A Bayesian classification method for treatments using microarray gene expression data

Yuan Ji, Kam-Wah Tsui, and KyungMann Kim

An important application of microarray gene expression data is classification of treatments. One pioneering work in this area is by Golub et al. (1999) who classified samples into three types of leukemia using oligonucleotide microarrays. Recently, with more applications appearing in the literature, researchers began to realize the importance of constructing accurate and efficient classification methods in order to obtain desirable classification results. Dudoit et al. (2002) provided comprehensive comparisons of several discriminant methods, and Lee and Lee (2002) developed multicategory support vector machines and applied them to classification of cancer data. In this article, we propose an empirical Bayesian classification method that accommodates data structures emerging from microarray experiments such as correlations among gene expressions and high dimensionality with more genes (as predictor variables) than treatments (as data points). Specifically, we build a latent cluster structure for genes into a Bayesian classification process to achieve two goals: to account for the correlation among gene expressions and to reduce the number of nuisance parameters in the Bayesian model. Our simulation studies show that the proposed method generally yields fewer classification errors than the naive Bayesian method which ignores the latent cluster structure of genes by assuming homogeneity among them. We apply our method to two data sets from the literature and demonstrate that desirable results are achieved.

KEY WORDS: Bayes rule, empirical Bayesian method, method of moment, model-based cluster.

---

1Yuan Ji is Ph.D. candidate, Department of Statistics, University of Wisconsin - Madison, Madison, WI 53706 (E-mail: yuanj@stat.wisc.edu). Kam-Wah Tsui is Professor, Department of Statistics, University of Wisconsin - Madison, Madison, WI 53706 (E-mail: kwtsui@stat.wisc.edu). KyungMann Kim is Professor, Department of Biostatistics and Medical Informatics, University of Wisconsin - Madison, Madison, WI 53792 (E-mail: kmkim@biostat.wisc.edu). The authors thank Chris Bradfield and the Bradfield Lab for providing a microarray gene expression data set which is used in one of the examples in this paper. This research was partially supported by the NCI grant CA52733 and the Merck Foundation fellowship.
1 Introduction

With the recent advance in microarray technology, researchers in various areas are beginning to investigate and answer important scientific questions that could not be addressed before. Among them, classification of treatments using gene expression profiles has received much attention. Here, we use the generic term “treatment” to denote target objects that need to be classified. In medical research, classifying cancer types is crucial to diagnosis, but standard classification methods rely on clinical variables that provide limited and unreliable information about the disease. In toxicological and environmental health studies, an initial screen of toxic chemicals is essential to reducing the total cost of the analysis.

Microarray technology provides researchers with a powerful tool to solve these and similar problems. Using gene expression levels to classify disease types results in finer and more reliable classification. Success in classifying toxic chemicals would greatly speed the entire process of screening and testing, thus improving the environmental condition by reducing the exposure to hazardous chemicals. Golub et al. (1999) and Thomas et al. (2001) have initiated pioneering work after recognizing the importance of classification with microarray data. The former carried out classification of three types of leukemia with oligonucleotide microarrays and the latter classified twenty four toxicants into five toxicological categories using complementary DNA (cDNA) microarrays. As more applications appear in the literature, focus is being directed to constructing accurate and efficient classification methods. Dudoit et al. (2002) provided a comprehensive review of several discriminant methods, and Lee and Lee (2002) extended the classical binary
support vector machine to a multicategorical version. In this article, we consider Bayesian classification with microarray data. Although Bayesian modeling for microarray gene expression data has been investigated by authors including Efron et al. (2001), Newton et al. (2001), and Ibrahim et al. (2002), they were focusing on identifying differentially expressed genes instead of classification for treatments.

Bayesian classification methods, formalized by Ferguson (1967) and extended by Young (1995), are known for their flexibility and accuracy. However, with the special features in microarray gene expression data such as correlation among gene expressions and extremely high dimensionality with many more genes than treatments, the traditional Bayesian classification methods do not appear to work properly. Kontkanen et al. (1998) proposed a naive Bayes approach which was adopted by Thomas et al. (2001) to classify toxicants. Their method requires a discretization of gene expression levels which results in obvious loss of information. In addition, the method is sensitive to the set of genes selected for classification. As it is well known, identifying informative genes is itself a challenging task.

In this article, we propose an empirical Bayesian classification method specifically designed for microarray gene expression data. We construct a latent cluster structure for genes to accommodate the special features in the data. We then incorporate the information contained in the cluster structure into the Bayesian classification process. In section 2, we describe how to construct a latent cluster structure for genes using the model-based cluster method. We introduce the Bayesian classification method that utilizes clustering results in Section 3. In Section 4, we describe simulation studies and summarize the results, and in Section 5, we evaluate the performance of the proposed
method with two examples using real data. Finally, we summarize our findings and conclusions in Section 6.

2 Model-based cluster analysis

Cluster analysis has been used by researchers as a tool to reveal structural or functional patterns in the microarray gene expression data. For example, Eisen et al. (1998) proposed hierarchical clustering and Tamayo et al. (1999) introduced self organizing maps, both of which are based on nonparametric methods. Parametric methods have been investigated by authors such as Fraley and Raftrey (1999) and Yeung et al. (2001). When properly employed, the resulting gene cluster structure will reflect, to some extent, the true underlying biological relationship among genes. To obtain such a cluster structure for genes, we apply the model-based cluster method by Fraley and Raftery (1999).

Let $x = (x_1, ..., x_n)$ denote a vector of $n$ observations where each observation $x_g$ is a $m \times 1$ vector, $g = 1, ..., n$. Assume that each $x_g$ is generated by a finite mixture of underlying probability distributions where each component in the mixture is viewed as a distinct cluster. Denote the density of the $l$th component by

$$f_l(x_g|\mu_l, \Sigma_l) = \frac{\exp\left\{-\frac{1}{2}(x_g - \mu_l)^T \Sigma_l^{-1}(x_g - \mu_l)\right\}}{(2\pi)^{m/2} |\Sigma_l|^{1/2}}, \quad l = 1, ..., L.$$  

Thus, $x_g$ follows a $m$-dimensional multivariate normal distribution with mean vector $\mu_l$ and covariance matrix $\Sigma_l$ when it belongs to the $l$th component. Let $\theta_l$ denote $(\mu_l, \Sigma_l)$ and let $\pi_l$ be the probability that an observation belongs to the $l$th component. Thus, with a total of $L$ components, the mixture likelihood function of $(\theta_1, ..., \theta_L; \pi_1, ..., \pi_L)$ is
of the form

\[ L_M(\theta_1, \ldots, \theta_L; \pi_1, \ldots, \pi_L|x) = \prod_{g=1}^{n} \sum_{l=1}^{L} \pi_l f_l(x_g|\theta_l). \]  

(1)

The EM algorithm by Dempster et al. (1977) can be used to maximize the likelihood function (1). We determine the number of clusters objectively using the Bayesian information criteria (BIC) by Schwarz (1978) by choosing the number of clusters with the largest BIC value. Details of how to apply the model-based cluster method with EM to gene expression data are given in Appendix A.

