# Finding Predictive Gene Groups from Microarray Data

Marcel Dettling, Peter Bühlmann

*Seminar für Statistik, ETH Zürich, CH-8092 Zürich, Switzerland*

**Abstract**

Microarray experiments generate large datasets with expression values for thousands of genes, but not more than a few dozens of samples. A challenging task with these data is to reveal groups of genes which act together and whose collective expression is strongly associated with an outcome variable of interest. To find these groups, we suggest the use of supervised algorithms: these are procedures which use external information about the response variable for grouping the genes. We present *Pelora*, an algorithm based on *pe*nalized *lo*gistic *r*egression *a*nalysis, that combines gene selection, gene grouping and sample classification in a supervised, simultaneous way. With an empirical study on six different microarray datasets, we show that *Pelora* identifies gene groups whose expression centroids have very good predictive potential and yield results that can keep up with state-of-the-art classification methods based on single genes. Thus, our gene groups can be beneficial in medical diagnostics and prognostics, but they may also provide more biological insights into gene function and regulation.

## 1 Introduction

Large-scale monitoring of gene expression by microarrays is considered to be one of the most promising techniques to improve medical diagnostics and functional genomics. Given efficient statistical methods for exploiting large gene expression datasets, accurate classification of tumor subtypes may become reality, allowing for specific treatment that maximizes efficacy and minimizes toxicity. Moreover, gene expression data are an important resource to reconstruct gene regulatory sub-networks, or more globally, to enhance understanding how the genome works.

*Email address:* `dettling@stat.math.ethz.ch` (Marcel Dettling).

*Preprint submitted to Elsevier Science* *19 December 2003*

An important task is to reveal groups of genes which act together, for example in pathways, and whose collective expression is optimally predictive for a certain response variable $y$. Our goal is to find rules such as: "if the centroid of gene 534, gene 837 and gene 235 is high, as well as the centroid of gene 2194, gene 1438, gene 931 and gene 694 is low, this is indicative of cancer subtype A". Such gene groups and their centroids can be understood as molecular signatures, which are of potential interest to accurately predict the phenotypes of new individuals in medical diagnostics, and to gain insights into biological and gene regulatory processes. However, finding the groups is difficult: we are facing computational problems due to the sheer amount of predictor variables (genes), and statistical difficulties due to the "small sample size $n$, large predictor dimension $p$"-phenomenon.

To tackle the search for groups of co-regulated genes, unsupervised clustering algorithms are widely applied in microarray analysis: mostly hierarchical clustering, but also k-means clustering, self-organizing maps and principal components, among other tools, are used. All these methods cluster genes according to unsupervised similarity measures computed from the gene expressions, but without regarding the variation of the $y$-values. Our approach differs from these popular clustering techniques, as its primary goal is to reveal gene groups that are strongly predictive for the response $y$, rather than to find homogeneous clusters made up of co-expressed genes. Hence, we suggest supervised algorithms that group genes by incorporating information from the $y$-values.

Previous work in this field encompasses partial least squares [1], a tool from chemometrics, constructing weighted linear combinations of genes that have maximal covariance with the outcome. The drawback is that every fitted component involves all (usually thousands of) genes, rather than a few genes in a group. Moreover, partial least squares for every component yields a linear combination of gene expressions which completely lacks the biological interpretation of having a group of genes acting similarly in the same pathway. Another supervised approach that improves these drawbacks is tree harvesting [2], a two-step method: first, it generates numerous candidate groups by unsupervised hierarchical clustering, and then, all group centroids are considered as potential predictor variables in a supervised response model. The gene groups that are most predictive for tissue discrimination are selected, but the initial partition remains fixed and unsupervised. A more direct approach is to combine supervised gene selection and gene grouping in one single step. We proposed such a procedure under the heading "Supervised clustering of genes" in [3]. Another single-step approach based on Rissanen's minimum description length principle was pursued by Jörnsten and Yu [4].

