# Simultaneous Gene Clustering and Subset Selection for Sample Classification via MDL

Rebecca Jörnsten[1]
Department of Statistics, Rutgers University
501 Hill Center, Piscataway, NJ 08854

Bin Yu*
Department of Statistics, UC Berkeley
367 Evans Hall, Berkeley, CA 94720

[1]To whom correspondence should be addressed.

## ABSTRACT

**Motivation:** The microarray technology allows for the simultaneous monitoring of thousands of gene expression for each sample. The high-dimensional gene expression data can be used to study similarities of genes' expression profiles across different samples, that is gene clustering. The gene clusters may be indicative of genetic pathways. Another important application is sample classification, based on all or selected gene expressions. The gene clustering and sample classification are often undertaken separately, or in a directional manner (one as aid for the other). However, such separation of the two tasks may occlude informative structure in the data. Here we present an algorithm for the simultaneous clustering of genes and subset selection of gene clusters for sample classification. We develop a new model selection criterion based on Rissanen's MDL (minimum description length) principle. For the first time, an MDL code length is given for both explanatory variables (genes) and response variables (sample class labels).

**Results:** We apply our algorithm for simultaneous gene clustering and subset selection for classification to 3 data sets: (1) the NCI60 data set of Ross *et al.* (2000), (2) the Acute Leukemia data set of Golub *et al.* (1999), (3) the Colon tumor vs. normal data set of Alon *et al.* (1999). Competitive or improved test error rates, compared with the best reported methods, are demonstrated on all 3 data sets. Furthermore, we also show that the model selection instability associated with single gene selection is much reduced by the selection of gene clusters. We comment on how the inclusion of a classification criterion affects the gene clustering, bringing out class informative structure in the data.

**Availability:** The methods presented in this paper have been implemented in the R language. The source code is available from the first author.



**Contact:** rebecka@stat.rutgers.edu,
binyu@stat.berkeley.edu


## INTRODUCTION

Gene clustering identifies groups of genes that exhibit similar expression profiles across samples. Examples of gene clustering methods that have been used are hierarchical clustering, k-means, partitioning around mediods (PAM), and the SVD based geneshaving of Hastie *et al.* (2000a). For sample classification, we build predictive models for the sample classes based on all or selected gene expressions. Many classification methods have been applied to gene expression data: some traditional methods such as discriminant analyses (DA), nearest neighbors (NN), and Classification and regression trees (CART): some novel and more complex methods such as support vector machines (SVM), boosting and bagging CART, and neural nets. On most data sets, the simple and complex methods perform near equally well, or poorly. In fact, a recent study of Dudoit *et al.* (2000) indicates that simple NN, or diagonal linear DA (DLDA) often result in the best test error rate performance. Gene expression data sets consist of few samples on which to build and validate models. It is possible that the use of more complex classifiers will be more justified and necessary as the number and size of data sets grow. Many classifiers show improved performance with variable or feature selection. In addition, parsimonious classification models are often easier to interpret. This motivates gene or variable subset selection. Gene clustering and sample classification with variable selection are in most cases treated as separate problems, or in a directional manner (one as an aid for the other). As an example of the first, Hong and Li (2001) cluster genes and use cluster representatives with "soft-max" sample classification, without variable selection. An example of the latter, the supervised gene shaving of Hastie *et al.* (2000a) uses the sample class information for improved gene clustering. Another example of a directional approach, is the supervised harvesting of expression trees of Hastie *et al.* (2000b). There, initial clustering is followed by the selection of cluster centroids for classification, but the clustering remains fixed.

Here, we view the clustering of genes, and the subsequent selection of clusters that discriminate between samples, as *one* model selection problem. We use Rissanen's minimum description length (MDL) principle, which formalizes Occam's razor. MDL stipulates that one should choose the model with the shortest description, or code length of the data. Most of the time, the code lengths take on the form of a penalized likelihood, where the penalty reflects the model complexity. By turning classification (linear discriminant analysis) into regression via *Optimal Scoring*, we form a combined code length for the gene clustering and sample classification. We use a two-part code, where the first part describes the gene clustering based on a Gaussian mixture model, and the second part describes the sample class labels conditioned on the gene clusters. We select the model that minimizes the *sum of the two parts* of the code length. Thus, the clustering is affected by the ability of the corresponding cluster centers to discriminate between the sample classes. Similarly, the selection of clusters for classification is affected by the clustering. For the first time, an MDL model selection criterion is given for both explanatory variables (genes), and response variables (class labels). Competitive or improved test error rates, compared with the best reported methods, are shown on the National Cancer Institute's anti-cancer drug-screen data (NCI60) of Ross *et al.* (2000), the Acute Leukemia data set of Golub *et al.* (1999), and the Colon data set of Alon *et al.* (1999). We also show that the model selection instability is reduced by selecting gene clusters, rather than individual genes. The simultaneous clustering and subset selection for classification results in gene clusters with class informative structure.

The paper is organized as follows. In the **Methods** section we briefly outline the MDL principle, optimal scoring, and the derivation of our simultaneous clustering and subset selection MDL criterion. Test error rates and model selection results are presented in the **Results** section for the 3 gene expression data sets. We conclude with a discussion of our method and ideas for extensions.

