Manuscript 3

Mixture models applied to microarray gene expression data

Bjarte Dysvik[1], Trond Hellem Bø[1] and Inge Jonassen[1,2]
1 Department of Informatics, University of Bergen, Norway. 2 Computational Biology Unit, Bergen Centre for Computational Science, University of Bergen, Norway

ABSTRACT

Feature selection is an important method for finding underlying processes and systems in microarray gene expression data. Microarray experiments often involve questions such as which genes are changing between classes of samples. Given a gene expression dataset and sample class labels, feature selection methods aim at extracting features that allow classification of a new example into one of the classes. We present a novel method based on mixture models for feature selection and sample classification applied to gene expression data. We show that the method is able to compete with and even outperforms existing methods in terms of classification accuracy on a set previously published datasets.

INTRODUCTION

The primary focus of this paper is on developing a methodology allowing more flexible probability models to capture the distribution of a gene's expression values across a set of microarrays. Most existing methods for identification of differentially expressed genes, for feature selection, and for class prediction, assume that a gene's expression values is normally distributed within each class of arrays. We investigate whether this assumption sometimes is inappropriate and whether superior results can be obtained using more flexible models. We investigate mixture models and develop methods for estimating a mixture model from a set of expression measurements, for utilizing these models in feature subset selection, and in class prediction. The methods allow for multi-class analysis and prediction. We analyze the effectiveness of the approach on a range of typical data sets and compare the results to those obtained using existing methods. Our novel method in most cases gives prediction accuracies about the same as the other methods and in some cases it produces superior results.

Background and terminology

Microarrays can be used to obtain measurements of transcript abundance for thousands of genes in parallel [1, 2]. A microarray typically contains probes for all genes in the genome of the organism under study in addition to control probes used for calibration and quality control. Microarray experiments often aim to investigate the difference, in terms of gene expression patterns, between different classes of biological samples or to follow the behavior of a biological system in time through a biological process of interest [3-5]. In the first case, the samples in each class have some genotypic of phenotypic feature in common that is not shared across the groups. The features defining the classes should be characterized using other information and not be based on the microarray data. For instance, each class could contain samples of tissue with one specific cancer form or one class could contain cancerous tissue

samples and another benign samples. In such cases, one is interested in finding which genes have different expression pattern between the classes and such genes can be found by using methods for identification of differentially expressed genes such as the t-test or SAM [6]. For each gene one reports to be differentially expressed between the classes, one often also reports a p-value that is the probability of observing an expression pattern difference of the observed magnitude under the null-hypothesis that there is no difference in expression between the classes. Since thousands of genes are analyzed, one needs to correct for multiple testing, and since Bonferroni correction in most cases is overly conservative, one typically reports the false discovery rate (FDR) for a list as an estimate of the proportion of genes on the list that might be false positives [7]. We will refer to this analysis as *marker gene identification*. Most methods are best suited for analyzing experiments with two classes, but there are also methods for multi-class analysis.

In addition to producing lists of genes and groups of genes showing differential expression between pre-defined groups of samples, one is often interested in finding out whether the expression patterns of the different classes are sufficiently different to allow prediction, i.e., to predict which class a new un-labeled sample belongs to. This type of analysis is referred to as *class prediction* [8, 9]. In a number of cases it has been shown that microarray gene expression data allows prediction of disease type or outcome with a precision that is higher than that obtained using previously established methods (e.g. [10]). Different classification methods have been successfully applied including linear discriminants, support vector machines, neural networks, and k-nearest neighbors classification [1, 11, 12].

In order to estimate the prediction accuracy that can be expected, a method known as cross-validation is typically used. This involves training the classifier on the expression data for a subset of the samples (training set) and testing the learned classifier on the held-back examples (test set). An additional analysis sometimes done is to check if the accuracy obtained is better than what would be obtained if there was no systematic difference between the classes under analysis [13]. This can be done by a performing an analysis of the same data set for a number of permutations, each permutation randomly scrambling the relationship between the samples and the classes, and recording for how many of the permuted data sets one obtains a prediction accuracy as high as that obtained on the real data set.

Performing class prediction on typical microarray gene expression data sets, one is faced with a problem known as the curse of dimensionality. The number of features is in the thousands while the number of examples in each class is at best in the tens, and many classification methods will not work well, or at all, in such cases. In addition, a majority of the features (genes) will carry no information relevant for the difference between the classes and only add noise making classification even harder. Therefore it is desirable to limit the set of features to a smaller number either by selecting a subset of the features to be used for classification or by defining a set of new features based on the original ones. The former option is known as *feature subset selection (FSS)* [14]. Methods for identification of marker genes can also be appropriate for identifying features useful for classification. The evaluation of features in the FSS step should somehow be consistent with the working of the classifier to be employed on the selected feature set. One way to ensure this is to use the classifier to evaluate feature sets, known as wrapper method for FSS [15]. Alternatively, in filter methods, the FSS step uses a separate method for assessing features and sets of feature in order to produce a feature set to be used in the classification step.

Many methods commonly applied for marker gene identification, for feature subset selection, and for classification make the assumption that the expression values obtained for a gene in samples in any one class follows a parametric statistical distribution, typically the normal distribution. This includes methods such as t-tests, regularized t-tests such as SAM and empirical bayes, and linear discriminants. Even if the assumption appears to be supported and at least give useful results in many cases, there is reason to believe that cases exist where this assumption is far from appropriate. One reason is that a pre-defined class may in fact be composed of sub-classes each with different expression patterns. For instance in a group of cancer patients with similar phenotype, there could be different genomic alterations each giving rise to distinct expression patterns all resulting in similar observable clinical characteristics [9, 16]. Methods for clustering and class discovery could be used to look for such subclasses in cases where class prediction gives lower than desired prediction accuracy. Alternatively one could apply methods allowing more flexible modeling of gene expression values.