Here, we formulate a generic strategy for supervised grouping approaches: it combines gene selection and gene grouping in a single step, and is based on sequentially improving an empirical objective function that measures the groups' strength for explaining the outcome $y$. We briefly review our first implementation from [3], which is called *Wilma*, since its grouping criterion is based on the *Wil*coxon and *ma*rgin statistics. Then, we present *Pelora*, a novel approach to supervised grouping of genes, using an objective function based on *pe*nalized *lo*gistic *r*egression *a*nalysis. It improves upon *Wilma* in many ways. It allows for overlapping groups of genes, as motivated from biology, since some genes operate in multiple pathways; furthermore, *Pelora* yields better interaction between the gene groups, it is more robust, it allows for including additional clinical covariates to refine the grouping, it can be easily adapted to continuous response problems and it encompasses a built-in classifier. But the improvements are not just on the theoretical and methodological side: our new implementation *Pelora* also yields very good empirical prediction results, especially when the discrimination between tissue types is difficult.

## 2 Some Motivation for Supervised Grouping of Genes

### *2.1 Gene Expression Data*

Our stochastic notion of a microarray experiment is given by a random pair $(\mathbf{x}, y)$, where $\mathbf{x} \in R^p$ is the gene expression profile, monitoring up to several thousands of genes. $y \in \{0, 1\}$ is a dichotomous response, extensions to polytomous or continuous response are discussed in section 3.4.3. The data are assumed to be independent and identically distributed realizations of such random pairs,

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n),$$

where the number of experiments $n$ is typically between 10 and 100. The predictor variables are stored in a $(n \times p)$-matrix $(x_{ig})$, where rows $\mathbf{x}_i$ correspond to experiments and are printed in bold face, whereas columns $x_g$ correspond to genes and are printed in normal font. For our supervised grouping methodology, the expression profile $\mathbf{x}$ can be either from Affymetrix oligonucleotide chips or two-color cDNA arrays, but we assume it to be thoroughly preprocessed and log-transformed.

### *2.2 Two-Population Models*

Our approach for grouping genes is very different from popular clustering based on similarity measures such as correlation (between genes or cluster centroids). For understanding supervised grouping of genes, it is instructive to consider first a simple model: we have two populations, encoded by 0 and 1, according to the value of the binary response $y = 0$ or $y = 1$, respectively. For notational simplicity, we order the data samples such that the first $n_0 = \sum_{i=1}^{n}(1 - y_i)$ observations belong to population 0 and the last $n_1 = \sum_{i=1}^{n} y_i$ to population 1. The model is then

$$\begin{aligned} \mathbf{x}_1, \ldots, \mathbf{x}_{n_0} &\text{ i.i.d. with c.d.f. } F(\cdot - \mu^{(0)}) \text{ in population } 0, \\ \mathbf{x}_{n_0+1}, \ldots, \mathbf{x}_n &\text{ i.i.d. with c.d.f. } F(\cdot - \mu^{(1)}) \text{ in population } 1, \end{aligned} \tag{1}$$

where $F(\cdot)$ is a $p$-dimensional cumulative distribution function with expectation equal to the zero vector. Thus, the populations differ only in their mean vectors which is one of the simplest models of this class. Model (1) becomes a simple two-population group model if

$$\begin{aligned} \mu^{(0)} &= (\mu^{(0)}_{\mathcal{G}_1}, \ldots, \mu^{(0)}_{\mathcal{G}_1}, \mu^{(0)}_{\mathcal{G}_2}, \ldots, \mu^{(0)}_{\mathcal{G}_2}, \ldots, \mu^{(0)}_{\mathcal{G}_q}, \ldots, \mu^{(0)}_{\mathcal{G}_q}), \\ \mu^{(1)} &= (\mu^{(1)}_{\mathcal{G}_1}, \ldots, \mu^{(1)}_{\mathcal{G}_1}, \mu^{(1)}_{\mathcal{G}_2}, \ldots, \mu^{(1)}_{\mathcal{G}_2}, \ldots, \mu^{(1)}_{\mathcal{G}_q}, \ldots, \mu^{(1)}_{\mathcal{G}_q}), \end{aligned} \tag{2}$$

where we have $q$ groups $\mathcal{G}_1, \ldots \mathcal{G}_q$ that form a partition of the gene index set $\{1, \ldots, p\}$. Within each gene group $\mathcal{G}$, all genes have the same expectation $\mu^{(0)}_{\mathcal{G}}$ or $\mu^{(1)}_{\mathcal{G}}$, respectively; for notational simplicity, we have reordered the genes such that the first group $\mathcal{G}_1$ consists of the first genes $1, 2, \ldots, |\mathcal{G}_1|$, and the last group consists of the last genes $p - |\mathcal{G}_q| + 1, \ldots, p$.