With the microarray gene expression data, each \( x_g \) refers to a vector of expression levels of gene \( g \). We recommend using the gene expression levels in log scale for cluster analysis as gene expression levels are always positive, thus violating the normality assumption. Furthermore, since different treatments may affect expression levels of each gene in different ways, it is important to select a set of gene expression levels with homogeneity for success of the cluster analysis. Hence, gene expression levels under control conditions are a natural choice for the cluster analysis. Alternatively, gene expression levels under the same treatment or the same class of treatments are another choice because they are presumably affected in the same way.

The model-based cluster analysis described above assigns cluster labels to all the genes in the data, resulting in the special data structure shown in Table 1. We name it the latent cluster structure since it is not directly observed. In Table 1, the data contains \( m \) treatments and \( n \) genes with a known class label for each treatment. Such a set of data is usually called the training sample. The goal is to construct classification rules based on the gene expression levels from the training sample to predict the class of new treatments.
later. In Table 1, $C_t \in \{1, ..., K\}$ denotes the class label of treatment $t$, $t = 1, ..., m$, where $K$ is the total number of classes; $x_{gt}$ denotes the expression level of gene $g$, $g = 1, ..., n$, measured under treatment $t$. The $n$ genes are grouped into $L$ clusters as a result of the model-based cluster analysis. So the original $n$ genes $\{1, ..., n\}$ are rearranged and relabeled as $\{1, ..., n_1, n_1 + 1, ..., n_2, ..., n_{L-1} + 1, ..., n_L\}$ with genes $\{n_{l-1} + 1, ..., n_l\}$ in the cluster $l$, $l = 1, ..., L$, and with $n_0 = 0$ and $n_L = n$.

We observe two distinct features in Table 1: first, the genes are clustered together; second, the number of genes is much larger than the number of treatments, or using our notation, $n \gg m$. The second feature is sometimes denoted by other authors as “$p \gg n$”, with $p$ and $n$ being the generic notations for the number of predictors and observations, respectively. While the first clustering feature implies underlying relationships among the genes in the same cluster, which must be accounted for, the second feature $n \gg m$ raises a challenging modeling question: How do we construct a model for such data? We answer this question in the next section.

3 Empirical Bayes classification

3.1 The Bayes rule

The Bayes rule for the multiple class classification problem has been obtained by, for example, Ferguson (1967). We state the basic result below.

Result 1 Assume that the true class $\delta$ is in one of the $K$ classes labeled as $\Delta = \{1, ..., K\}$. Let $\Pi$ be the prior distribution which assigns probability $\pi_j$ if $\delta = j$, for $j \in \Delta$. Suppose that the observation $x$, which could be either a vector or scalar, follows the density
given \( \delta = j \). Then under the 0-1 loss function, which equals 0 if the decision rule agrees with the true class and 1 otherwise, the Bayes rule with respect to \( \Pi \) for a given \( x \) is to choose class \( j^* \) where \( j^* \) is such that

\[
\pi_{j^*} f_{j^*}(x) = \max_{j \in \Delta} \pi_j f_j(x).
\]

With the microarray gene expression data, \( x \) refers to the set of gene expression levels observed and \( \delta \) refers to the class label of the treatment to be classified. In order to employ the Bayes rule, we need to construct the model with appropriate density functions \( f_j \) and prior distributions.

### 3.2 Model building

Our model is built upon the latent cluster structure described in Table 1. In addition to the notation introduced, we let \( k_i \) denote the number of treatments in class \( i \), \( i = 1, ..., K \), thus \( \sum_{i=1}^{K} k_i = m \).

We assume that gene expression levels are measured under both treatment conditions and control conditions. Let \( X_{g(l)t} \) denote the expression level of gene \( g \) in cluster \( l \) measured under treatment \( t \). Let \( X_{g(l)c} \) denote the expression level of gene \( g \) in cluster \( l \) under the corresponding control condition. In dual channel microarrays which most cDNA arrays are, \( X_{g(l)t} \) and \( X_{g(l)c} \) are the fluorescence intensities for each gene. For example, if the treatment is dyed with green and the control with red, \( X_{g(l)t} \) and \( X_{g(l)c} \) are the average green and red intensities across all the spots with gene \( g \) printed. In single channel microarrays such as oligonucleotide arrays, \( X_{g(l)t} \) and \( X_{g(l)c} \) are the average ratios of normalized gene expression levels between the perfect match and the mismatch, under the treatment condition and the control condition, respectively. We propose the following
log-normal distributions: For gene $g = 1, ..., n_L$, cluster $l = 1, ..., L$, treatment $t = 1, ..., m$, and class $i = 1, ..., K$,

$$X_{g(l)}|\{\mu_l, \sigma^2_l, \Delta_{gi}, C_l = i\} \sim \log N(\mu_l + \Delta_{gi}, \sigma^2_l),$$

$$X_{g(l)c}|\{\mu_l, \sigma^2_l\} \sim \log N(\mu_l, \sigma^2_l),$$

where $\mu_l$ and $\sigma^2_l$ are the mean and variance of expression levels of the genes in cluster $l$ and $\Delta_{gi}$ is the differential effect of treatments in class $i$ on gene $g$.

According to the model given by (2) and (3), the distribution of gene expression level $X_{g(l)c}$ under the control condition is primarily determined by the latent cluster $l$ to which gene $g$ belongs; the mean $\mu_l$ and variance $\sigma^2_l$ of $X_{g(l)c}$ only depend on cluster label $l$. This is consistent with the assumptions in the model-based cluster analysis. Recall that in the model-based clustering, gene expression levels in log scale under control conditions are assumed to be generated by a mixture of normal distributions with each component of the mixture being a distinct cluster. So for genes in the same cluster, their expression levels under control conditions follow the same distribution, which agrees with our model.