Mixture models are statistical models that express one probability density function as a weighted sum of a number of simpler functions, for example normal distributions [17]. Mixture models have been used in bioinformatics, for example as Dirichlet mixtures in sequence analysis [18] and more recently in the analysis of microarray gene expression data, e.g., Pan et al used them to estimate the number of replicates needed in an experiment [19] and McLachlan et al used them in clustering [17].

Overview of novel method

For each gene, we construct a mixture model capturing its distribution of expression values, across the samples in each class. The mixture model is simply a sum of normal distributions, one for each observed expression value centered at the observed value, and all normal distributions having the same variance and equal weight in the mixture. The variance is chosen so that the resulting mixture model becomes a smooth function and so that its cumulative distribution approximates the empirical cumulative distribution.

For feature selection we calculate a statistic for each gene reflecting how well it is suited for classifying one class (target class) versus the rest. The statistic is calculated based on the mixture models for that gene for each of the classes and is simply the probability of classification error given the mixture models for the sample classes. The statistic thus reflects the probability of a vote for the wrong class status assuming that the mixture models accurately estimate the true probability distributions for each of the classes.

For class prediction, each class is treated separately as target-class versus the rest, and the chosen features are used to accumulate votes for/against membership in the target class. The class that receives the most votes in favor of membership is the one predicted for the new sample.

Finally we explore how mixture models can be used in unsupervised data analysis. One simple approach is to estimate both a normal distribution and a mixture model for each gene in each class and evaluate the difference between the two probability density functions. We argue that a larger difference reflects a higher chance that the gene reveals the existence of subclasses or expression outliers.

MATERIALS AND METHODS

Mixture Models

A mixture model is a model composed of two or more independent probability distributions. We explore characterizing the distribution of a gene's expression values across a set of samples by a mixture of normal distributions. Each distribution in the mixture model may have an individual mean and variance. The combined probability density function is then defined as the weighted sum of the component probability functions. For a mixture of p equally weighted normal distributions we get:

$$P(x) = \frac{1}{p}\sum_{j=1}^{p}\frac{1}{\sqrt{2\pi}\sigma_j}e^{-\frac{(x-\mu_j)^2}{2\sigma_j^2}}$$

Here $P(x)$ is the modeled probability density function and p is the number of components in the model. In our implementation, each gene is modeled with one mixture model for each class of samples. Given an expression profile for gene $i, X_i = (x_{i1},...,x_{ip})$, we model the distribution of expression values with the mixture model

$$f_i(x) = \frac{1}{p}\sum_{j=1}^{p}\frac{1}{\sqrt{2\pi}\sigma_i}e^{-\frac{(x-x_{ij})^2}{2\sigma_i^2}}$$

The contributing normal distributions have the same standard deviation σ_i, and are centered at the observed expression values for that gene. The cumulative distribution of the mixture will fit better to the cumulative distribution of the observed expression values the smaller we set σ_i. The challenge is to find an appropriate σ_i, so that the cumulative distribution of the mixture is neither under- (too large σ_i) nor over-fitted (too small σ_i) to the cumulative distribution of the observed data (see Figure 1).

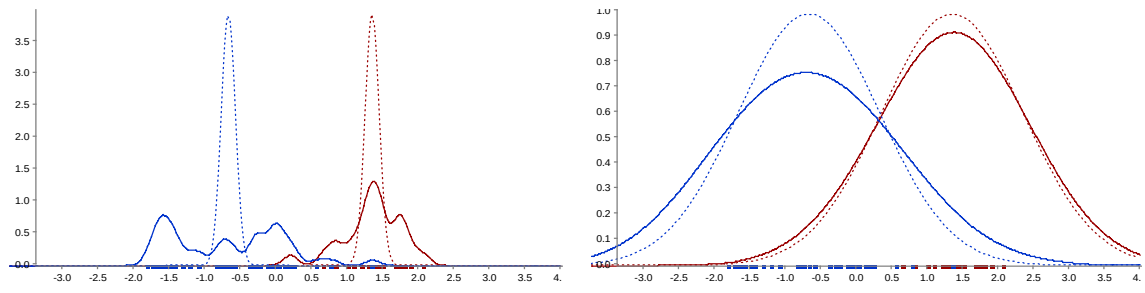

Figure 1: Too low (left) standard deviation leads to overfitting while too high standard deviation (right) decreases flexibility. Whole lines represent the mixture model probability density function defined by a mixture model with one normal distribution per data point (indicated by squares on the horizontal axis). Dotted lines represent the standard normal distribution estimated from the same data points.

The fit between our mixture model and the cumulative distribution of the observed expression levels can be quantified using the Kolmogorov-Smirnov (KS) goodness of fit [20]. We devised a simple search method to find a value for σ_i (for each gene i) so that the KS fit is equal to the median expected KS fit for normal distributed data. This

search method works reasonably well in most cases. However, in some cases this approach leads to severe over- or under-fitting, and in addition it is rather time-consuming. A simpler, yet more successful approach is to define σ_i as a function depending on the empirical standard deviation s_i and the number of observations p. We found that the following function leads to a mixture that approximately follows the normal distribution if the data is normal distributed, while it otherwise follows the distribution of the data points well.

$$\sigma_i = \frac{s_i}{\ln(2\sqrt{p})}$$

Tests performed using the KS statistic for goodness of fit, shows that the resulting mixture fits well to the data. The KS statistics obtained with the mixture model are in a reasonable range compared to a distribution of KS statistics from simulations, e.g. the distributions are approximately the same but with smaller variance using the mixture model. The simulations used randomly drawn observations from a normal distribution, and the KS statistics from the normal distribution fit to these data points.