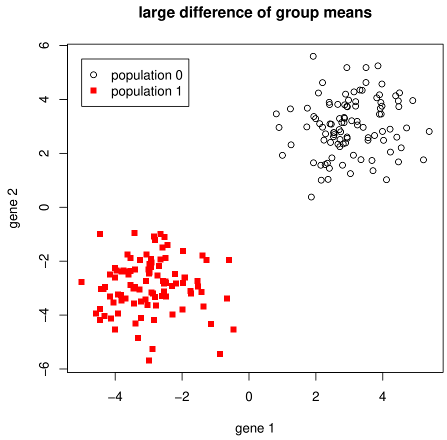


Fig. 1. Scatterplot of two genes from a group $\mathcal{G}$ with $\mu^{(0)}_{\mathcal{G}} = -3$, $\mu^{(1)}_{\mathcal{G}} = 3$.

The magnitude of the difference $|\mu^{(0)}_{\mathcal{G}} - \mu^{(1)}_{\mathcal{G}}|$ for a certain gene group $\mathcal{G}$ heavily influences the ability to recover such a structure from data. We simulated genes from one group of size $|\mathcal{G}| = 10$ according to model (1), with the cumulative distribution function $F(\cdot)$ chosen as the $\mathcal{N}_{10}(0, I)$-distribution. Figure 1 shows the scatterplot of two genes from this group $\mathcal{G}$ with $\mu^{(0)}_{\mathcal{G}} = -3$, and $\mu^{(1)}_{\mathcal{G}} = 3$, which exhibits a large difference compared to the noise level and in turn, implies a large sample correlation of 0.91 between the two genes in figure 1. Thus, if the difference $|\mu_{\mathcal{G}}^{(0)} - \mu_{\mathcal{G}}^{(1)}|$ is large, it is quite likely that such a group of genes can be detected by clustering methods based on the correlation similarity measure.

When taking the same setup but with smaller $|\mu_{\mathcal{G}}^{(0)} - \mu_{\mathcal{G}}^{(1)}| = 2$, the empirical correlation between two genes from group $\mathcal{G}$ drops down to 0.53 and there is no clear separation between the populations, as evident from the left panel in figure 2. The correlation of 0.53, which is low in the context of microarray

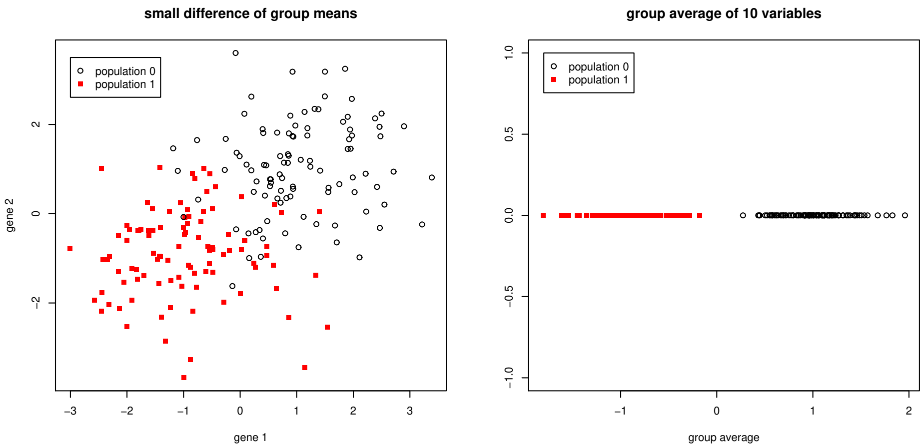


Fig. 2. Left: scatterplot of two genes from a group $\mathcal{G}$ with $\mu_{\mathcal{G}}^{(0)} = -1$ and $\mu_{\mathcal{G}}^{(1)} = 1$. Right: average expression $\tilde{x}$ of a group with size $|\mathcal{G}| = 10$.

gene expression data, is an indication that correlation based clustering will have difficulties in recovering the group $\mathcal{G}$ from data.