According to (2), the variance of gene expression levels $X_{g(l)t}$ under treatment $t$ is $\sigma^2_l$, the same for all genes in the same cluster $l$ across treatments. The effect of treatment $t$ on the expression level of gene $g$ is through the mean of $X_{g(l)t}$ by adding a quantity $\Delta_{gi}$ to it, where $i$ denotes the class to which treatment $t$ belongs.

Our model is general in the sense that it covers a range of models that have been used in the literature. For example, if all the genes belong to one cluster, i.e., $L = 1$, we have a homogeneous model. If each gene forms a distinct cluster, then we obtain a completely heterogeneous model in which case $L = n$ and each gene is described by a
different variance $\sigma_g^2$. In between these two extreme cases, intermediate models can be obtained where $1 \ll L \ll n$, and thus problems with either homogeneous or completely heterogeneous models can be avoided. Homogeneous models are unrealistic and lack flexibility, while completely heterogeneous models are often computationally intractable.

According to the model given by (2) and (3), the difference of two $X$’s in log scale

$$D_{gl} = \log X_{g(l)t} - \log X_{g(l)c}$$

follows

$$D_{gl}|\{\sigma_l^2, \Delta_{g_i}, C_t = i\} \sim N(\Delta_{g_i}, 2\sigma_l^2). \quad (4)$$

The quantity $D_{gl}$ measures how much gene expression levels differ when measured under the treatment and control conditions. Since all the genes in cluster $l$ share the same variance $\sigma_l^2$, correlations among the expression levels of these genes are built into the modeling process. For the genes in different clusters, their expression levels are independent as they do not share any common parameter according to (4). Under the model given by (4), we do not need to estimate the cluster mean $\mu_l$.

To classify a new treatment $m + 1$ given the training sample described in Table 1, we consider the posterior classification probability

$$P(C_{m+1} = i|\{D_{g(l)t}\}, D_{m+1}), \quad i = 1, \ldots, K,$$

where $C_{m+1}$ is the class label of the new treatment. $\{D_{g(l)t}\} = \{D_{g(l)t}; g = 1, \ldots, n, t = 1, \ldots, m\}$ denotes the set of $D_{g(l)t}$ for all the genes and treatments in the training sample, and $D_{m+1} = \{D_{g(l)m+1}; g = 1, \ldots, n\}$ is the set of $D_{g(l)m+1}$ of all the genes for the new treatment $m + 1$. Essentially, $\{D_{g(l)t}\}$ and $D_{m+1}$ are the observed data and $C_{m+1}$ is the
decision rule. According to the Bayes theorem, we have

\[ P(C_{m+1} = j \mid \{D_{g(l)t}\}, D_{m+1}) \propto P(D_{m+1} \mid \{D_{g(l)t}\}, C_{m+1} = j) P(C_{m+1} = j). \]  

(5)

According to Result 1 in Section 3.1, the Bayes rule is to assign class \( j^* \) to \( C_{m+1} \) where

\[ j^* = \arg \max_j P(D_{m+1} \mid \{D_{g(l)t}\}, C_{m+1} = j) P(C_{m+1} = j). \]

If we assign equal prior probability to the class label \( C_{m+1} \), that is,

\[ P(C_{m+1} = j) = \frac{1}{K}, \quad j = 1, \ldots, K, \]

the Bayes rule is to choose \( j^* \) so that

\[ j^* = \arg \max_j P(D_{m+1} \mid \{D_{g(l)t}\}, C_{m+1} = j). \]

Note that

\[ P(D_{m+1} \mid \{D_{g(l)t}\}, C_{m+1} = j) = \int P(D_{m+1} \mid \{D_{g(l)t}\}, \{\Delta_{gi}\}, \{\sigma_i^2\}, C_{m+1} = j) \times \]

\[ P(\{\Delta_{gi}\}, \{\sigma_i^2\} \mid \{D_{g(l)t}\}) d\{\Delta_{gi}\} d\{\sigma_i^2\} \]  

(6)

where \( \{\Delta_{gi}\} \) and \( \{\sigma_i^2\} \) denote the set of parameters for all the genes, classes and clusters.

To evaluate the probability (6) above, we need to specify the prior distributions for \( \{\Delta_{gi}\} \) and \( \{\sigma_i^2\} \).

### 3.3 Prior and posterior distributions

The unknown parameters in (4) are the differential effect \( \Delta_{gi} \) and variance \( \sigma_i^2 \). We propose a family of conjugate priors as follows. For \( g = 1, \ldots, n, \ l = 1, \ldots, L, \ i = 1, \ldots, K, \) and \( a > 0, \)

\[ \Delta_{gi} \mid \sigma_i^2 \sim \text{indep. } N(\Delta_{gi0}, 2a^{-1}\sigma_i^2) \]  

(7)
and

\[ \sigma_0^{-2} \overset{\text{indep.}}{\sim} \text{Gamma}(\alpha_l, \beta_l) \]  

(8)

where \( \text{Gamma}(\cdot, \cdot) \) denotes the gamma distribution with density \( f(y | \alpha, \beta) = \frac{y^{\alpha-1}e^{-y/\beta}}{\Gamma(\alpha)\beta^\alpha} \).

The variable \( a \) in (7) is a nuisance parameter taking positive values. We estimate the prior parameters \( \Delta_{g0}, \alpha_l \), and \( \beta_l \) using the marginal method of moments (MMOM), the details of which are given in Appendix B.

With the conjugate priors given by (7) and (8), the integral in (6) becomes

\[
P(D_{m+1}|D_{g(t)l}, C_{m+1} = j) \propto \prod_{l=1}^{L} \left[ \left( \frac{1}{\xi_j} \right)^{n_l-n_{l-1}} \times \left( \frac{1}{\eta_{ji}} \right)^{\zeta_l} \right],
\]

(9)

\[ j = 1, \ldots, K, \] after substituting the parameters \( \Delta_{g0}, \alpha_l \), and \( \beta_l \) with their MMOM estimates where

\[
\xi_j \ = \ 1 + \frac{1}{a + k_j},
\]

\[
\eta_{ji} \ = \ \frac{\sum_{g \in l}(D_{g,m+1} - \hat{\Delta}_{g0})^2}{2\xi_j} + \frac{1}{2} \sum_{g \in l} \sum_{i=1}^{K} \sum_{t \in i} (D_{g(t)l} - \hat{\Delta}_{g0})^2 + \frac{1}{\beta_l},
\]

\[
\zeta_l \ = \ \frac{1}{2} (m + 1)(n_l - n_{l-1}) + \hat{\alpha}_l,
\]

and

\[
\hat{\Delta}_{g0} \ = \ \frac{\sum_{t \in i} D_{g(t)l}}{k_i}.
\]

The summation \( \sum_{g \in l} \) is over the genes in cluster \( l \) and the summation \( \sum_{t \in i} \) is over the treatments in class \( i \). The derivation of formula (9) is sketched in Appendix C.