One might argue that the described mixture model could have been simplified to a mixture of fewer normal distributions, as the described model contains many parameters. Estimating the optimal number of normal distributions together with parameters for each of these is far from trivial. Our proposed scheme leads to straightforward parameter estimation and, importantly, the resulting model fits well to the cumulative distribution of the observed data points and give excellent results in a practical setting (see Results).

Parameter estimation in the presence of sample classes

In the presence of sample classes, we will estimate one standard deviation $\sigma_{i,j}$ for each gene i for each sample class j. This will be based on the standard deviation calculated from the observed expression values that can be calculated for each class separately and optionally pooled over the classes. This leads to the following adjustments to the function for $\sigma_{i,j}$:

$$\sigma_{i,j} = \frac{s_{i,j}}{\ln(2\sqrt{p_j})}$$

leading to a $\sigma_{i,j}$ for a class that is based on the standard deviation in that class only ($s_{i,j}$ is the standard deviation of the expression measurements for gene i in class j, and p_j is the size of class j). Alternatively, we can calculate $s_{i,pooled}$, the standard deviation pooled over all sample classes, resulting in

$$\sigma_{i,j} = \frac{s_{i,pooled}}{\ln(2\sqrt{p_j})},$$

leading to a $\sigma_{i,j}$ for each class that depends on the standard deviation across all classes. We refer to the mixture model using a group-wise standard deviation as MMg and the mixture model using a pooled standard deviation as MMp.

Class prediction and feature selection for two classes.

We first consider the case with two sample classes. We describe a simple scheme for predicting the class of a new sample X based on its expression values ($x_1,x_2,...,x_K$) for the K genes in the selected feature set. It is assumed that a mixture model $f_{i,j}$ has been

defined for each gene i in each of the two classes ($j=1,2$). The class predicted for X is the one that receives the most votes where each of the K genes gives a vote for either class 1 or class 2. Gene i gives a vote for class 1 if $f_{i,1}(x_i) \geq f_{i,2}(x_i)$, for class 2 if $f_{i,2}(x_i) > f_{i,1}(x_i)$. In other words, a vote is given for the class that has the mixture model with the highest probability for the observed expression value. If both classes receive the same number of votes, it is counted as a wrong classification.

In the training phase, we are given a set of sample expression profiles, some belonging to class 1 and some to class 2. The number of features to be used is given as a parameter K. In order to perform feature subset selection, we estimate for every gene a mixture model for each of the two classes. Let us assume that the mixture models reflect the underlying distribution of gene expression values in each of the two classes and consequently that future examples will follow the same distribution. It is then easy to see that the probability of gene i giving a vote for the wrong class is $S_i = \int_{x:f_{i,1}(x)>f_{i,2}(x)} f_{i,2}(x)dx + \int_{x:f_{i,2}(x)>f_{i,1}(x)} f_{i,1}(x)dx = P(1|2) + P(2|1)$, where the first (respectively, second) element is the probability that gene i will contribute a vote for class 1 (2) if it comes from class 2 (1) (see Figure 2).

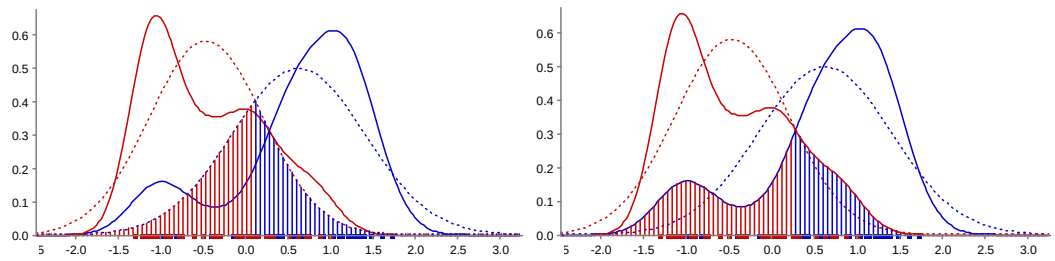

Figure 2: A feature in the prostate dataset (row #5808) shown with the two groups in different colors (group1 = red and group2 = blue). The mixture model (MMg) is in solid and the corresponding normal distribution is dotted. The chart illustrates the mixture model scoring S_i, so that the striped pattern for the mixture models (right plot) and corresponding normal distributions (left plot). When scoring for the red class, the red striped area corresponds to the first element, $P(1|2)$, of the equation and the blue striped area corresponds to the second element, $P(2|1)$, of the equation above.

The feature subset selection procedure is

(1) Estimate mixture models $f_{i,1}$ and $f_{,i2}$ for each gene i,
(2) Calculate S_i for each gene i,
(3) Select the K genes with the lowest S_i values.