However, we can actively make use of the information which samples belong to population group 0 and 1 by plotting gene group averages

$$\tilde{x} = \tilde{x}_{\mathcal{G}} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} x_g$$

and check how well the group average $\tilde{x}$ separates the two population groups. This is demonstrated in the right panel of figure 2 for a true group of size $|\mathcal{G}| = 10$ and with the "difficult" structure having small differences between the population group means $|\mu_{\mathcal{G}}^{(0)} - \mu_{\mathcal{G}}^{(1)}| = 2$.

The key observation why the approach illustrated in the right panel of figure 2 works, is that the group average $\tilde{x}$ has smaller variability than single genes. In particular, $\tilde{x}_i$ for a true group $\mathcal{G}$ is an estimate of both $\mu_{\mathcal{G}}^{(0)}$ and $\mu_{\mathcal{G}}^{(1)}$, depending whether the sample index $i$ belongs to $y_i = 0$ or $y_i = 1$, respectively. Moreover, if the true group size $|\mathcal{G}|$ is sufficiently large, we will obtain a perfect separation of the populations with $\tilde{x}$, i.e.

$$\max_{i,y_i=0} \tilde{x}_i < \min_{i,y_i=1} \tilde{x}_i \quad \text{or} \quad \min_{i,y_i=0} \tilde{x}_i > \max_{i,y_i=1} \tilde{x}_i. \tag{3}$$

Hence, we "only" need to check - and we can do this because we are working in a supervised context - how well the candidate group average $\tilde{x}$ separates the two populations as in the right panel of figure 2. In summary, if the true group size $|\mathcal{G}|$ is large relative to the magnitude of the population mean differences $|\mu_{\mathcal{G}}^{(0)} - \mu_{\mathcal{G}}^{(1)}|$, we will have a good chance to discover $\mathcal{G}$ from data. This can be quantified, since under reasonable conditions on the correlation between the genes,

$$\sqrt{\mathrm{Var}(\tilde{x}|y)} \sim C_y/\sqrt{|\mathcal{G}|} \text{ for some constant } C_y > 0 \text{ as } |\mathcal{G}| \to \infty,$$

which will be small relative to $|\mu_{\mathcal{G}}^{(0)} - \mu_{\mathcal{G}}^{(1)}|$ if $|\mathcal{G}|$ is large.

### *2.3 Beyond the Two-Population Group Model*

The two-population group model in (2) seems somewhat unrealistic. First, for both populations, the genes within group $\mathcal{G}$ may have different mean values instead of being all exactly equal to some $\mu_{\mathcal{G}}^{(y)}$. More importantly, when going through the arguments above, we can achieve a separation rule as in (3) if

$$|\overline{\mu}_{\mathcal{G}}^{(0)} - \overline{\mu}_{\mathcal{G}}^{(1)}| \text{ is large relative to } max\left(\sqrt{\mathrm{Var}(\tilde{x}|y=0)}, \sqrt{\mathrm{Var}(\tilde{x}|y=1)}\right),$$

where $\overline{\mu}_{\mathcal{G}}^{(y)} = \frac{1}{|\mathcal{G}|}\sum_{g\in\mathcal{G}} \mu_g^{(y)}$ $(y \in \{0,1\})$. Requiring the maximum of the conditional standard deviations may be a bit too stringent, but certainly sufficient.