## METHODS

Rissanen's minimum description length principle formalizes Occam's razor, and states that one should choose the model that gives the shortest description, or code length of the data. In the MDL framework, a model is meant to aid in the data coding or transmission. The basic idea is that though a complex model reduces the data coding cost, given the model, the *total* code length required to describe the *model and the data* may be smaller if we use a simple model. MDL model selection is thus trading off model complexity with model goodness-of-fit, based on the model's data coding efficiency. There are many ways of constructing code lengths for a data string $x^n$. Here we restrict the discussion to parametric models, indexed by parameter vectors $\theta$. A code length based on a mixture distribution on the members of the model family defined $\theta$, is called a mixture code length, and is given by $CL(x^n) = -\log \int f(x^n|\theta)w(\theta)d\theta$. We have a closed form expression for the mixture code length if $f(x^n|\theta)$ is an exponential family, and $w(\theta)$ is a corresponding conjugate prior. Comparing two models using the mixture code length is effectively looking at Bayes factors. To the first order, we can approximate the mixture code length by a two-stage code. We first encode the estimated model parameters $\hat{\theta}$ at precision $1/\sqrt{n}$, and then the data given the estimated model. The two-stage code length is given by $-\log f(x^n|\hat{\theta}) - dim(\theta)\log 1/\sqrt{n} = -\log f(x^n|\hat{\theta}) + dim(\theta)\log n/2$, which we recognize as BIC. Alternatively, via a Laplace expansion of $CL(x^n)$ we get Rissanen's Stochastic Information Complexity $SIC(x^n) = -\log f(x^n|\hat{\theta}) + \frac{1}{2}\log|nI(\hat{\theta})|$, with $I(\hat{\theta})$ equal to the Fisher information matrix. $SIC$ can be interpreted as a two stage code, with a parameter value dependent precision $SE(\hat{\theta}_{mle,n}) = 1/\sqrt{nI_{jj}(\theta)}$.

Using coding arguments as a safeguard against overfitting in clustering was first introduced in 1968 by Wallace and Boulton. Their *Minimum Message Length* (MML) information measure is interpreted from a strictly Bayesian point of view. The MDL and MML concepts are similar, and the differences are philosophical rather than technical. The commonly used MDL (or MML) approach to clustering selects the number of clusters $K$ in a data set using an SIC two-stage code (Wallace and Boulton (1994)). At the first stage, we describe the $K$ cluster models, the cluster sizes, and cluster memberships. At the second stage, we encode for each datum its deviance from the cluster model. For cluster $k \in \{1, \cdots, K\}$, the $k$th cluster model is defined by 3 parameters $\theta_k$: the centroid $\mu_k$, the standard deviation $\sigma_k$ of independent additive mean zero Gaussian errors, and the cluster proportion $p_k$ (mixture probability). Uninformative priors $w$ are assigned to all parameters. The SIC approximation of the MML code length is $MML = -\log w(\theta) + \frac{1}{2}\log|I(\theta)| - \log f(x^n|\theta)$, where $f(x^n|\theta)$ is a Gaussian mixture likelihood and $w(\theta)$ the prior. We have $\frac{1}{2}\log|I(\theta)| = \sum_{k=1}^{K} \frac{\sqrt{2}n_k}{\sigma_k^2}$, where $n_k$ are the number of observations in cluster $k$. The deviances from the cluster models are $r_i = \sum_{k=1}^{K} 1\{i \in b(k)\}\ (x_i - \mu_k)$, $i = 1, \cdot, n$, where $b(k)$ denotes the cluster membership of cluster $k$. If $b(k)$ is empty, or the cluster variance is 0 ($\hat{\sigma}_k^2 = 0$), no additional code is needed for $r_i$. The MML code length is given by

$$MML(K) = K\log(2\sigma^2) - \log(K-1)! + \sum_{k=1}^{K} \frac{\sqrt{2}n_k}{\sigma_k^2}$$
$$-\sum_{i=1}^{n}\sum_{k=1}^{K} 1\{i \in b(k),\ |b(k)| > 0,\ \hat{\sigma}_k^2 > 0\}\ \log\phi(r_i|0, \hat{\sigma}_k^2),$$

and we select $K$ that minimizes $MML(K)$. The extension to $D$-dimensional data is straight forward.

Model selection in linear regression identifies a subset of columns of the predictor matrix $X$, associated with the response variable $y$. The regression model is written as $y = X_\gamma\beta_\gamma + \epsilon$, $\epsilon \sim N(0, \sigma^2 I)$, where $\beta_\gamma$ is the coefficient vector. The indicator variable $\gamma$ denotes which variables (columns of $X$) are included in the model. The two-stage MDL code length BIC takes on the familiar form $BIC_\gamma = \frac{n}{2}\log(RSS_\gamma) + \frac{k_\gamma}{2}\log n$, where $\sum 1\{\gamma = 1\} = k_\gamma$, and $RSS_\gamma = ||y - X_\gamma\hat{\beta}_\gamma||^2$ is the residual sum of squares with $\hat{\beta}_\gamma = (X_\gamma^T X_\gamma)^{-1}X_\gamma^T y$. BIC thus penalizes all models of size $k_\gamma$ equally. Hansen and Yu (2001) introduced the gMDL criterion, which includes an *adaptive penalty* on the model size, that is a penalty that reflects the goodness-of-fit of the model. The gMDL is a mixture MDL code length derived as follows. For convenience we drop the subscript $\gamma$ from the notation, and denote the variance by $\tau = \sigma^2$. The mixture code length for response variable $y$, conditional on $X$, is $CL(y|X) = -\log\int f(y|X,\beta,\tau)w(\beta,\tau)d\beta d\tau$. The conjugate family of priors is the Normal-Inverse-Gamma, where we often use

$$\beta|\tau \sim N(0, \tau c\Sigma),\ \tau \sim \frac{(a/2)^{1/2}}{\Gamma(a/2)}\tau^{-1}\exp(-\frac{a}{2\tau}).$$