Following the Bayes rule, the probability given in (9) is the classification probability to be calculated for each class \( j \). We need to find the class \( j^* \) that maximizes the right
hand side of (9). Intuitively, if we ignore the information in the gene expression levels, we would be more likely to assign the new treatment to the class containing more treatments, i.e., to the class with a larger value of $k_j$. Formula (9) indicates that when $k_j$ is large, $\xi_j$ is small and the classification probability given in the formula becomes large. So the Bayes rule appears to agree with the intuitive classification rule. However, with additional information from the data, the Bayes rule improves the intuition by incorporating the term $\eta_{jl}$ into formula (9). In the formula for $\eta_{jl}$, only the first term $\frac{\sum_{g \in l} (D_{g,m+1} - \hat{\Delta}_{g,j})^2}{2\xi_j}$ is related to $j$. A small value of this term makes $\eta_{jl}$ small and, in turn, makes the classification probability (9) large. This term resembles the usual sample variance if we view the denominator $\xi_j = 1 + \frac{1}{a + k_j}$ as a degree of freedom. So this term serves as a measure of variation. A smaller value of this variation implies a better fit. Hence, the probability given in (9) not only agrees with the intuitive classification rule, but also improves it by accounting for the variations in the model.

4 Simulation studies

Two simulation studies have been carried out to evaluate the performance of the proposed empirical Bayesian classification method in Section 3. In both studies, we consider the classification problems with three classes, i.e., $K = 3$.

In the first simulation study, we generate gene expression levels for $(9+ B)$ genes under $m$ treatments in each of the three classes. Among the $(9+ B)$ genes, $B$ of them have their expression levels sampled from the standard normal distribution $N(0, 1)$ for all the $3m$ treatments. Because their expression levels do not discriminate between classes, the $B$ genes are regarded as “non-informative” for the classification. The remaining 9 genes are
“informative” with expression levels generated from different distributions for different classes of treatments. Specifically, for treatment \( t \), three of the 9 genes have expression levels sampled from \( N(C_t \times \Delta_1, \sigma_1^2) \), another three from \( N(C_t \times \Delta_2, \sigma_2^2) \), and the remaining three from \( N(C_t \times \Delta_3, \sigma_3^2) \). Recall that \( C_t \), which takes values in \( \{1, 2, 3\} \), denotes the class label of treatment \( t \) for \( t = 1, ..., 3m \). The differential effects of treatments \( \Delta_1 \), \( \Delta_2 \), and \( \Delta_3 \), are sampled from normal priors \( N(1, 0.5) \), \( N(3, 0.5) \), and \( N(-3, 0.5) \), respectively. The precisions \( \sigma_1^{-2} \), \( \sigma_2^{-2} \), and \( \sigma_3^{-2} \) are sampled from gamma priors \( \text{Gamma}(2.5, 4/3) \), \( \text{Gamma}(3.5, 4/3) \), and \( \text{Gamma}(3, 4/3) \), respectively.

We illustrate a simple run of simulation with the following example. According to the simulation scheme, \( B \) genes have their expression levels sampled from \( N(0, 1) \) for all \( 3m \) treatments. For each of the remaining 9 “informative” genes, let \( X_g = \{X_{g1}, ..., X_{g,3m}\} \) denote the vector of gene expression levels measured for all the treatments, \( g = 1, ..., 9 \). Then for treatment \( t \) in class \( j \), gene 1, 4 and 7 have expression levels sampled from the same normal distribution as \( X_{1t}, X_{4t}, X_{7t} \sim N(j \times \Delta_1, \sigma_1^2) \), where \( \Delta_1 \) and \( \sigma_1^2 \) are generated for each of the three genes using the normal and gamma prior distributions; gene 2, 5 and 8 have their expression levels \( X_{2t}, X_{5t}, X_{8t} \sim N(j \times \Delta_2, \sigma_2^2) \); gene 3, 6 and 9 have their expression levels \( X_{3t}, X_{6t}, X_{9t} \sim N(j \times \Delta_3, \sigma_3^2) \). With this simulation scheme, the three genes in the same set not only form a cluster, but are predictive of the class of treatments.

With the remaining \( B \) genes forming one cluster, we have a total of four clusters: cluster 1 contains genes \( \{1, 4, 7\} \), cluster 2 contains genes \( \{2, 5, 8\} \), cluster 3 contains genes \( \{3, 6, 9\} \), and cluster 4 contains genes \( \{10, ..., B + 9\} \).

We implement our Bayesian method for the simulated data in which \( B \) equals 2000. We generate expression levels for the 2009 genes for \( 3m \) treatments with \( m = 10 \) and
For either value of $m$, we repeat 10 simulation runs and in each run we count the number of misclassified treatments using three different methods. The first is the proposed Bayesian classification method with the model-based clustering, the second is the proposed Bayesian classification method with the true cluster structure containing the four clusters, and in the last method, we force the cluster size to be one so that all genes are in one cluster. We denote the first method by “PMMC” (representing the proposed method with the model-based clustering), the second method by “PMTC” (representing the proposed method with the true cluster structure), and the third method by “PM1C” (representing the proposed method with the 1-cluster structure).

Among the three methods, the PMTC should be the best, producing the smallest number of classification errors since it uses the true cluster information of the 2009 genes. The PMMC method, which estimates the true cluster structure by the model-based clustering, might be inferior to the PMTC depending on how different the estimated cluster structure is from the true one and on how sensitive the proposed Bayesian classifier is to this difference. The PM1C method should be the worst of the three since it assumes one variance component for all the gene expressions.