Extension to more than two classes

We extend the voting scheme to be used in the multi-class case by first considering each class separately performing a one-versus-rest voting resulting in a number of votes for the class (positive votes) and a number of votes for "the others" (negative votes), and based on this assigning the sample to the class that received the most positive votes. We choose to perform the voting based on one MM for each gene and each sample class. Assume that we have C classes and that the sample to be classified has expression values $X=(x_1,x_2,\ldots,x_K)$. Considering class 1 and gene i, a positive vote (in favor of class 1) results if $f_{i,1}(x_i)$ is larger than each of $f_{i,2}(x_i)$, …, $f_{i,C}(x_i)$. Since

different genes may be better suited to distinguish each class from the rest, we propose a FSS method that identifies in an independent manner a set of *K* genes for each class. For class 1 we calculate for each gene *i* the statistic

$$S_i^1 = \sum_{j \neq 1} \int_{x: f_{i,1}(x) > f_{i,j}(x)} f_{i,j}(x)dx + \int_{x: f_{i,j}(x) > f_{i,1}(x)} f_{i,1}(x)dx = P(1 \mid \neg 1) + P(\neg 1 \mid 1),$$

where the first (respectively, second) integral is the probability that gene i will contribute a vote for class 1 (against class 1) if it comes from one of the other classes (class 1). The measure is the probability of gene *i* contributing an incorrect vote under the assumption that the mixture models $f_{i,j}$ capture the underlying probabilities. The present formulation corresponds to assuming equal prior probabilities for each class; the equation could be extended to take into account different priors, but this will not be explored in this paper. The feature subset selection procedure for more than two classes becomes

(1) Estimate mixture models $f_{i,1}, f_{i,2,\ldots,} f_{i,C}$ for each gene *i*,
(2) For each class *c*
    a. Calculate S_i^c for each gene *i,*
    b. Select as feature set for class *c* the *p* genes with the lowest S_i^c values.

It should be noted that if this is applied on a problem with two classes, it results in the approach described above for two classes. In this case, the same genes will be chosen by FSS for each of the two classes.

Diagonal Linear Discriminant Analysis

For comparison purposes, we have used diagonal linear discriminant analysis (DLDA) which has shown to perform surprisingly well [21]. DLDA is a maximum likelihood discriminant rule for multivariate normal class densities and, in contrast to mixture models, it assumes a normal distribution of values for each class. The classifier rule is a simple linear rule that assigns class membership to the class *k* that minimizes

$$C_k = \sum_{j=1}^{p} \frac{\left(x_j - \mu_{kj}\right)^2}{\sigma_j^2}$$

where p is the number of features, x_j is the expression of gene *j*, μ_{kj} is the sample mean of gene *j* in class *k* and σ_j^2 is the pooled variance for gene *j*.

Two class DLDA

For two-class data, the feature selection for DLDA is done by simply scoring each gene using Wilcoxon statistics applied to class 1 vs. class 2 and picking the top *p* genes from the list. A sample is classified as class 1 if C1 < C2, and otherwise to class 2. The genes $x_{1\ldots}$ x_p are from the top of the Wilcoxon score list.

Extension to more than two classes

The classification rule is a simple extension of that for two classes so that a C_i is calculated for each class *i=1,….,k* and the sample is classified according to the smallest value. The feature set is produced by merging the one-versus-all Wilcoxon score lists for each class so that the same number of genes from each list is included in the merged list. The merging procedure selects one feature at the time from each of the class feature lists. If feature with index j is already selected from one of the other class lists, j+1 is added instead.

Estimating prediction accuracy by cross-validation

We performed two basic forms of cross-validation. The first is known as leave one out cross-validation (LOOCV). Here FSS and estimation of gene- and class-specific mixture models is done on all but one of the available examples, and this is used to classify the one remaining example. This is repeated so that each example is held back once and the prediction accuracy is reported as the proportion of examples that are correctly classified. One advantage with this method is that it will always produce the same results since there is no random element in the procedure itself (theoretically differences can occur in the case of ties in FSS and in the number of votes for each class). However, one can argue that better estimates of the prediction accuracy can be obtained by using a larger subset of the examples as test set. For this reason we also use leave one third out cross-validation (LOTOCV). Here a third of the samples are left out in the training phase and used to test the prediction accuracy. The procedure should be balanced so that the training and test sets contain approximately the same proportion of each of the classes. In LOTOCV, one will perform training on two-thirds of the examples and testing on the remaining and repeat this many times reporting the average prediction accuracy obtained. This procedure should be also be balanced so that each example is included as test example the same number of times. Specifically, for the data sets analyzed in this study (see Results) there are some "hard cases" (samples that are easily assigned to the wrong class) and the accuracy will strongly depend on how many times these were included in the test set. We refer to this (doubly) balanced version of LOTOCV as balanced (BLOTOCV).

RESULTS

In this section we present results on the performance of the new methods for feature subset selection and class prediction on a number of publicly available data sets for which other investigators have reported results obtained using a number of previously published methods. We use in particular the Dettling et al paper [21] that compare the prediction accuracy obtained using a number of different class prediction methods among them Diagonal Linear Discriminant (DLD) classification and Wilcoxon feature scoring. Dettling et al do not provide sufficient details to allow precise reproduction of results. Therefore we include DLD/Wilcoxon in our tests to allow indirect comparison of the MM based methods to those included by Dettling et al. In this section we first give a brief description of the data sets used. Then we present the results of an analysis leading to the choice of a particular number of features to be used in our class prediction experiments. The results obtained using our MM based method on the Dettling et al data sets are presented and finally we present some preliminary results regarding use of the MM based method for identification of interesting features.

Data sets

We downloaded the datasets (see Table 1) from Dettling et al's web site at http://stat.ethz.ch/~dettling/bagboost.html and used the preprocessed expression data matrices available there.

Dataset	Number of genes, samples	Number and sizes of sample classes
Leukemia [9]	3571, 72	2 (25, 47)
Prostate [14]	6033, 102	2 (52, 50)
Lymphoma [4]	4026, 42	3 (42,9,11)
SRBCT [12]	2308, 63	4 (23, 20, 12, 8)
Brain tumor [22]	5597, 42	5 (10, 10, 10, 4, 8)
Colon [5]	2000, 62	2 (40, 22)

Table 1: Summary data about the data sets used to test the feature selection and class prediction methods. See original publications and Dettling et al for more information on the datasets.