Here, $a$ and $c$ are hyper parameters, and $\Sigma = (X'X)^{-1}$ is the $k \times k$ prior scaled covariance matrix for $\beta|\tau$. The integration with respect to prior $w(\beta, \tau)$ is standard. In a strict Bayesian approach, we place hyperpriors on $a$ and $c$. In the derivation of gMDL, Hansen and Yu take an *empirical Bayes* approach, and estimate the parameters $a$ and $c$ by maximizing the marginal likelihood. Hansen and Yu's *gMDL* criterion equals

$$gMDL = \frac{n}{2}\log(RSS) + \frac{k}{2}log\Big((\frac{R^2}{1-R^2})/(\frac{k/n}{1-k/n})\Big) + \frac{n}{2}log(\frac{1}{1-k/n}) + log(n),$$

if $R^2 \geq k/n$, and $\frac{n}{2}\log(y^Ty) + \frac{1}{2}\log n$ otherwise. Using $gMDL$, the penalty thus depends on the fit of the model through the multiple r-squared $R^2$, as well as the model size. gMDL implicitly compares models based on the coefficient estimates $\hat{\beta}_{gMDL}$, where

$$\hat{\beta}_{gMDL} = \max(1 - \Big(\frac{1-k/n}{k/n}\Big)\Big(\frac{1-R^2}{R^2}\Big), 0)\hat{\beta}_{LS}.$$

gMDL thus generates a form of shrinkage estimates.

**Optimal scoring and MDL in classification**

Model selection criteria in classification that are based on 0-1 loss are often intractable. It is therefore common to use an L2 loss approximation for selection. Here, we thus take the route of turning classification into regression via optimal scoring. We then appeal to the existing MDL methodology in regression to construct an MDL model selection criterion for classification. We restrict our attention to linear discriminant analysis (LDA) here. Let $X$ be the matrix of feature vectors. In LDA we find discriminant functions $\mu$ such that the linear combinations of feature vectors (projections onto the space spanned by the discriminant functions) $X\mu$ have maximal Between-to-Within-class sum of squares. LDA is thus an eigen value problem. Classification of a new observation is done in the subspace spanned by $\mu$ by assigning an observation to the closest class mean (euclidean distance) in the subspace. Hastie *et al.* (1994) discuss the familiar equivalence of LDA and regression via optimal scoring. The point of optimal scoring is to turn a categorical problem into a quantitative one. We find the discriminant functions $\mu$ via linear regression, instead of solving the eigen value problem, by forming a $n \times C$ dummy matrix $Y$, and regressing it on $X$. The $j$th column of $Y$ has $i$th row entry "1" if sample $i$ belongs to class $j \in \{1, \cdots, C\}$, and "0" otherwise. The simultaneous estimation of the optimal scores, and the regression coefficients can be stated as $\min_{\theta,B} ||Y\theta - XB||^2$, with constraint $\theta^T D_p \theta = I$, where $D_p$ is a diagonal matrix of class proportions. Given $\theta$, the minimizing $B$ are the least squares coefficients such that

$$\min_{\theta,B} ||Y\theta - XB||^2 = \min_{\theta} Tr\{(\theta^*)^T\theta^* - \theta^T(Y^T\hat{Y})\theta\}.$$

From this follows that a third alternative to finding $\mu$ is to minimize $||\theta^* - XB^*||^2$, i.e. regress the orthogonal *optimal scores* $\theta^* = Y\theta$ on X. The regression coefficient estimates $B^*$ are proportional to the discriminant functions $\mu$ such that the $l$th discriminant function

$$\mu_l = \frac{1}{\sqrt{R_l^2(1-R_l^2)}}B_l^*, R_l^2 = 1 - RSS_l(\theta_l^*, X)/((\theta_l^*)^T\theta_l^*),$$

where $R_l^2$ is the multiple r-squared of the $l$th column of $\theta^*$ regressed on $X$. The model selection (selecting columns of $X$) aspect of discriminant analysis is more complex. If the selection criterion is a function of the individual $RSS_l$, the selection is affected by choice of initial optimal scores $\theta^*$, themselves functions of $X$. It is then recommended to iterate between selection, and updating of the optimal scores using the selected columns of $X$.

We construct an MDL model selection criterion for classification by applying an extension of $gMDL$ to multi-response regression of the orthogonal optimal scores. Since the scores are orthogonal, the closest Gaussian approximation consists of independent response models. For a $C$ class problem, there are $C-1$ non-trivial optimal scores $\theta^*$. We write $gMDL_C = \sum_{l=1}^{C-1} gMDL(\theta_l^*|X)$, where

$$gMDL(\theta_l^*|X) = \frac{n}{2(C-1)}\log(\sum_l^{C-1} RSS_l) + \frac{k}{2}log\Big(\frac{(\frac{R_l^2}{1-R_l^2})}{(\frac{k/n}{1-k/n})}\Big) + \frac{n}{2}log(\frac{1}{1-k/n}) + log(n),$$

if $R_l^2 \geq k/n$ and $\frac{n}{2}\log((\theta_l^*)^T\theta_l^*) + \frac{1}{2}\log n$ otherwise. The derivation of $gMLD_C$ requires a lengthy and delicate discussion of the choice and form of model priors, and is omitted here to conserve space. The details can be found in Jornsten (2001). The shrinkage estimates of the discriminant functions equal

$$\hat{\mu}^{(l)} = \frac{\hat{\beta}_{LS}^{(l)}}{\sqrt{R_l^2(1-R_l^2)}}\ 1\Big\{R_l^2 \geq k/n\Big\},$$

where $\hat{\beta}$ are the optimal score regression coefficient. For some $l$, the estimates $\hat{\mu}^{(l)}$ may equal zero, and those discriminant functions are thus dropped from the model. Reducing the dimension in which we classify in this fashion, often improves the class predictive performance of the selected models. The optimal scoring turned the categorical problem into a quantitative one, but the response variables are nevertheless discrete. A Gaussian coding scheme may not be the most effective, but we hope that the coding redundancy induced by the approximation is independent of the model size and complexity. Then, a relative comparison of the $gMDL_C$ code lengths *between* models is still valid. In practice, we find that the Gaussian based selection criterion performs well on simulated and real data (RESULTS section, and Jornsten (2001)).