******** Table 2 about here ********

The simulation results are summarized in Table 2. Entries in the table are the number of misclassifications out of $3 \times m$ treatments. Column 1 contains the index of simulation runs, and columns 2, 3, and 4 contain the results with the PMMC, PMTC and PM1C methods, respectively. As shown in Table 2, the number of misclassifications from the PMMC and the PMTC methods is always zero while the PM1C resulted in at least 3 misclassified treatments in all runs. The findings agree with our conjecture about the
performance of the three methods. Surprisingly, the PMMC method correctly classifies every treatment in all runs, just as does the PMTC method. Investigation of the clustering results from the model-based cluster method indicates that out of a total of 20 simulation runs, the model-based cluster method predicts four clusters 16 times and predicts only the three clusters four times. However, the estimated cluster labeling for genes does not agree exactly with the true cluster labeling. This seems to suggest that even when the estimated cluster structure is somewhat different from the true one, the proposed Bayesian classification method can still classify treatments correctly, demonstrating a robustness of the empirical Bayesian classification method.

In the first simulation study described above, we generate gene expression levels from normal distributions. In the second simulation study, we generate them from gamma distributions described below, while keeping all other simulation settings identical to the first simulation study. This second simulation study helps evaluate the sensitivity of the proposed method to the normality model assumption. We use the notation $\text{gamma}(\mu, \sigma^2)$ to denote a gamma distribution with mean $\mu$ and variance $\sigma^2$ so that the difference among the sampling gamma distributions can be readily recognized. Specifically, expression levels of $B$ “non-informative” genes are sampled from $\text{gamma}(5, 5)$, and the expression levels of 9 “informative” genes are sampled as follows. For treatment $t$ in class $j$, $X_{1t}, X_{4t}, X_{7t} \sim \text{gamma}(\mu_{1j}, \sigma_1^2)$, $X_{2t}, X_{5t}, X_{8t} \sim \text{gamma}(\mu_{2j}, \sigma_2^2)$, and $X_{3t}, X_{6t}, X_{9t} \sim \text{gamma}(\mu_{3j}, \sigma_3^2)$. The mean $\mu$ and variance $\sigma^2$ are sampled from another set of gamma distributions such that $\mu_{1j} \sim \text{gamma}(j \times 0.5, 1)$, $\mu_{2j} \sim \text{gamma}(j \times 5, 1)$, and $\mu_{3j} \sim \text{gamma}(j \times 5 + 15, 1)$; $\sigma_1^2 \sim \text{gamma}(1, 1)$ and $\sigma_2^2, \sigma_3^2 \sim \text{gamma}(2, 1)$. To see the differences among the sampling distributions for the means $\mu$, we plot the density curves of $\text{gamma}(j \times 0.5, 1)$, $\text{gamma}(j \times 5$, $\text{gamma}(j \times 5 + 15, 1)$, and $\text{gamma}(2, 1)$.
5, 2), and \( \gamma(j \times 5 + 15, 2) \) for \( j = 1, 2, 3 \) in Figure 1.

We let the number of “non-informative” genes \( B = 500 \) and the number of treatments \( m = 10 \). We increase the simulation runs from 10 to 500. We compare the number of misclassifications with the three methods, PMTC, PMMC, and PM1C, and summarize the simulation results in Figure 2. From the box-plot in Figure 2, the number of misclassifications from all three methods increases compared to the results in the first simulations. From the three boxes of numbers of misclassifications produced by the three methods, we observe a trend “PMTC < PMMC < PM1C” that describes the order of the mean and variance of the three boxes. Thus, not only does the proposed Bayesian classification method generally yield smaller errors, but it is more stable than the PM1C method. From the first histogram in Figure 2, which compares the PMMC and PMTC methods, we note that out of 500 runs, 250 times, or 50% of the time, the two methods both produce zero misclassification. We also note that the PMTC method never yields more misclassifications than the PMMC method, which is of no surprise. From the second histogram, which compares the PMMC and PM1C method, the PMMC method always gives fewer misclassifications than the PM1C except for a few runs.

5 Examples

5.1 Classification of toxicants with cDNA microarrays

The Bradfield Lab at the University of Wisconsin-Madison has been conducting microarray experiments in which gene expression profiles are used to classify commercial
chemicals into known classes of toxicants. According to the National Toxicology Program (NTP), a thorough analysis of each chemical requires $2 to 4 million and several years to complete (NTP 1996). Success in the microarray experiments will not only save a tremendous amount of time and money, but also change the way in which chemical analysis is conducted.

In one of the experiments from the Bradfield Lab, about 1,240 genes are selected with expression levels measured under twenty-four treatment conditions. The twenty-four treatments are combinations of different chemicals with multiple time courses which belong to five well-studied toxicological classes. Gene expression levels are also measured under twenty-four corresponding vehicle controls. The goal is to develop a classification rule that would accurately predict the class for each of the twenty-four treatments. See Thomas et al. (2001) for details.

In this example, we implement our method for several selected subsets of genes. We show that the classification results are not sensitive to the set of genes selected. We select each subset of genes using a simple score $d$ described as follows: For each gene, we calculate the average of the expression levels corresponding to all the treatments in each class; the score $d$ is the average of the absolute differences of all distinct pairs of these class averages. Basically, this score measures how much each gene discriminates between different classes. We choose $n$ genes with the largest $d$ scores. We note that this is a crude way of selecting genes. A more complicated procedure for gene selection could be used, but since we are focusing on classification problems and we want to investigate the sensitivity of the proposed method to the different sets of genes selected, there is no need for a more elaborate gene selection procedure. We further discuss this in the last section.
After $n$ genes are selected, we carry out the model-based cluster analysis to group them into clusters. As we do not have gene expression levels measured under control conditions for this data, we select gene expression levels measured under treatments in the same class for the model-based cluster analysis. Specifically, we select two classes that contain 10 treatments and 5 treatments, respectively. We obtain similar cluster structures with both of the classes. We draw a heatmap in Figure 3 to show the latent cluster structure, which is obtained using the class containing 10 treatments. In the figure, we select $n = 50$ genes, and group them into 5 clusters. This number of cluster is chosen according to the BIC criterion. From the bottom up, the cluster indices $l = 1, \ldots, 5$ represent the five clusters formed by the model-based cluster analysis.