Thus, a gene group $\mathcal{G}$ pays off, if every gene $g \in \mathcal{G}$ has: a) a *large expected differential expression* $|\mu_g^{(0)} - \mu_g^{(1)}|$, as well as the same $sign(\mu_g^{(0)} - \mu_g^{(1)})$, and: b) the pairwise conditional correlations $Cov(x_g, x_{g'}|y)$ are low for all genes $g, g' \in \mathcal{G}$, yielding small conditional variances $Var(\tilde{x}|y)$.

Clearly, this involves a trade-off between expected differential expression and variance: if a gene $g^*$ has the largest expected differential expression, the absolute difference $|\overline{\mu}_{\mathcal{G}}^{(0)} - \overline{\mu}_{\mathcal{G}}^{(1)}|$ will be smaller (which is worse) for any superset group $\mathcal{G} \supset g^*$, while the conditional variances $Var(\tilde{x}|y)$ will decrease.

In addition, we want to construct multiple gene groups, each of which exhibiting a good trade-off between expected differential expression and conditional variance of the group mean as discussed above. The reason is that for a two-population model, the response $y$ can typically be more accurately predicted with multiple group averages $\widetilde{x}_1, \dots, \widetilde{x}_q$, at least as long as these $q$ group representatives are not too strongly conditionally dependent given the binary response $y \in \{0, 1\}$.

### *2.4 Structure of Supervised Gene Groups*

In summary, our methods for supervised grouping of genes, as described in sections 3.3 and 3.4, aim to identify multiple class separating groups $\mathcal{G}_1, \dots, \mathcal{G}_q$, such that each group exhibits a good trade-off between expected differential expression and conditional variance of the group mean, and such that the $q$ groups together contribute most in predicting the response $y$. These gene groups are not necessarily "homogeneous" gene clusters, and they will typically not reflect "co-expression" in the classical sense that all genes in a group would be very tightly over- or under-expressed, respectively. However, we do get gene groups whose representatives $\widetilde{x}_1, \dots, \widetilde{x}_q$ can be interpreted as a gene signature that is strongly differentially expressed and carries substantial information about predicting $y$.

## 3 Methods

### *3.1 Probabilistic Model*

To account for the fact that not all $p$ genes on the chip, but rather a few functional gene subsets determine nearly all of the outcome variation, we model the conditional probability by

$$P[y = 1|\mathbf{x}] = f\left(\widetilde{\mathbf{x}}\right) \text{ with } \widetilde{\mathbf{x}} = \left(\widetilde{x}_1, \widetilde{x}_2, \dots, \widetilde{x}_q\right), \tag{4}$$

where $f(\cdot)$ is an unknown nonlinear function and $\widetilde{x}_j$ are 'representative' values for $q \ll p$ unknown gene groups $\mathcal{G}_1, \dots, \mathcal{G}_q$. Similarly as in section 2, we use the centroid

$$\widetilde{x}_j = \frac{1}{|\mathcal{G}_j|} \sum_{g \in \mathcal{G}_j} \alpha_g x_g \text{ with } \alpha_g \in \{-1, 1\}$$

as the representative group value. The unknown discrete parameter $\alpha_g$ is used to allow for over- and underexpressed genes in the same group. These sign-flips can be regarded as an optional feature in our method and software.

### 3.2 *Supervised Grouping: A Generic Strategy*

The combinatorial complexity for grouping gene expression data is huge. As a toy example, consider a dataset of 5,000 genes: there are more than $2 \cdot 10^{30}$ possibilities for obtaining one single group of 10 genes. Because the partition of thousands of genes into a few signature components that virtually determine the probability structure as in (4) is by far more complex than our toy example, it is impossible to use an exhaustive search to reveal the optimal partition among all possible solutions. Thus, we suggest a computationally intensive grouping heuristic that turns out to yield good empirical results.

Our approach is based on a strategy which proceeds in a "cautious" forward way. We start from scratch and rely on growing the groups incrementally by adding one gene after the other. Regularly recurring cleaning steps help us to remove spurious genes that were incorrectly added to the groups at earlier stages. We repeat growth and pruning of a single group until it stabilizes and cannot be improved any further. Once a group is found to be terminated, a new group is started and the composition of the former groups is left unchanged, while they can still have an effect on the construction of the new group. All these grouping operations are based on an empirical objective function $S$, which measures the strength of the gene groups for explaining the response $y$. Its choice is discussed in sections 3.3 and 3.4.

