comes from [5, 4], which are also available at the web site http://www.cse.ucsc.edu/research/compbio/genex/expressdata.html. The data consist of 2467 annotated yeast genes with 80 elements, which were collected at various time points during the mitotic cell division cycle, sporulation, temperature and reducing shocks, and the diauxic shift.

In our experiment, the DNA microarray expression data of *Saccharomyces cerevisiae* needs to be preprocessed. Let $X = \{x_{ki}, k = 1, \cdots, n\} \in \mathcal{R}^m$ be the input data, while $\tilde{X} = \{\tilde{x}_{ki}, k = 1, \cdots, n, i = 1, \cdots, \tilde{m}\}$ is the original expression data. Because many genes do not have the last 3 elements, only 77 elements are selected for each gene. Therefore, $m = \tilde{m} - 3 = 77$. After then, each expression vector is normalized by the following equation,

$$x_{ki} = \frac{\log \tilde{x}_{ki}}{\sqrt{\sum_{j=1}^{m} (\log \tilde{x}_{kj})^2}}, \quad k = 1, \cdots, n. \tag{11}$$

As a result, initial analysis described here are carried out by using a set of 77-element expression vectors for 2467 yeast genes, i.e. $X = \{x_{ki}, k = 1, \cdots, 2467\} \in \mathcal{R}^{77}$.

For training the original bottlenecks, we use the class definitions made by the MIPS Yeast Genome Database (MYGD). There are six functional classes: tricarboxylic acid cycle (TCA), respiration, cytoplasmic ribosomes, proteasome, histones and helix-turn-helix proteins. The MYGD class definitions come from biochemical and genetic studies of gene function, while the microarray expression data measures mRNA levels of genes. Many classes in MYGD, especially structural classes such as protein kinases, will be unlearnable from expression data by any classifier. The first five classes were selected because they represent categories of genes that are expected, on biological grounds, to exhibit similar expression profiles. Furthermore, Eisen et al. suggested that the mRNA expression vectors for these classes cluster well using hierarchical clustering. The sixth class, the helix-turn-helix proteins, is included as a control group. Since there is no reason to believe that the members of this class are similarly regulated, we did not expect any classifier to learn to recognize members of this class based upon mRNA expression measurements[5, 4].

For using the HMBC method, sample data of each available class should be provided. Based on the class definition of MYGD, there are 17, 30, 121, 35, 11, and 16 samples selected for the six classes, respectively. By learning the samples of each class individually, six initial bottlenecks are generated. Each bottleneck consists of a set of clusters. The numbers of clusters in six classes are 10, 6, 4, 8, 10, and 10, respectively.

After this first step, a semi-parametric mixture of each bottleneck is constructed. The form of the basis kernel function is fixed as Gaussian here. The steepest descent method is used to solve the optimization problem in equation (8). The initial values of weights $\{w_i^l\}$ is set as,

$$w_i^l = \frac{dp_i^l}{dp^l} \tag{12}$$

where, dp_j^l is the dispersion of the cluster $\mathbf{P}_j^l$ in the class $\mathbf{L}$, and dp^l is the dispersion of the class $\mathbf{L}$. Its prior probability, $P(\mathbf{L})$, is defined as,

$$P(\mathbf{L}) = \frac{n^l}{N} \tag{13}$$

where, N is the number of data in the input data set $\mathbf{X}$, and n^l denotes the number of data in the class $\mathbf{L}$.

Thus, by minimizing the objective function in equation (7), optimal weights in each bottleneck are obtained. Based on the obtained 'optimal' bottlenecks and the distance defined in equation (9), the unknown function gene can be predicted.

As reported in [5, 4], there are 25 genes for which the developed SVMs consistently disagree with the MYGD classification. In table 1, the classification results, by using the HMBC and SVMs methods, of these 25 genes are listed. The FP stands for false positive which occurs when the machine learning techniques include the gene in the given class but the MYGD classification does not. A false negative (FN) occurs when the machine learning techniques do not include the gene in the given class but the MYGD classification does. The positive (P) and negative (N) refer to the agreement of the classification results between MYGD and HMBC. It can be seen that 14 of these 25 genes are classified into suitable classes by using the HMBC method, while 11 genes are still misclassified based on the class definition of MYGD. Many of the disagreements reflect the different perspectives, provided by the expression data concerning the relationships between genes, of the machine learning techniques and MYGC. The different functional classification can illustrate the new information that expression data brings to biology[4].