************* Figure 3 about here ******************

We then employ the proposed method for this data. Since we know the class label for each of the twenty-four treatments, we use the leave-one-out cross validation to evaluate the performance of our method: we delete one treatment at a time and use the remaining twenty-three treatments with corresponding gene expression profiles to predict the class of the deleted treatment. The results are summarized in Table 3.

************* Table 3 about here ******************

By selecting $n = 40, 50, 100, 200, 300$ or 500 genes, we obtain a consistently low classification error rate. The number of misclassifications is no more than 1 for all the cases. This shows that the classification result is not sensitive to the set of selected genes. When we use all 1,242 genes, the number of misclassifications increases to 3. By including all 1,242 genes, we seem to introduce a lot of noisy genes which not only contribute no additional information to the classification, but also detract from correct classification.
5.2 Leukemia Data from Oligonucleotide Microarrays

The leukemia data set below was first reported in Golub et al. (1999) and has been analyzed extensively by a number of researchers. The data set is available on-line at http://www-genome.wi.mit.edu/mpr/data_set_ALL_AML.html.

According to Golub et al. (1999), about 7,200 genes are included. Initially, samples are classified into two types of leukemia, acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). Later, two subclasses in ALL are identified as T-cell and B-cell, and subsequently, we have a three-class classification problem. The training sample contains 34 observations, while another independent test sample contains 38 observations.

We carry out the same gene selection procedure according to the \( d \) score described in Section 5.1. We choose \( n \) genes with the largest \( d \) scores. We count the number of misclassified observations in the test sample using the Bayes rule based on the training sample as well as in the training sample via leave-one-out cross validation. We do this for both the two-class (AML vs. ALL) and the three-class (AML, T-ALL and B-ALL) problem with \( n = 100, 500, 1000, \) and \( 1500 \) genes. The results are summarized in Table 4. The number of misclassifications using the test sample is 1 in all cases and is no more than 1 when using leave-one-out cross validation for the training sample. The results in Table 4 are comparable to the ones obtained by other authors such as Dudoit et al. (2002).

************ Table 4 about here *******************
6 Discussion

We have developed an empirical Bayesian method for classifying treatments using microarray gene expression data. This method utilizes information contained in the latent cluster structure to achieve two goals: first, to reflect correlations among gene expressions in the modeling process and second, to solve the \( p \gg n \) issue by reducing the dimensionality of the problem substantially. We use a family of conjugate priors and obtain moment estimators for the hyper-parameters in the prior distributions, so that the usual Monte Carlo Markov Chain (MCMC) type of computation is avoided. We adopt the model-based method for cluster analysis because we can determine the number of clusters objectively and because it agrees with the assumptions of our Bayesian classification method. Any other sensible clustering method which is able to roughly capture the true underlying correlation structure among genes may be used together with the subsequent Bayesian classification method to form a powerful tool for classification of treatments.

From the results of the simulation studies and examples, the classification results by our method do not appear to be influenced by the set of genes selected at all. We note that the number of misclassifications is consistently small using most subsets of selected genes in both examples. In practice, this property can be useful if the accuracy of classification is of primary concern. For example, in the toxicological studies as the one in Thomas et al. (2001), a fast screen of chemicals is especially desirable, and our Bayesian method seems to fit well according to our first example.

Our method appears to be insensitive to the normality distribution assumption based on the results from the second simulation study. From the box-plot in Figure 2, the
number of misclassifications is within a relatively small range for both PMMC and PMTC methods. PMTC performs better than PMMC since the former uses the true cluster information, while the latter estimates it. Hence, the quality of cluster analysis directly affects the accuracy of classification, which supports our original idea of incorporating cluster structure of genes into classification.

We note that the success of the method is due to a proper model building process. From the model given by (2), the mean of the log normal distribution contains two parts. The first part is the cluster mean $\mu_l$ which is shared by the expression levels of all genes in the same cluster. The second part is the gene specific mean $\Delta_{gi}$ which measures the differences of gene expression levels between treatment and control conditions. The model attempts to reflect the underlying biological events. Usually, genes co-regulate biological functions through certain pathways. If the treatment affects some genes in the early steps of the pathway, all the latter genes in that pathway might consequently be influenced. However, if the treatment only affects genes in the last step of the pathway, then genes upstream will often remain unaffected. For example, let us consider a simple pathway “A→B→C” which indicates that the order of gene regulation is from gene A to B and then C. If a treatment affects gene A, expressions of gene B and gene C will consequently be influenced since they depend on gene A. However, if a treatment shuts off gene C, genes A and B usually will not be affected. As such, gene expressions are altered in different ways under different scenarios. Genetically, that justifies our introduction of gene specific parameters $\Delta_{gi}$. Statistically, $\Delta_{gi}$ of each gene is not determined by the specific treatment $t$, but by the class to which the treatment belongs. Thus, differences among $\Delta_{gi}$ will help discriminate between different classes.
Finally, we discuss the effect of nuisance parameter $a$ in the prior given by (7). In the moment estimation described in Appendix B, we assign a positive value for the parameter $a$. We claim that the value of $a$ hardly affects the results of classification. In fact, if we look at the classification rule given by (9), $a$ only appears in the denominator of $\xi_j$. When $a$ increases from 0 to $\infty$, $\xi_j$ decreases from $1 + \frac{1}{k_j}$ to 1 with only a change of $\frac{1}{k_j}$ with $k_j \geq 1$ for all $j$. So $a$ does not play a deciding role in the classification process. Taking smaller values of $a$ is favorable because it implies a vaguer prior for $\Delta_{gi}$ given by (7). However, with the MMOM empirical Bayes procedure, $a$ cannot be assigned zero as $a = 0$ results in an improper prior for $\Delta_{gi}$ which in turn will make the marginal distribution of $D_{g(i)t}$ improper. Since MMOM is based on the moments of this marginal distribution, our Bayesian method will then become infeasible.