**Simultaneous clustering and classification**

When the predictor variables are correlated, model selection in regression is difficult. The instability of such selection problems is well-known. We consider the case when the number of predictors $p$ is very large compared to the sample size $n$. We approach the model selection by clustering the predictors, and selecting cluster centroids for response variable prediction. If a predictor cluster model describes the data well, the cluster centroids are less correlated than the individual predictors, which reduces the instability of selection. When $p \gg n$, and the predictors highly correlated, using cluster centroids as new predictors is essentially equivalent to model averaging, with significant improved response predictive performance over individually selected models (Jornsten (2001)).

MDL model selection in regression assumes that the predictors are known at the decoder. For the purpose of simultaneous clustering of the predictors and subset selection of cluster centroids for response variable prediction, we now assume that the predictors are not known. In this case, the predictors have to be transmitted to the decoder, prior to transmitting the response. We use the MDL clustering code length to encode the predictors. For now, fix the number of predictor clusters to $K$. Given the clustering structure of the predictors, we encode the response using the $gMDL_C$ regression criterion and optimal scoring, where the explanatories in the regression model are now the $K$ cluster centroids. The *joint coding* of $(\theta^*, X)$ is thus a simultaneous selection of a $K$ cluster model for the predictors $X$, and the selection of cluster centroids in the regression model for $\theta^*$. The joint code length has two parts. The first part $CL(X)$ deals with the cluster structure of the predictors. The second part describes the response, given the predictor cluster structure. We call this component $CL(\theta^*|X^*)$, where $X^*$ is a $n \times K$ cluster centroid matrix $(X^*)_{ik} = \frac{1}{|b(k)|}\sum_{j=1}^{p} 1\{x_{ij} \in b(k)\}x_{ij}$, $b(k)$ are the predictors in cluster $k$. The code lengths $CL(X)$ and $CL(\theta^*|X^*)$ have very different magnitudes. $CL(X)$ is a code length of $n \times p$ "observations" $X$. In contrast, only $n \times (C-1)$ observations $\theta^*$ contribute to $CL(\theta^*|X^*)$. We want the two code lengths to have the same relative importance if we increase the sample size, or include more predictors in the selection. It is therefore natural to consider the coding *rates*, and select the model that minimizes $CL(\theta^*, X) = \frac{CL(X)}{np} + \frac{CL(\theta^*|X^*)}{n(C-1)}$. By minimizing the joint coding rate the gene clustering is affected by the inclusion of a classification model selection criterion.

It is often better to reduce the dimension of $X$ $(n \times p)$ prior to clustering. Clustering high-dimensional data is difficult, and we tend to underestimate the number of clusters if we cluster in a higher dimension than necessary. For supervised gene clustering, we have outcome variables $y$ that are categorical. If the within class variation is sufficiently small, instead of clustering $X$, we cluster the reduced dimension matrix $\bar{X}$, $(\bar{X})_{cj} = \frac{1}{n_c}\sum_{i=1}^{n} x_{ij}1\{y_i = c\}$, $c = 1, \cdot, C$, $j = 1, \cdot, p$, where $n_c$ is the number of samples in class $c$. If $X$ consists of $p$ "observations" from a C-dimensional cluster model, the residuals $r_{ij} = x_{ij} - \bar{x}_{cj}$ are clearly independent of the clustering, and can be transmitted at fixed cost. This strategy is similar, though motivated differently, to the supervised geneshaving of Hastie *et al.* (2000a).

We now describe the simultaneous gene clustering and subset selection algorithm, where $K$ is allowed to vary. We perform an initial selection of genes, removing genes with near constant variation across classes. We compute the between-to-within-class sum of squares (B/W), and consider only the $T$ largest B/W genes in our models.

**Algorithm**

1. Select the $T$ largest B/W genes. Use the $S \leq T$ largest B/W genes to compute the initial optimal scores $\theta^*$. Set $q = 0$.
2. Compute $\bar{X}$, dimension $C \times p$ ($\bar{X}$ is a matrix of gene class means), $(\bar{X})_{cj} = \frac{1}{n_c}\sum_{i=1}^{n} x_{ij}1\{y_i = c\}$.
3. Select an initial number of clusters, $K_q, q = 0$.
4. Using k-means, generate a $K_q$ clustering of $\bar{X}$. Compute the clustering code length $CL(\bar{X}|K_q)$. Denote the clustering assignment by $b_q(k), k = 1, \cdot, K_q$.
5. Compute $X^*(K_q)$, dimension $n \times K_q$, the matrix of cluster centroids $(X^*(K_q))_{ik} = \frac{1}{|b_q(k)|}\sum_{j=1}^{p} x_{ij}1\{j \in b_q(k)\}$, $i = 1, \cdot, n$, $k = 1, \cdot, K_q$. Select $k_q^*$ centroids that minimize the code length $CL(\theta^*|X^*(K_q))$.
6. Iterate steps 4 and 5 $B(K_q)$ times, and choose the $K_q$ model with the smallest total coding rate $\frac{CL(\bar{X}|K_q)}{Cp} + \frac{CL(\theta^*|X^*(K_q))}{(C-1)n}$.
7. Go to step 4, set $K_q = K_q + 1$, and $q = q + 1$. If $q = q_{stop}$ then stop, and select the $q^0$ that minimizes $\frac{CL(\bar{X}|K_q)}{Cp} + \frac{CL(\theta^*|X^*(K_q))}{(C-1)n}$.
8. If necessary, recompute $\theta^*$ using the selected model, and rerun the algorithm from step 3.
9. The final model for is a $K_{q^0}$ gene clustering, selected $k_{q^0}$ cluster centroids, and a set $C_{q_0} \leq (C-1)$ discriminant functions.