### 3.3 *Wilma - a First Implementation*

Our first supervised algorithm for gene grouping is called *Wilma* and follows the generic strategy described above. It was published under the heading "Supervised clustering of genes" [3]. The name *Wilma* is an acronym for the *Wil*coxon and *ma*rgin criteria which are used for the objective function $S$. The procedure yields convincing empirical results in terms of the predictive potential, the stability and the relevance of its groups. However, it suffers from a few limitations. First, the groups need to be disjoint, and hence *Wilma* cannot capture genes that operate in multiple pathways. Next, each group is (up to the disjointness to the former groups) built independently of all the others. So, it may happen that each group tries to optimally predict the response $y$ on its own, instead of finding an ensemble of interacting groups. Then, the grouping criterion $S$ is non-penalized, which might lead to overfitting. Moreover, it is non-robust and may result in very hard supervision. *Wilma* has been successful in "easy" classification problems, but some milder form of supervision (less influence of the response) leads to better empirical results in difficult, inhomogeneous classification problems with substantial Bayes risk.

### 3.4 Pelora

We present now a new supervised grouping algorithm called *Pelora*. It still follows the generic strategy described in section 3.2, but addresses all the limitations of *Wilma*. It mainly differs in the supervised grouping criterion $S$. We employ the $\ell_2$-penalized negative log-likelihood function

$$S = -\sum_{i=1}^{n} \left( y_i \cdot \log p_\theta(\tilde{\mathbf{x}}_i) + (1 - y_i) \cdot \log(1 - p_\theta(\tilde{\mathbf{x}}_i)) \right) + n\frac{\lambda}{2}\theta^T P\theta, \tag{5}$$

based on estimated conditional class probabilities $p_\theta(\tilde{\mathbf{x}}) = P_\theta[y = 1|\tilde{\mathbf{x}}]$ from *pe*nalized *lo*gistic *r*egression *a*nalysis, hence the name *Pelora*. Note that $\theta$ is the parameter vector, $\lambda$ is a tuning parameter that controls the penalization and $P$ is a penalty matrix, for further details we refer to section 3.4.1. The binomial log-likelihood is an attractive choice as a grouping criterion, since it is the 'natural' goodness-of-fit measure for dichotomous problems. Another advantage is that with multiple groups, it allows to judge the discriminatory power of the $(q + 1)$-dimensional predictor $\tilde{\mathbf{x}} = (1, \tilde{x}_1, \ldots, \tilde{x}_q)$, whereas the Wilcoxon and margin criteria in *Wilma* only work with one-dimensional input. By computing the grouping criterion directly from multiple groups instead of single groups only, we obtain better interacting gene groups that explain the response $y$ as an ensemble. Technical issues concerning penalized logistic regression and full details about the grouping procedure are given in the next two sections.

#### 3.4.1 Penalized Logistic Regression Analysis

Penalized logistic regression analysis [5] has been used as a stand-alone for classification of microarray gene expression data with single genes. Eilers et al. [6] as well as Zhu and Hastie [7] focus on computational issues that arise from the "small $n$, large $p$" dimensionality phenomenon and report improved results compared to non-penalized logistic regression. Since we use the penalized version as an estimator in conjunction with our $q < n$ groups, we avoid such difficulties and can apply computationally simple methodology. The classical logistic model is then defined as

$$\log\left(\frac{p_\theta(\tilde{\mathbf{x}}_i)}{1 - p_\theta(\tilde{\mathbf{x}}_i)}\right) = \sum_{j=0}^{q} \theta_j \tilde{x}_{ij} = \tilde{\mathbf{x}}_i\theta, \text{ for observations } i = 1, \ldots, n,$$

with parameter vector $\theta^T = (\theta_0, \theta_1, \ldots, \theta_q)$ and $x_{i0} = 1$. The idea of penalized logistic regression is to estimate $\theta$ by a $\ell_2$-penalized maximum likelihood principle. We minimize