References


A  EM for model-based cluster analysis

Under the microarray gene expression data with \( n \) genes and \( m \) treatments, \( \mathbf{x}_g \) denotes a \( m \times 1 \) vector of gene expression levels for gene \( g \). As we recommended, gene expression levels under control conditions or under the same treatment or same class of treatment conditions should be used for cluster analysis. Here, we assume that \( \mathbf{x}_g \) satisfies this requirement. Also, we assume that the gene expression levels are in log scale and suppress the log notation in the rest of the descriptions. According to the model-based clustering and our model assumptions, the multivariate normal density function \( f_l(\cdot) \) would have a simple form with constant mean vector \( \mu_l = \mu_l \mathbf{J} \) and diagonal covariance matrix \( \Sigma_l = \sigma_l^2 \mathbf{I} \), where \( \mathbf{J} \) is the vector with all elements being 1 and where \( \mu_l \) and \( \sigma_l^2 \) are two scalars. In the EM for model-based clustering, the “complete” data are considered to be \((\mathbf{x}_g, \mathbf{z}_g)\), where \( \mathbf{z}_g = (z_{g1}, \ldots, z_{gL}) \) with

\[
    z_{gl} = \begin{cases} 
        1 & \text{if gene } g \text{ belongs to cluster } l \\
        0 & \text{otherwise}
    \end{cases}.
\] (10)

Conditional on \( \mathbf{z}_g \), the density of an observation \( \mathbf{x}_g \) is given by \( \prod_{l=1}^L f_l(\mathbf{x}_g | \theta_l)^{z_{gl}} \). Each \( \mathbf{z}_g \) is assumed independent and identically distributed according to a multinomial distribution of one draw on \( L \) categories with probability \( \pi_1, \ldots, \pi_L \). So the log-likelihood for complete data can be written as

\[
l(\theta_1, \pi_1, z_{gl}) = \sum_{g=1}^n \sum_{l=1}^L z_{gl} [\log \pi_l f_l(\mathbf{x}_g | \theta_l)].
\] (11)

Define \( z_{gl}^* = E[z_{gl}|\mathbf{x}_g, \hat{\theta}_1, \ldots, \hat{\theta}_L] \) to be the conditional expectation of \( z_{gl} \) given the observation \( \mathbf{x}_g \) and parameter values \( \hat{\theta}_1, \ldots, \hat{\theta}_L \) which is a maximizer of (1). Consequently, \( z_{gl}^* \) is the conditional probability that gene \( g \) belongs to cluster \( l \); the clustering of gene \( g \)
is taken to be \( \{ j | z_{gl}^* = \max_i z_{gl}^i \} \). The EM algorithm iterates between an E-step where
values \( z_{gl}^* \) are computed from the data with the current parameter estimates, and a M-step in which the log-likelihood (11), with each \( z_{gl} \) replaced by its current conditional expectation \( z_{gl}^* \), is maximized with respect to the parameters. Under certain conditions, the algorithm has been proved to converge to a local maximum of the mixture likelihood (1).

The specific EM algorithm is as follows:

1. Initialize \( z_{gl}^* \). This can be done by, for example, classifying \( x_g \) to \( L \) clusters using the K-means cluster method.

2. **M-step**: Given \( z_{gl}^* \), maximize (11) with respect to the parameters. Note that

\[
\hat{\pi}_l = \frac{\sum_{g=1}^{n} z_{gl}^*}{n}
\]

\[
\hat{\mu}_l = \frac{\sum_{g=1}^{n} z_{gl}^* \bar{x}_g}{\sum_{i=1}^{n} z_{gl}^*}
\]

\[
\hat{\Sigma}_l^2 = \frac{\sum_{g=1}^{n} \sum_{j=1}^{m} z_{gl}^*(x_{gj} - \mu_l)^2/(nm)}
\]

where \( \bar{x}_g = \frac{\sum x_{gj}}{m} \).

3. **E-step**: Compute \( z_{gl}^* \) given the parameter estimates from the M-step

\[
z_{gl}^* = E[z_{gl}|x_g, \hat{\theta}_1, \ldots, \hat{\theta}_L] = P(z_{gl} = 1|x_g, \hat{\theta}_1, \ldots, \hat{\theta}_L)
\]

\[
= \frac{f_l(x_g|x_{gj} = 1, \hat{\theta}_j, \ldots, \hat{\theta}_L)P(z_{gl} = 1)}{\sum_{j=1}^{L} f_l(x_g|x_{gj} = 1, \hat{\theta}_j, \ldots, \hat{\theta}_L)P(z_{gj} = 1)}
\]

\[
= \frac{\hat{\pi}_l f_l(x_g|\hat{\mu}_l, \Sigma_l)}{\sum_{j=1}^{L} \hat{\pi}_j f_j(x_g|\hat{\mu}_j, \Sigma_j)}
\]
4. repeat M-step and E-step until convergence criteria are satisfied.

For a range of values of $L$, we compute corresponding BIC and the final $L$ is chosen at which maximum BIC is achieved. Note that $BIC = -2 \log(\hat{L}) + \log(N)K$ where $\hat{L}$ is the maximum likelihood and $N$ and $K$ are respectively the number of observations in the data and the number of unknown parameters in the model.
B Empirical Bayes marginal method of moments estimators

Consider the marginal moments of $D_{g(l)t}$. For the mean, we have

$$E(D_{g(l)t}) = E[E[D_{g(l)t} | \Delta_{gi}, \sigma_t^2]] = E[\Delta_{gi}] = \Delta_{g0}.$$  

According to the method of moments, we estimate $\Delta_{g0}$ by

$$\hat{\Delta}_{g0} = \sum_{k \in i} \frac{D_{g(l)t}}{k_i}. \quad (12)$$

For the variance of $D_{g(l)t}$, we have

$$Var(D_{g(l)t}) = Var[E[D_{g(l)t} | \Delta_{gi}, \sigma_t^2]] + E[Var[D_{g(l)t} | \Delta_{gi}, \sigma_t^2]]$$

$$= Var(\Delta_{gi}) + 2E(\sigma_t^2)$$

$$= E[Var(\Delta_{gi} | \sigma_t^2)] + 2E(\sigma_t^2)$$

$$= \frac{2(1 + 1/a)}{\beta_t(\alpha_t - 1)}, \quad (13)$$

where $a$ is a positive value. $\alpha_t$ is the shape parameter of the inverse gamma prior for variance $\sigma_t^2$. When $\alpha_t$ is small, the variation coefficient of the inverse gamma distribution is large, which implies a relative vague prior for $\sigma_t^2$. In this article, we choose $\alpha_t = 1.5$ for all $l$. Note that $\alpha_t$ must be greater than 1 in order to make the marginal variance of $D_{g(l)t}$ positive. For a given $\alpha_t$, the scale parameter $\beta_t$ can be obtained via the method of moment according to (13). For example, if $a = 2$ and $\hat{\alpha}_t = 1.5$, then

$$\hat{\beta}_t = \frac{2(1 + 1/a)}{S_t^2(\hat{\alpha}_t - 1)}$$

$$= \frac{6}{S_t^2},$$

where $S_t^2$ is the sample variance of $\{D_{g(l)t}\}$ for all the genes in cluster $l$ for $l = 1, ..., L$. 