Choosing the number of features to be used

The accuracy obtained in class prediction depends on the feature sets utilized and in particular on the number of features used. For the MM based methods we utilized the feature scoring described in the Methods section and used the top *p* features for class prediction. We chose to limit our analysis to one particular value of *p* and we performed a BLOTOCV analysis using DLDA for different *p* values (ranging from 5 to 500) on all the datasets and on two of the datasets (colon and brain) using MM to have a basis for choosing an appropriate value for *p*. The value *p*=99 seemed to be a reasonable number of features to use in subsequent testing (See figure 3).

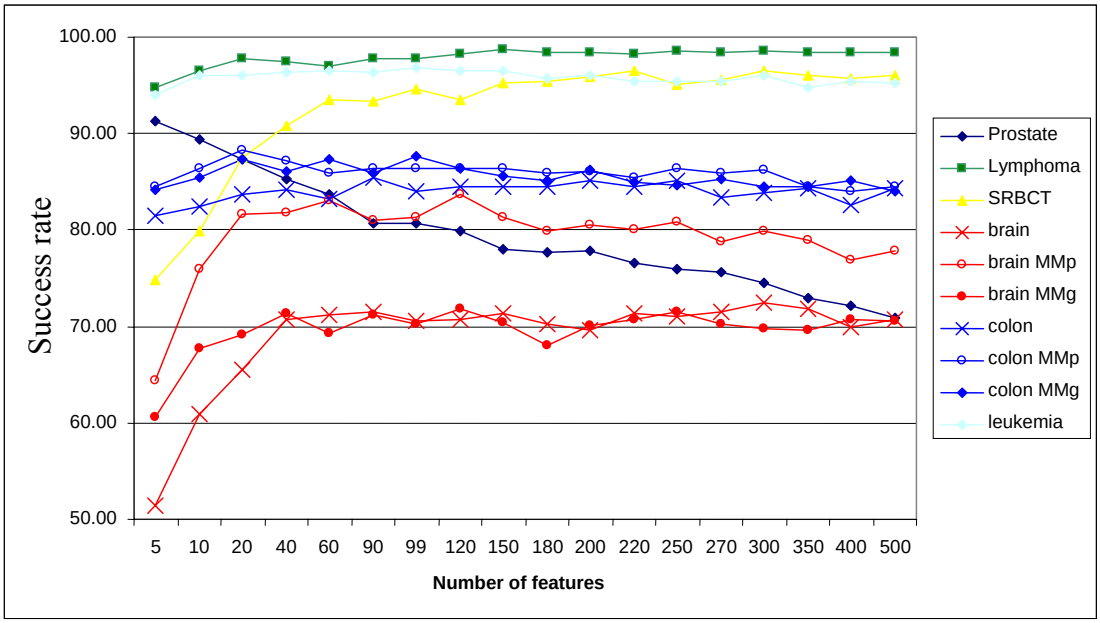

Figure 3: Success rate using different feature sizes. Prediction success rate is estimated using a BLOTOCV for all datasets described above. Datasets in the legend without suffix are based on our own implementation of the DLDA method described in the Methods section. MMp corresponds to the pooled standard deviation and MMg corresponds to the group-based standard deviation (see details in the Methods section). Note that the spacing of points along the x-axis is not linear. Success rate using 99 features is included for comparison to Dettling's implementation of bagboost DLDA in Table 3.

Effect of balancing LOTOCV

The results obtained using leave-one-third-out-cross-validation (LOTOCV) method is highly dependent on the number of times "hard cases" have been included in the test set (see Methods). In balanced LOTOCV we simply ensure that every sample is left out (included in test set) exactly once every three LOTOCV cycles. We performed a comparison between prediction accuracies estimated using LOTOCV and BLOTOCV where we 100 times estimated the prediction accuracy using 51 LOTOCV (respectively BLOTOCV) cycles and the Wilcoxon/DLDA FSS/class prediction methods and recorded the estimates for each of the 100 runs. The results are summarized in Figure 4 that shows that the prediction accuracies obtained using BLOTOCV are more consistent (less spread) than those obtained using LOTOCV. Note that this analysis shows the variation of the mean prediction rate over 100 sets of 51 LOTOCV cycles – while Figure 2 in Dettling et al shows the variation of prediction rate over 50 LOTOCV cycles.