$$S(\theta) = -\sum_{i=1}^{n} (y_i \cdot \log p_\theta(\tilde{\mathbf{x}}_i) + (1 - y_i) \cdot \log(1 - p_\theta(\tilde{\mathbf{x}}_i))) + n\frac{\lambda}{2}\theta^T P\theta \tag{6}$$

for fixed $\tilde{\mathbf{x}}_i$ with respect to $\theta$. Note that (5) and (6) are identical, but the goal in (6) is to estimate the parameter vector $\theta$ by minimizing $S$ for fixed predictors, whereas for supervised grouping, we try to find the (possibly overlapping) partition whose centroid-predictors optimize $S$ in (5) with optimal parameter $\theta$ from (6). $P$ is the penalty matrix, defined as

$$P = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ 0 & Var(\tilde{x}_1) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & Var(\tilde{x}_{(q-1)}) & 0 \\ 0 & 0 & \dots & 0 & Var(\tilde{x}_q) \end{pmatrix} \tag{7}$$

a matrix which has the predictors' variance in the diagonal and zeros elsewhere. The reason to use this non-unit penalty matrix is that, in contrast to common practice in penalized regression, we do not standardize the predictors, i.e. the group representatives $\tilde{x}_j$, to unit variance. By using $P$ as defined above, we obtain the same solution as when using the standard unit matrix as a penalty in conjunction with standardized predictors. The proof is given in Appendix A. To get to the solution of the minimization problem in (6), we take derivatives with respect to $\theta$,

$$\frac{\partial S}{\partial \theta} = \widetilde{X}^T(y - \pi_\theta) - n\lambda P\theta \stackrel{!}{=} 0 \quad \in \quad R^{q+1},$$

where $\widetilde{X} = (1, \tilde{x}_{i1}, \dots, \tilde{x}_{iq})_{i=1,\dots,n}$ is the design matrix containing the group centroids and $\pi_\theta = (p_\theta(\tilde{\mathbf{x}}_1), \dots, p_\theta(\tilde{\mathbf{x}}_n))^T$ is the conditional probability vector for all $n$ observations. This yields $(q+1)$ non-linear equations, whose solution needs to be approximated. We do this iteratively by Newton-Raphson stepping and obtain the new estimate $\theta^{new}$ from

$$\theta^{new} = \theta - \left(\frac{\partial^2 S}{\partial\theta\partial\theta^T}\right)^{-1} \cdot \frac{\partial S}{\partial \theta}$$

For an explicit computation of the step length, we use the second derivative

$$\frac{\partial^2 S}{\partial\theta\partial\theta^T} = -\left(\widetilde{X}^T W_\theta \widetilde{X}\right) - n\lambda P \quad \in \quad R^{(q+1)\times(q+1)},$$

where the matrix $W_\theta$ is a diagonal weight matrix, defined as

$$W_\theta = diag\left((p_\theta(\widetilde{\mathbf{x}}_i)(1 - p_\theta(\widetilde{\mathbf{x}}_i)))_{i=1,\ldots,n}\right).$$

Then, we plug in and with

$$\theta^{new} = \left(\widetilde{X}^T W_\theta \widetilde{X} + n\lambda P\right)^{-1} \cdot \left(\widetilde{X}^T(y - \pi_\theta) + (\widetilde{X}^T W_\theta \widetilde{X})\theta\right),$$

we obtain an iterative procedure for estimation of the parameter vector $\theta$. The initial values for $\theta$ are chosen as

$$\theta_0^{(0)} = \log\left(\frac{\bar{y}}{1-\bar{y}}\right) \text{ and } \theta_j^{(0)} = 0 \text{ for all } j = 1, \ldots, q,$$

where $\bar{y} = \frac{1}{n}\sum y_i$. This means that $p_{\theta^{(0)}}(\widetilde{\mathbf{x}}_i) = \bar{y}$, that is, the initial probabilities reflect the class proportions in the training data. If these are not representative and a priori probabilities are known, the initial parameter values should be chosen appropriately. The Newton-Raphson algorithm in general converges rapidly and not more than 5-10 iterations are necessary until the solution stabilizes. For our grouping algorithm, we do not iterate until convergence, but restrict to two full rounds, meaning that