Here, we perform a direct search over different cluster sizes $K_q$, and adjust for the different sizes of the model classes for each $K_q$ with the iteration factor $B(K_q)$.

**Validation and Comparison of Methods**

To estimate the test error rate of selected models we use threefold crossvalidation. Each crossvalidation training set is formed by randomly selecting two thirds of the samples. This training set is used for initial gene selection, model fitting and selection. The withheld third of the full data set is used only for estimating the test error. We ensure that the crossvalidation training and test sets mimic the full data set, by resampling according to the class proportions in the full data set. Threefold crossvalidation is more appropriate than traditional leave-one-out crossvalidation when the number of samples is as small as in the gene expression data sets. Leave-one-out crossvalidation with few samples often leads to overfitting and underestimation of the true error rate.

In Dudoit *et al.* (2000), a comparative study of classifiers was conducted. The simple nearest neighbor (NN) and diagonal linear discriminant (DLDA) methods were found to give the best test error rate results on several gene expression data sets (including NCI60, and Acute Leukemia). We implement NN and DLDA here also. DLDA uses all $T$ genes. NN is based on the $k$ nearest neighbors, where $k$ is selected by leave-one-out crossvalidation on the training set. We also implement the supervised geneshaving of Hastie *et al.* (2000a). This method resembles our approach, if the clustering is done *separately* from the subset selection. We follow Hastie *et al.* and generate 8 geneshaving clusters, with sizes chosen by the gap-statistic. Since the geneshaving clusters are (almost) orthogonal, a simple LDA classifiers is used in conjunction with supervised geneshaving.

## RESULTS

We apply our algorithm for simultaneous clustering and subset selection for classification to 3 gene expression data sets. For all data sets we perform threefold crossvalidation 150 times to estimate the test error rate of the selected models, and to investigate the stability of model selection. We also run our algorithm on the full data sets, and discuss the selected models and gene clusters. For each training data set, an initial gene selection removes genes that have near constant variation across samples. For all 3 data sets there is a sharp drop-off in the between-to-within-class (B/W) sum of squares after 10-15 genes. We select the T=200 largest B/W genes as a starting point, and use the S=10 largest B/W genes to compute the initial optimal scores. We run the algorithm with the number of clusters ranging from $K_q$=1 to 20. For each $K_q$, we iterate $B(K_q)$ times, where $B(K_q)$=50 for $K_q \leq 10$, $B(K_q)$=100 for $K_q$=11 to 20. The $B(K_q)$ were selected as reasonable trade-off between sufficient sampling of the model space, and computing time. We allow for up to 20 clusters or individual genes to be selected in the optimal scoring regression model, and perform exhaustive search over all possible subsets.

Model selection may be affected by the choice of initial optimal scores (the $S$ genes). To avoid the computationally intensive exercise of iterating the algorithm, we try to pick a reasonable $S$ such that iteration is not necessary. On the NCI60 data we run the algorithm twice, starting with $S$=30 genes to compute the initial scores. The selected model is identical to the one we get with $S$=10, except that one additional cluster of 5 genes is selected for classification. A second iteration of the algorithm, with updated optimal scores, generates the model we get with $S$=10. If we start with $S$=40 or $S$=50 genes, the optimal scores are "noisy" and a large model is selected after the first run of the algorithm. The updated optimal scores are therefore also noisy, and iteration does not help. Reiterating the algorithm with $S$=10 as a starting point gives back the same model. We conclude that $S$=10 limits the need for iterating the algorithm.

The simultaneous clustering and subset selection algorithm generates "active" and "inactive" clusters. Active clusters are those that are selected in the classification model. For convenience, in this section we will refer to our algorithm as SimClust (simultaneous clustering and subset selection). Separate clustering followed by subset selection is referred to as SepClust, individual gene selection as IndSelect, and geneshaving as GS.

**NCI60 data**

The National Cancer Institute's anti-cancer drug-screen data (NCI60) of Ross *et al.* (2000) consists of n=61 samples from human cancer cell lines. Gene expression levels were measured for p=9703 genes. The samples were grouped into 8 classes according to tumor site: 9 breast, 5 central nervous system (CNS), 7 colon, 8 leukemia, 8 melanoma, 9 non-small-lung-carcinoma (NSCLS), 6 ovarian and 9 renal. Genes that had more than 2 missing values were screened out. The remaining missing values were imputed by the mean value of 5 nearest neighbor genes. The retained data matrix is thus of dimension $61 \times 5244$. We run our algorithm on the full data set. The SimClust selected model consist of 8 gene clusters, 6 of which are "active", and 5 discriminant variables (out of 7). The active clusters contain 145 genes and cluster sizes range from 4 genes to 47 genes. The training error rate is 11.1 %. With SepClust we get 5 gene clusters, 4 of which are active. The cluster sizes range from 12 to 78, and the active gene clusters contain 195 genes. The training error is 19.7 %. DLDA and GS also have training error 11.1 %, and NN has training error 14.8 %. In NN 2 nearest neighbors were selected. The GS clusters range

*Fig 1* in size from 2 to 8, containing 40 genes. In Figure 1 we show the difference between SepClust and Simclust gene clusters. The left plot shows the samples (columns) and gene clusters (rows) generated by SepClust. It is difficult to see clear block patterns on this noisy data set. The inactive cluster is heterogeneous within sample classes, and is dropped in the subset selection. The plot to the right show the results using SimClust. The active clusters have been split into new active and inactive clusters. The new inactive clusters are noisy and heterogeneous, but the new active clusters are much cleaner than those generated by SepClust.