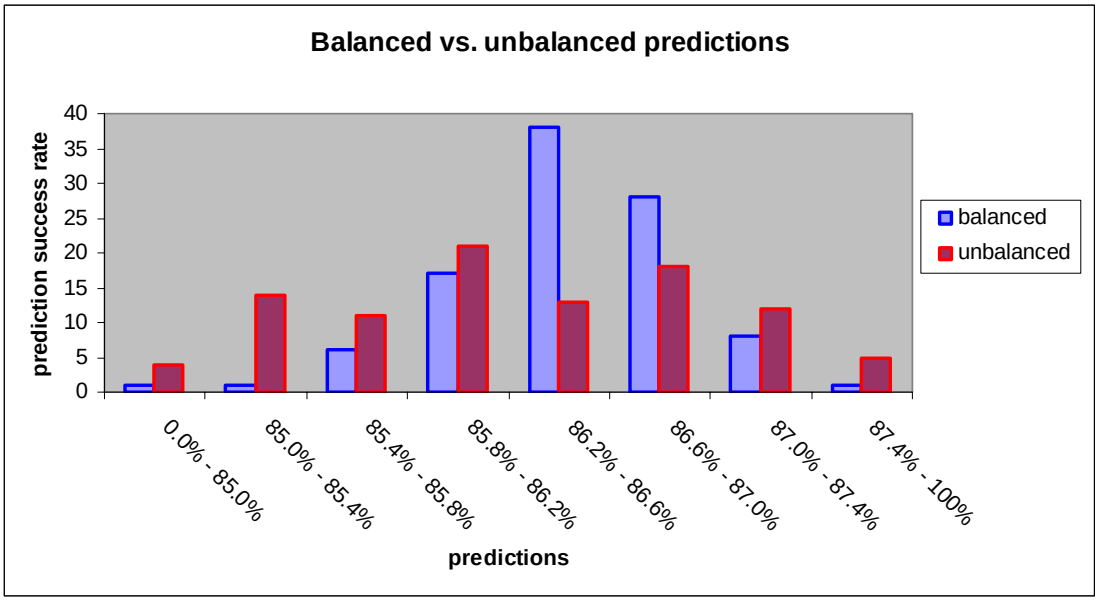

Figure 4: Unbalanced predictions vs. balanced cross-validation for the colon data set. By making sure every sample is included in the test set the same number of times, prediction results are less dependent on how many times the hard-cases are included in the test set.

Prediction rates on the cancer data sets estimated using LOOCV

We performed a comparative analysis using our MM based method as well as the Wilcoxon/DLDA method on the data sets summarized in Table 1. For the MM method we used both the class specific standard deviation (MMg) and the pooled standard deviation (MMp) in order to better understand the relative merits of the two in practical application. We performed LOOCV on the 6 datasets using the two MM based methods as well as DLDA using features selected using Wilcoxon. The results are summarized in Table 2. We see that prediction accuracy obtained using MMg and MMp are relatively similar, and both show equally good or superior prediction rates compared to the DLDA method. The LOOCV method has shown to lead to biased misclassification rates [23], but the results can be directly compared to prediction rates from other methods so we have included the comparison.

DataSet	MMp	MMg	DLDA
SRBCT	**98.4%**	**98.4%**	96.8%
Lymphoma	98.4%	**100.0%**	96.8%
Colon	87.1%	**90.3%**	87.1%
Leukemia	**98.6%**	97.2%	97.2%
Brain	**97.6%**	95.2	76.2%
Prostate	**92.2%**	**92.2%**	80.4%
Average	95.4%	**95.6%**	89.1%

Table 2: LOOCV prediction success rates on different datasets (column 1). The second column show the prediction accuracy obtained using the mixture model method with a pooled standard deviation. The third column shows the accuracy obtained using group standard deviation. The fourth column gives the accuracy obtained using DLDA. In all cases, 99 features (genes) are used. The best result for each dataset is marked in bold.

Prediction rates estimated using BLOTOCV

The six datasets (Table 1) were analyzed using BLOTOCV with 51 cycles so that each sample was included in the test set 17 times. The results obtained are summarized in Table 3. Table 3 includes the prediction rates obtained using our implementation of Wilcoxon/DLDA, our two MM methods, and for reference it also contains the Wilcoxon/DLDA results obtained by Dettling along with Dettling's BagBoost method. The results show that our method using mixture models are very competitive and in particular that the MMp is on average the best method among those tested on these data sets with the current parameters.

In the analysis we keep track of how many times each sample is misclassified. Based on this we identify hard cases. If a sample is misclassified often, this may be interpreted as an indication that the sample has been mislabeled. For instance, in the leukemia dataset, all samples except sample #5 were always correctly classified using either DLDA or the MM based methods. One of these (sample #67) is also classified incorrectly in the LOOCV procedure. The mixture model classified these samples incorrectly 14 and 16 times respectively which corresponds to approximately 87 and 100 percent of the classifications. Two of the incorrectly classified samples (samples #66 and #67) are also reported as hard cases in other studies ([11]) including the original publication ([9]). The fact that sample #66 is classified incorrectly almost every time in the BLOTOCV procedure, but correctly classified in the LOOCV procedure can be an indication of overfitting when too many of the samples are in the training set.

DataSet	MMp	MMg	DLDA	DDLDA	Bagboost
SRBCT	98.3%	96.6%	94.9%	91.1%	**98.8%**
Lymphoma	98.0%	**99.0%**	97.7%	94.7%	98.4%
Colon	**86.5%**	86.1%	83.2%	85.2%	83.9%
Leukemia	**97.0%**	95.6%	96.7%	**97.0%**	95.9%
Brain	**83.6%**	70.9%	70.7%	72.8%	76.1%
Prostate	86.9%	86.7%	80.74%	86.3%	**92.5%**
Average	**91.7%**	89.2%	87.3%	87.9%	90.9%

Table 3: Prediction rates using a balanced leave one third out cross validation compared to results given in Dettling et al. Columns two and three show prediction rates obtained using our mixture model based methods (pooled and group based standard deviation). The fourth column (DLDA) gives the results of our own implementation of DLDA. The fifth column (DDLDA) is Dettling's prediction rate for DLDA. The last column gives prediction rates for Dettling's Bagboost. The two last columns contain numbers from Dettling's paper. The best method for each dataset is marked in bold.