6. Conclusion

In this paper, the multi-bottleneck concept is proposed. Subsequently, the hierarchical multi-bottleneck classification (HMBC) method is proposed and applied for classification of DNA microarray expression data of *Saccharomyces cerevisiae*. The characteristics of six functional classes, defined by MYGD, are studied in a parallel and supervised fashion, and they are represented by information bottlenecks. Through the bottlenecks, classes can be discriminated well. In this paper, the analysis of the experiment results is very simple and initial. In the future, more sophisticated experiments will be presented. It can be expected that the HMBC method is used to analyze other gene features, such as the presence of transcription factor binding sites in

Table 1: **Classification results by using the HMBC method for 25 consistently misclassified genes in [5, 4].**

Class	Genes	SVMs	HMBC
TCA	YPR001W	FN	P
	YOR142W	FN	FN
	YNR001C	FN	P
	YLR174W	FN	P
	YIL125W	FN	P
	YDR148C	FN	P
	YDL066W	FN	FN
	YBL015W	FP	FP
Resp	YPR191W	FN	P
	YPL271W	FN	P
	YDL067X	FN	P
	YPL262W	FP	FP
	YML120C	FP	FP
	YKL085W	FP	N
Ribo	YLR406C	FN	P
	YPL037C	FP	FP
	YLR075W	FP	FP
	YAL003W	FP	FP
Prot	YHR027C	FN	P
	YGR270W	FN	P
	YDR069C	FN	FN
	YDL020C	FN	P
	YGR048W	FP	FP
Hist	YOL012C	FN	P
	YKL049C	FN	FN

the promoter region or sequence features of the translated protein.

References

[1] P. Baldi. On the convergence of a clustering algorithm for protein-coding regions in microbial genomes. *Bioinformatics*, 4(16):367–371, 2000.

[2] P. Baldi and A. D. Long. A bayesian framework for the analysis of microarray expression data: Regularized t-test and statistical inferences of gene changes. *Bioinformatics*, 2001.

[3] R. Bekkerman, R. El-Yaniv, N. Tishby, and Y. Winter. On feature distributional clustering for text categorization. In *Proceedings of the SIGIR-2001*, January 2001.

[4] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. Sugnet, M. Ares, and J. D. Haussler. Support vector machine classification of microarray gene expression data. Technical Report Technical report UCSC-CRL-99-09, University of California, Santa Cruz, 1999.

[5] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. Sugnet, T. S. Furey, M. Ares, and J. D. Haussler. Knowledge-based analysis of microarray gene expression data using support vector machines. In *Proceedings of the National Academy of Sciences*, volume 97, pages 262–267, 2000.

[6] M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. In *Proceedings of the National Academy of Sciences USA*, volume 95, pages 14863–14868, December 1998.

[7] N. Friedman, M. Linial, I. Nachman, and D. Pe'er. Using Bayesian networks to analyze expression data. In *Proceedings of the 4th Annual International Conference on Computational Molecular Biology (RECOMB)*, pages 127–135, Tokyo, Japan, 2000. Universal Academy Press.

[8] N. Friedman, O. Mosenzon, N. Slonim, and N. Tishby. Multivariate information bottleneck. In *Proceedings of the UAI-2001*, March 2001.

[9] T. Golub, D. Solnim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfileld, and E. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[10] A. D. Keller, M. Schummer, L. Hood, and W. L. Ruzzo. Bayesian classification of dna array expression data. Technical Report Technical report UW-CSE-2000-08-01, University of Washington, Seattle, 2000.

[11] K. Lin, Y. Kuang, J. Joseph, and P. Kolatkar. Conserved codon composition of ribosomal protein coding genes in escherichia coli, mycobacterium tuberculosis and saccharomyces cerevisiae: lessons from supervised machine learning in functional genomics. *Nucleic Acids Research*, 30(11):2599–2607, 2002.

[12] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, 1991.

[13] S. J. Roberts, C. Holmes, and D. Denison. Minimum-entropy data partitioning using reversible jump Markov Chain Monte Carlo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(8), 2001.

[14] B. Schlkopf, I. Guyon, and J. Weston. Statistical learning and kernel methods in bioinformatics. Technical report, 2002.

[15] J. Sinkkonen and S. Kaski. Clustering based on conditional distributions in an auxiliary space. *Neural Computtation*, 14:217–239, 2001.

[16] N. Slonim, R. Somerville, N. Tishby, and O. Lahav. Objective classification of galaxy spectra using the information bottleneck method. *Monthly Notes of the Royal Astronomical Society*, 323:270–284, April 2000.

[17] N. Slonim and N. Tishby. The power of word clustering for text classification. In *Proceedings of the European Colloquium on IR Research, ECIR 2001*, January 2001.

[18] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proceedings of the The 37th annual Allerton Conference on Communication, Control, and Computing*, September 1999.

[19] N. Tishby and N. Slonim. Data clustering by Markovian relaxation and the information bottleneck method. In *Proceedings of the NIPS-2000*, May 2000.

[20] X. Xiong. *Image Classification for Image Database Organization in a Content-Based Image Retrieval System*. PhD thesis, Nanyang Technological University, 2002.

[21] X. Xiong and K. L. Chan. Towards an unsupervised optimal fuzzy clustering algorithm for image database organization. In *Proceedings of the 15th International Conference on Pattern Recognition*, pages 73–77, Barcelona, Spain, August 2000.

[22] X. Xiong, K. L. Chan, and L. Wang. An image database semantically structured based on automatic image annotation for content-based image retrieval. In *Proceedings of the 5th Asian Conference on Computer Vision (ACCV'02)*, Melbourne, Australia, January 2002.

[23] M. Zhang. Large-scale gene expression data analysis: a new challenge to computational biologists. *Genome Res.*, 9:681–688, 1999.