*Fig 2* In Figure 2 the crossvalidation test error rates are shown, comparing SimClust, SepClust and IndSelect (top row), to NN, GS, and DLDA. The SimClust test error rates are comparable to NN and DLDA, with the added benefit of gene cluster information. SepClust and GS perform worse, and IndSelect poorly. In Table 1 the average (over crossvalidation training sets) selected models are shown. The models selected by SimClust are on average larger than the ones selected by SepClust, but the SimClust models contain fewer genes. On average SimClust selects 9.4 gene clusters, of which 6.9 are active. Selecting clusters of genes instead of individual genes reduce the instability in model selection. The standard deviation of SimClust selected number of clusters, and active clusters are .6 and .4 respectively. The IndSelect scheme selects on average 9.3 genes, with a standard deviation of 2.0.

**Acute Leukemia data, 3 classes**

The leukemia data set of Golub *et al* (1999) consists of 25 AML (acute myeloid leukemia) and 47 ALL (acute lymphoblastic leukemia) samples. The ALL samples were subdivided into 38 B-cell ALL and 9 T-cell ALL. The gene expressions were measured using high-density oligonucleotide arrays with $p$ =6817 human genes. After excluding genes with $max/min < 5$ and $max - min < 500$ (across samples) the retained data matrix is of dimension $72 \times 3571$. We run our algorithm on the full data set. SimClust results in 7 gene clusters, 3 of which are active and contain 100 genes. The cluster sizes range from 14 to 51. SepClust generates 5 gene clusters, 3 active containing 144 genes. The cluster sizes range from 22 to 53. The training error rate is 1.4 % with either scheme (one error, sample # 67). NN also results in one error, whereas GS and
*Fig 3* DLDA have a training error of 0. In Figure 3 we compare the clusters generated by SepClust (left plot) and SimClust (right plot). The picture is clearer on the Leukemia data than NCI60. The active clusters generated by SepClust, though inhomogeneous, consist of genes that are upregulated in one sample class only. The inactive clusters contain genes that are upregulated in multiple sample classes, to varying degree. In the right plot SimClust has split the active clusters into new homogeneous clusters. The active clusters are now clearly comprised of genes that are upregulated only within a given sample class. The inactive clusters are mixed, consisting of genes that are upregulated to a lesser degree in multiple classes.

In Figure 4 the crossvalidation test error rates are *Fig 4*
shown for the 3-class leukemia data set. SimClust performs as well as NN, but with added gene clustering information. Sepclust is on par with DLDA, whereas IndSelect and GS perform worse. In Table 2 the selected models are compared. SimClust generates on average 6.8 gene clusters, 3.0 active consisting of 101.8 genes. The standard deviation of active and number of clusters are .3 and .7 respectively. SepClust results in on average fewer, but bigger clusters. IndSelect selects on average 7.3 genes, with standard deviation 1.5.

**Acute Leukemia data, 2 classes**

We now consider the Acute Leukemia data as a two class problem, not making any distinction between the ALL-B and ALL-T subclasses. SimClust generates 4 clusters on the full data set, with 2 active clusters consisting of 76 genes. SepClust generates 4 clusters, with 2 active clusters consisting of 71 genes. The simultaneous approach has little impact in this two-class problem. The active clusters contain largely the same set of genes. The training error is again 1.4 %, which is also the case with NN and GS. DLDA results in an error rate of 2.8 % in this case.

In Figure 5 the crossvalidation test error rates are *Fig 5*
shown. SimClust now outperforms NN and DLDA, shifting the mode of the error rate distribution to 0 as opposed to 1. SepClust is comparable to SimClust, GS performs worse than NN and DLDA, and IndSelect worse still. In the bottom portion of Table 2, the selected models are shown. The SimClust and SepClust methods perform similarly, and are both more stable than IndSelect.

**Colon data**

Alon *et al* (1999) used high-density oligonucleotide arrays to study gene expression patterns in tumor and normal colon tissues. The expression levels of $p$ =6500 genes were measures in $n$ =62 samples, 40 tumor and 22 normal colon tissue samples. The 2000 genes with the highest minimal intensity across samples were selected for further study. Using SimClust on the full data, 4 gene clusters are generated, 3 active containing 35 genes. The training error is 11.9 % (7 errors). SepClust results in 3 clusters, 2 active containing 193 genes. The training error is 24.2 %. NN and DLDA have training errors 12.9 %, whereas GS has training error 9.7 %.

In Figure 6 the crossvalidation test error rates are

*Fig 6* shown. SimClust outperforms NN and DLDA, shifting the mode of the error distribution toward zero. SepClust and IndSelect perform worse, with comparable results obtained with GS. In Table 3 the model sizes are shown. SimClust generates on average 4.1 gene clusters, with 2.4 active consisting of 70.6 genes. SepClust generates fewer, and bigger clusters. IndSelect selects on average 4.2 genes, with standard deviation 2.2. The standard deviations of the number of clusters and active clusters of SimClust are .6 and .4 respectively, indicating that clustering reduces the instability of model selection.

## CONCLUSIONS AND DISCUSSION

We present a new MDL model selection criterion for the simultaneous clustering of genes and subset selection of gene clusters for sample classification. Our method gives competitive or improved test error rate results, compared to some of the best methods reported in the literature, on 3 data sets. In addition, we show that the test error rates and model selection instability are much reduced by selecting gene clusters, instead of individual genes. We show that the gene clusters generated by our algorithm, including a classification criterion, are different than clusters that are generated in a supervised, but separate and directional manner. The simultaneous clustering and subset selection algorithm generates more homogeneous gene clusters, and separates "active", clusters that are class predictive, from "inactive" clusters, that exhibit cross-sample variation but are not necessarily good class predictors.