Another interesting case is the brain dataset group "primitive neuro-ectodermal tumors" (samples #36, #38 and #40). While 7 of these are classified correctly using the LOOCV, three of them are misclassified every validation cycle in the BLOTOCV. This can also be an indication of overfitting, but it is strange that only this class is affected and not any of the other classes. If class size is a success factor, the 4-sample class "human cerebella" should show lower score, which is not the case (none of these were incorrectly classified using LOOCV but one sample (# 30) was misclassified 9 times using BLOTOCV). By studying the feature sets we found only a few good features separating the "primitive neuro-ectodermal tumors" from the rest. We reduced the feature set size to remove bad features from the classifier and the success rate increased for all but two classes (#38 and #40) which now showed a 100% misclassification rate. This could be an indication of mislabeling of these two samples.

Finding irregular distributions

We can compare each mixture model f_i to a normal distribution estimated from the same observed expression values (having probability density function n_i), by plotting the two probability density functions together (e.g., Figure 5). We can measure the deviation between the two functions f_i and n_i by calculating the measure S_i substituting f_i for $f_{i,1}$ and n_i for $f_{i,2}$. The smaller the value for S_i, the larger deviation is between the two functions. To visually analyze the most irregular distributions, we sort the mixture models by their S_i values and present each mixture model as a graph. This can be used to interactively analyze whether a class might contain subclasses and may serve as a complement to performing clustering and class discovery. This may reveal irregular expression analogous to the COPA (cancer outlier profile analysis) analysis proposed by Tomlins et. al [24].

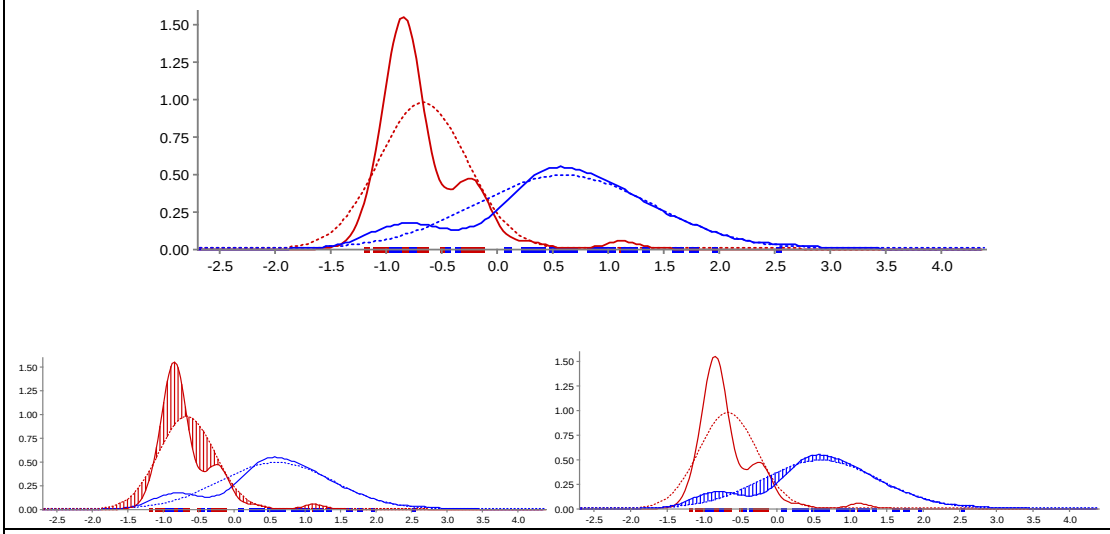

Figure 5: Finding irregular distributions: top: A gene from the prostate datset (row #5016). Solid lines are the two mixture models (MMg) and dotted lines represent the standard normal distribution. Lower left: the difference between red group (mean = 2.0, stdev = 2.0) mixture model and normal distribution. Lower right: normal distribution deviation for the blue group (mean = 0.0, stdev = 1.0)

DISCUSSION

We have shown that classification using a mixture model can outperform popular methods such as DLDA. The success rate of our proposed method (and other methods) does however depend on parameters such as the number of features used and the number of hard-cases in the dataset. The number of features has a significant effect on the error rate of both mixture models and DLDA. Further analysis is required to explore the relative merits of these and other methods on a representative set of data sets using a variety of parameter values. We can see that on datasets with few hard-cases (such as lymphoma and leukemia) both mixture models and DLDA have low error rates which is also consistent with other classifier studies.

The relatively high success rates using mixture models compared to the success rate using models assuming normal distribution of gene expression (such as DLDA), indicates that our assumption that normal distributions may sometimes be inappropriate and lead to sub-optimal prediction rates, is correct. Further studies should compare the distributions from features chosen by mixture model scoring and methods such as t-score to verify this.

The presence of (unkown) sub-classes with distinct genotypic features may lead to distributions that cannot be well described using normal distributions. Such cases can arise in different settings, and in particular for cancer such cases are known to exist [24]. Further studies should compare mixture model classification to other classification methods for different data types.

Presently we let the mixture model contain one normal distribution per observed data point. This alleviates the need to search for an optimal (or at least appropriate) number of components and parameters for each of these. Such an analysis would be time

consuming. On the other hand reducing the number of components to a smaller number, would produce more compact and easily interpretable feature descriptors. Furthermore the number of components and how the data points fit to these, would be valuable information if one were to search for presence of sub-classes or in other ways explore patterns within the pre-defined classes. The present approach produces good results in the tests that we have performed. Alternative approaches should be explored, and analyses performed to find out in which cases the current approach is sufficient.

We see that success rates vary between our MMp and MMg implementation and none of them are superior for all datasets. This suggests that the choice of standard deviation is an important success factor. Further studies should explore the relations of success rates between the two variants and patterns in the dataset to see if it is possible to choose one of the methods prior to classification. It is also possible that a third way of estimating a standard deviation exist that will ensure a stable superior success rate. We also leave the search for such method for further studies.