28
C Derivation of classification probability

We briefly illustrate how to derive the key classification probability written as formula (9) in Section 3.3.

First, it is easy to obtain $P(D_{m+1}|\{D_{g_{(l)t}}\}, \{\Delta_{gi}\}, \{\sigma^2_i\}, C_{m+1} = j)$ according to the model assumptions. So

$$P(D_{m+1}|\{D_{g_{(l)t}}\}, \{\Delta_{gi}\}, \{\sigma^2_i\}, C_{m+1} = j) = \prod_{l=1}^{L} \prod_{g=n_{i-1}+1}^{n_i} \phi(D_{g,m+1}|\Delta_{g_j}, 2\sigma^2_i),$$

where $\phi(\cdot|\mu, \sigma^2)$ is the density function of the normal distribution with mean $\mu$ and variance $\sigma^2$. This takes care of the first part of integrand in formula (6).

For the second part, $P(\{\Delta_{gi}\}, \{\sigma^2_i\}|\{D_{g_{(l)t}}\})$, we rewrite it as

$$P(\{\Delta_{gi}\}, \{\sigma^2_i\}|\{D_{g_{(l)t}}\}) = P(\{\Delta_{gi}\}|\{\sigma^2_i\}, \{D_{g_{(l)t}}\}) \times P(\{\sigma^2_i\}|\{D_{g_{(l)t}}\}).$$

Calculation for the first term, $P(\{\Delta_{gi}\}|\{\sigma^2_i\}, \{D_{g_{(l)t}}\})$, is standard using Bayes theorem and model assumptions. Plugging in MMOM estimators for the hyper-parameters in the prior distributions as described in Appendix B, we can show that

$$P(\{\Delta_{gi}\}|\{\sigma^2_i\}, \{D_{g_{(l)t}}\}) = \prod_{l=1}^{L} \prod_{g=n_{i-1}+1}^{n_i} \phi(\Delta_{gi}|\hat{\Delta}_{g_{i0}}, \frac{\sigma^2_i}{(k_i + a)}),$$

where $k_i$ is the number of treatments in class $i$. Calculation for the second term in (15), $P(\{\sigma^2_i\}|\{D_{g_{(l)t}}\})$, is similar but more tedious. For simplicity, we will not write down the details. It turns out that $P(\{\sigma^2_i\}|\{D_{g_{(l)t}}\})$ is proportional to a product of densities of the inverse gamma distributions. Since the inverse gamma and normal distributions belong to a conjugate family, the integration in the probability given by (6) can be solved analytically. After some algebra, we can write this key classification probability in the form of (9).
<table>
<thead>
<tr>
<th>Trt ID</th>
<th>class label</th>
<th>cluster 1</th>
<th>cluster 2 ⋯</th>
<th>⋯</th>
<th>cluster L</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$C_1$</td>
<td>$x_{11}$</td>
<td>$x_{n1,1}$</td>
<td>$x_{n1+1,1}$</td>
<td>$x_{n2,1}$</td>
</tr>
<tr>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
<td>⋮</td>
</tr>
<tr>
<td>$m$</td>
<td>$C_m$</td>
<td>$x_{1m}$</td>
<td>$x_{n1,m}$</td>
<td>$x_{n1+1,m}$</td>
<td>$x_{n2,m}$</td>
</tr>
</tbody>
</table>

Table 1: Latent Cluster structure of microarray gene expression data. $x_{gt}$ is the expression level of gene $g$ measured under treatment $t$, $g = 1, ..., n_L$ and $t = 1, ..., m$.

<table>
<thead>
<tr>
<th>Simulation runs with $m = 10$</th>
<th>PMMC</th>
<th>PMTC</th>
<th>PM1C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>7</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>0</td>
<td>5</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>0</td>
<td>8</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Simulation runs with $m = 30$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
<tr>
<td>10</td>
</tr>
</tbody>
</table>

Table 2: Results of simulation study I. The entries are the number of misclassifications out of $3m$ treatments.
<table>
<thead>
<tr>
<th>No. genes</th>
<th>40</th>
<th>50</th>
<th>100</th>
<th>200</th>
<th>300</th>
<th>500</th>
<th>1242</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. clusters</td>
<td>7</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>6</td>
<td>9</td>
</tr>
<tr>
<td>No. misclassifications (rate)</td>
<td>1 (4%)</td>
<td>1 (4%)</td>
<td>1 (4%)</td>
<td>1 (4%)</td>
<td>1 (4%)</td>
<td>0 (0%)</td>
<td>3 (12%)</td>
</tr>
</tbody>
</table>

Table 3: Results for classifying twenty-four treatments with cDNA microarrays. The number of clusters are predicted by the model-based cluster analysis with the BIC criterion. The number of misclassifications is made out of classifying twenty-four treatments using leave-one-out cross validation.

<table>
<thead>
<tr>
<th>No. genes</th>
<th>1500</th>
<th>1000</th>
<th>500</th>
<th>100</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. clusters</td>
<td>28</td>
<td>25</td>
<td>18</td>
<td>10</td>
</tr>
<tr>
<td>2-class</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3-class</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>TS</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>CV</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 4: Entries are numbers of misclassifications. $n$ is the number of genes selected and $L$ is the number of clusters obtained by the model-based cluster method. “TS” represents the number of misclassifications for the test sample and “CV” represents the number of misclassifications for the training sample using leave-one-out cross validation.
Figure 1: Gamma density curves used in the second simulation to generate 9 informative genes. Each row contains three gamma distributions from which expression levels of three informative genes are sampled for three different classes.
Figure 2: Summary plots for the results from second simulation. The top box-plot contains the number of misclassifications for the three methods. The middle histogram shows the difference of misclassifications between the PMMC and the PMTC methods. The bottom histograms shows the difference of misclassifications between the PMMC and the PM1C methods.
Figure 3: Heat map of the gene expression profiles: 50 genes selected based on the $d$ score are grouped into 5 clusters. The vertical ordinate represents the 50 genes, and the horizontal ordinate represents the 10 treatments in the same class.