If the sample class labels are unknown, a two-way simultaneous clustering approach is needed. We can extend our method to the simultaneous gene and sample clustering. Here, we based the gene clustering on gene expressions in $C$ dimensions, where $C$ is the number of sample classes. When the class labels are unknown, two-way clustering entails the *selection of the dimension* in which to cluster. The residual data matrix, from the $C$-dimensional average matrix, is no longer independent of $C$ or the clustering if the class labels are unknown. Thus, simultaneous clustering is obtained via the inclusion of a code length for the residual matrix.

We used linear discriminant analysis with our gene clustering method in this paper, but any postprocessing classifier can be applied to the selected models (Jornsten (2001)). On these data sets, the selected models are small and linear discriminant analysis performs well. To make the MDL treatment of simultaneous clustering and classification complete, we need to extend MDL variable selection to classification techniques other than discriminant analysis. This remains an open problem. Two-stage code lengths can be found for penalized discriminant analysis via penalized optimal scoring. However, the derivation of mixture MDL code lengths for generalized linear models, like logistic regression, requires development of new methodology.

## ACKNOWLEDGEMENTS

This research is partially supported by grants from NSF (FD98-02314, DMS-9803063, FD01-12731).

## FIGURES

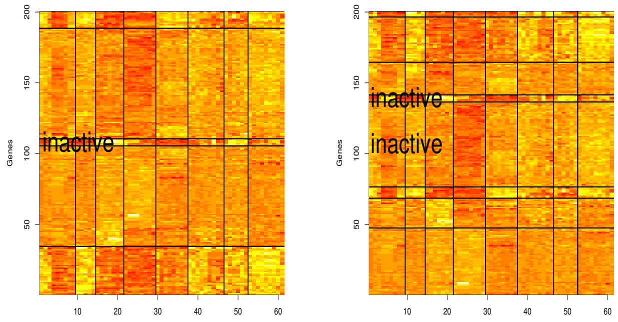

Figure 1:

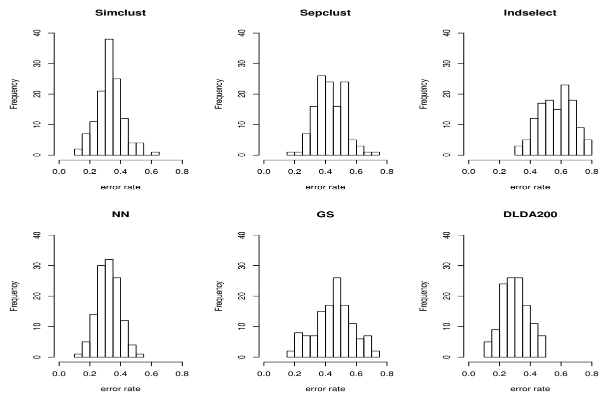

Figure 2:

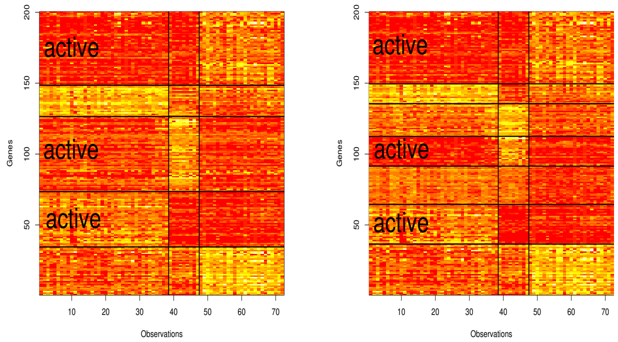

Figure 3:

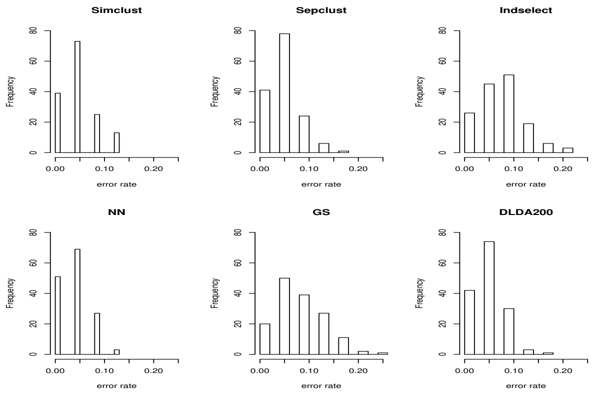

Figure 4:

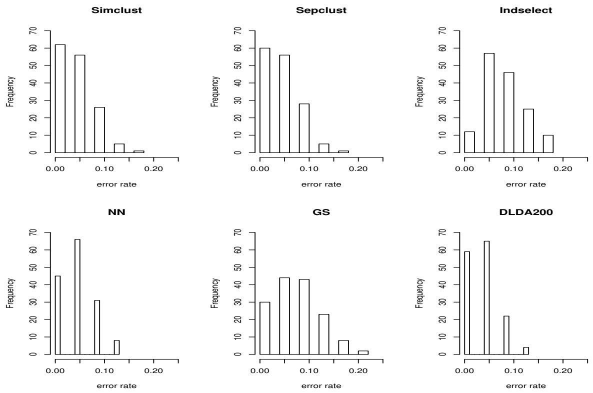

Figure 5:

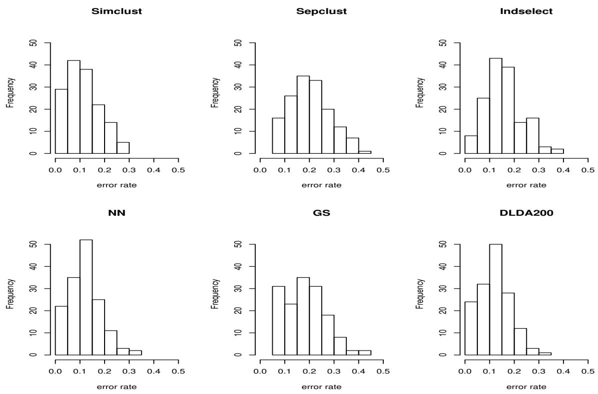

Figure 6:

## REFERENCES

Alon, U., Barkai, N., Notterdam, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J. (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci*, **96**:6745-6750

Dudoit, S., Fridlyand, J., Speed, T. (2000) Comparison of discrimination methods for the classification of tumors using gene expression data. *Technical report, Department of Statistics, UC Berkeley*

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286**: 531-537

Jörnsten, R. (2001) Data compression and its statistical implications, with and application to the analysis of microarray images. *PhD thesis, Department of Statistics, UC Berkeley*

Hansen, M., Yu, B. (2001) Model selection and the principle of minimum description length. *JASA* **96**(454): 746-774

Hastie, T., Tibshirani, R., Eisen, M., Brown, P., Ross, D., Scherf, U., Weinstein, J., Alizadeh, A., Staudt, L., Botstein, D. (2000) Gene Shaving: a new class of clustering methods for expression arrays. *Technical report, Department of Statistics, Stanford University.*

Hastie, T., Tibshirani, R., Botstein, D., Brown, P. (2000) Supervised harvesting of expression trees. *Technical report, Department of Statistics, Stanford University*

Hastie, T., Tibshirani, R., Buja, A. (1994) Flexible discriminant analysis. *JASA* **89** 1255:1270

Li, H.,Hong, F. (2001) Cluster-rasch model for microarray gene expression data. *Genome biology* **2**(8-31):1-13

Rissanen, J. (1989) Stochastic complexity *Singapore world scientific*

Ross, D.T., Scherf, U., Eisen, M., Perou, C.M., Spellman, P., Iyer, V., Jeffrey, S.S., Van de Rijn, M., Waltham, M., Pergamenschikov, A., Lee, J.C.F., Lashkari, D., Shalon, D., Myers, T.G., Weinstein, J.N., Botstein, D., Brown, P.O. (2000) Systematic variation in gene expression patterns in human cancer cell lines. *Nature genetics* **24**:227-234

Wallace, C.S., Boulton, D.M. (1994) Intrinsic classification by mml - the snob program. *Proc. 8th Australian Joint Conference on AI*

## FIGURE CAPTIONS

Figure 1: NCI60. Left; Separate clustering and subset selection. The inactive cluster is inhomogeneous within sample classes. Right; Simultaneous clustering and subset selection. Fewer genes belong to active clusters. The active clusters are more homogeneous than those in the left plot.

Figure 2: NCI60. Crossvalidation test error rates. The simultaneous clustering and subset selection method (Simclust) show competitive test error rates to NN and DLDA.

Figure 3: Acute Leukemia (3 classes). Left; Separate clustering and subset selection. The active clusters (though heterogeneous) consist of genes that are upregulated predominantly in one sample class. Right; Simultaneous clustering and subset selection. The active clusters are cleaner than those to the left. The are clearly comprised of genes that are upregulated in one sample class only, whereas inactive clusters consist of genes that are upregulated in multiple classes.

Figure 4: Acute Leukemia (3 classes). Crossvalidation test error rates. The simultaneous clustering and subset selection (Simclust) show competitive test error rates to NN.

Figure 5: Acute Leukemia (2 classes). Crossvalidation test error rates. The simultaneous clustering and subset selection (Simclust) outperforms NN and DLDA.

Figure 6: Colon (2 classes). Crossvalidation test error rates. The simultaneous clustering and subset selection (Simclust) outperforms NN and DLDA.

## TABLES

| NCI60 | SimClust | SepClust | IndSelect | NN | GS |
|---|---|---|---|---|---|
| k | 6.9(.4) | 3.9(.9) | 9.3(2.0) | 2.2(.6) | NA |
| K | 9.4(.6) | 4.8(.8) | NA | NA | NA |
| # genes | 132.1 | 162.6 | 9.3 | 2.2 | 38.9 |

Table 1: NCI60 - Selected models on 150 crossvalidation data sets. Average and (standard deviation) of model sizes. K is the number of clusters. k is the number of active clusters, number of individually selected genes, or number nearest neighbors respectively.

| Leukemia3 | SimClust | SepClust | IndSelect | NN | GS |
|---|---|---|---|---|---|
| k | 3.0(.3) | 3.1(.5) | 7.3(1.5) | 3.4(2.0) | NA |
| K | 6.8(.7) | 5.0(.3) | NA | NA | NA |
| # genes | 101.8 | 141.2 | 7.3 | 3.4 | 21.5 |
| Leukemia2 | | | | | |
| k | 2.2(.1) | 2.2(.3) | 4.3(1.4) | 3.2(1.6) | NA |
| K | 4.0(.2) | 4.0(0) | NA | NA | NA |
| # genes | 85.9 | 88.3 | 4.3 | 3.2 | 24.9 |

Table 2: Leukemia (3 and 2 classes) - Selected models on 150 crossvalidation data sets. Average and (standard deviation) of model sizes.

| Colon | SimClust | SepClust | IndSelect | NN | GS |
|---|---|---|---|---|---|
| k | 2.4(.4) | 2.1(.3) | 4.2(2.2) | 4.2(1.8) | NA |
| K | 4.1(.6) | 3.5(.7) | NA | NA | NA |
| # genes | 70.6 | 101.6 | 4.2 | 4.2 | 24.2 |

Table 3: Colon - Selected models on 150 crossvalidation data sets. Average and (standard deviation) of model sizes.