The mixture model framework contains methods for finding features with discriminate power, visualization methods for exploring expression distribution, methods for determining class membership (including certainty in form of misclassification rate) and cross validation functionality (LOOCV and LOTOCV). This framework is now included in the J-Express analysis tool [25], but should also be included in more accessible tools such as R or as web-tools. This will however be left for future work.

Acknowledgments
The present study has been supported by the technology platforms for bioinformatics and for microarrays (Norwegian Microarray Consortium) funded by FUGE, the functional genomics programme of the Research Council of Norway.

BIBLIOGRAPHY

1. Brown M, Grundy W, Lin D, Christianini N, Sugnet C, Ares M, Haussler D: **Support vector machine classification of microarray gene expression data.** *Technical Report UCSC-CRL 99-09 University of California, Santa Cruz CA* 1999.
2. DeRisi JL, Iyer VR, Brown PO: **Exploring the metabolic and genetic control of gene expression on a genomic scale**. *Science* 1997, **278**(5338):680-686.
3. White KP, Rifkin SA, Hurban P, Hogness DS: **Microarray analysis of Drosophila development during metamorphosis**. *Science* 1999, **286**(5447):2179-2184.
4. Alizadeh AA, Eisen MB, Davis RE, Ma C, Lossos IS, Rosenwald A, Boldrick JC, Sabet H, Tran T, Yu X *et al*: **Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling**. *Nature* 2000, **403**(6769):503-511.
5. Alon U, Barkai N, Notterman DA, Gish K, Ybarra S, Mack D, Levine AJ: **Broad patterns of gene expression revealed by clustering analysis of**

tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci U S A* 1999, **96**(12):6745-6750.
6. Tusher VG, Tibshirani R, Chu G: **Significance analysis of microarrays applied to the ionizing radiation response**. *Proc Natl Acad Sci U S A* 2001, **98**(9):5116-5121.
7. Benjamini Y, Hochberg Y: **Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing** *Journal of the Royal Statistical Society Series B (Methodological)* 1995, **57**(1):289-300.
8. Li T, Zhang C, Ogihara M: **A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression**. *Bioinformatics* 2004, **20**(15):2429-2437.
9. Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA *et al*: **Molecular classification of cancer: class discovery and class prediction by gene expression monitoring**. *Science* 1999, **286**(5439):531-537.
10. van 't Veer LJ, Dai H, van de Vijver MJ, He YD, Hart AA, Mao M, Peterse HL, van der Kooy K, Marton MJ, Witteveen AT *et al*: **Gene expression profiling predicts clinical outcome of breast cancer**. *Nature* 2002, **415**(6871):530-536.
11. Dudoit S, Fridlyand J, Speed T: **Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data**. *Technical report 576, Mathematical Sciences Research Institute, Berkeley, CA* 2000.
12. Khan J, Wei JS, Ringner M, Saal LH, Ladanyi M, Westermann F, Berthold F, Schwab M, Antonescu CR, Peterson C *et al*: **Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks**. *Nat Med* 2001, **7**(6):673-679.
13. Radmacher MD, McShane LM, Simon R: **A paradigm for class prediction using gene expression profiles**. *J Comput Biol* 2002, **9**(3):505-511.
14. Singh D, Febbo PG, Ross K, Jackson DG, Manola J, Ladd C, Tamayo P, Renshaw AA, D'Amico AV, Richie JP *et al*: **Gene expression correlates of clinical prostate cancer behavior**. *Cancer Cell* 2002, **1**(2):203-209.
15. Kohavi R, John GH: **Wrappers for feature subset selection** *Artif Intell* 1997 **97**(1-2):273 - 324
16. Perou CM, Sorlie T, Eisen MB, van de Rijn M, Jeffrey SS, Rees CA, Pollack JR, Ross DT, Johnsen H, Akslen LA *et al*: **Molecular portraits of human breast tumours**. *Nature* 2000, **406**(6797):747-752.
17. McLachlan GJ, Bean RW, Peel D: **A mixture model-based approach to the clustering of microarray expression data**. *Bioinformatics* 2002, **18**(3):413-422.
18. Brown M, Hughey R, Krogh A, Mian IS, Sjolander K, Haussler D: **Using Dirichlet mixture priors to derive hidden Markov models for protein families**. *Proc Int Conf Intell Syst Mol Biol* 1993, **1**:47-55.
19. Pan W, Lin J, Le CT: **How many replicates of arrays are required to detect gene expression changes in microarray experiments? A mixture model approach**. *Genome Biol* 2002, **3**(5):research0022.
20. D'Agostino RB, Stephens MA: **Goodness-of-fit techniques**. New York: Marcel Dekker; 1986.
21. Dettling M: **BagBoosting for tumor classification with gene expression data**. *Bioinformatics* 2004, **20**(18):3583-3593.

22. Pomeroy SL, Tamayo P, Gaasenbeek M, Sturla LM, Angelo M, McLaughlin ME, Kim JY, Goumnerova LC, Black PM, Lau C *et al*: **Prediction of central nervous system embryonal tumour outcome based on gene expression**. *Nature* 2002, **415**(6870):436-442.
23. Wit E, McClure J: **Statistics for microarrays : design, analysis and inference**. Chichester: Wiley; 2004.
24. de Wildt RM, Mundy CR, Gorick BD, Tomlinson IM: **Antibody arrays for high-throughput screening of antibody-antigen interactions**. *Nat Biotechnol* 2000, **18**(9):989-994.
25. Dysvik B, Jonassen I: **J-Express: exploring gene expression data using Java**. *Bioinformatics* 2001, **17**(4):369-370.