$$\theta^{(0)} \leadsto \theta^{(1)} \leadsto \theta^{(2)} = \theta$$

is our final estimate in the penalized logistic regression model. The reason is to save computing time: every iteration requires solving a linear equation system, which is by far the most time consuming operation in our supervised algorithm; note that we will run such 2-step Newton-Raphson very many times. The first iteration yields the least squares ridge-type linear regression solution. This is already a consistent estimator, if $\lambda$ is chosen appropriately. The second Newton-Raphson iteration typically yields asymptotic efficiency, see [8]. Thus, this guarantees from a theoretical viewpoint, that our procedure is precise enough. From an empirical viewpoint, we observed that the probability "pattern" over the $n$ observations did not change much after 2 iterations. Thus, the grouping did hardly ever change at all if more than 2 iterations were done.

#### *3.4.2 The Pelora Algorithm*

First, we give the details about 2 initial steps for our supervised grouping procedure. Start with the entire $(n \times p)$ gene expression matrix $(x_{ig})$.

1. Standardize the expression values $x_{ig} = (x_{1g}, \dots, x_{ng})$ of every gene $g$ to zero mean and unit variance:

$$x_{ig} \leftarrow \frac{x_{ig} - ave(x_g)}{sdev(x_g)}, \quad \text{for } i = 1, \dots, n.$$

With this standardization, we follow a widely adopted practice in gene clustering and in penalty-based methods. It can, however, be regarded as an optional step in our algorithm and software. Note that the rescaling to unit variance, but not the mean centering, affects the outcome of *Pelora*.

2. The algorithm can be started from scratch or with initial groups $\mathcal{G}_1, \dots, \mathcal{G}_{(q-1)}$ that reflect previous knowledge, for example about biochemical pathways. Compute the centroids of the initial groups,

$$\tilde{x}_j = \frac{1}{|\mathcal{G}_j|} \sum_{g \in \mathcal{G}_j} \alpha_g x_g \text{ for } j = 1, \dots, (q-1) \text{ and } \alpha_g \in \{-1, 1\},$$

where $|\mathcal{G}_j|$ is the number of genes in group $\mathcal{G}_j$. The optional parameter $\alpha_g$ allows one to have genes with different polarity, that is, one with low expression for class 0 and the other one with low expression for class 1, in the same group. It prevents their expressions from canceling out in the group centroid. In the next step, we detail how to identify the starting gene for a new group.

3.a) IF no groups are given, we start from scratch with predictor $\tilde{\mathbf{x}} = (1)$. The goal is to find the starting gene of group $\mathcal{G}_q$ with $q = 1$.

3.b) IF an initial structure of $(q-1)$ groups is given or already found, and the current predictor is $\tilde{\mathbf{x}} = (1, \tilde{x}_1, \dots, \tilde{x}_{(q-1)})$, the goal is to find the starting gene for group $\mathcal{G}_q$.

3.c) Fit penalized logistic regression with predictor $\tilde{\mathbf{x}}^{+g} = (\tilde{\mathbf{x}}, 1 \cdot x_g)$ for every gene $g$ with $\alpha_g = 1$ to obtain an estimated parameter vector $\theta^{+g}$ and conditional class probabilities $p_{\theta^{+g}}(\tilde{\mathbf{x}}^{+g})$. Use them to compute the penalized negative log-likelihood $S^{+g}$ as in (5). Determine the winning gene $g^* = \arg\min_g S^{+g}$ and set the initial centroid of the $q$th group to $\tilde{x}_q = x_{g^*}$.

For the remainder of the algorithm, we assume without loss of generality that $q$ groups with centroids $\tilde{x}_1, \dots, \tilde{x}_q$ are given. Group $\mathcal{G}_q$ is non-terminated and we try to add another gene. Assume that the current value of the objective function is $S^{old}$.

4. FOR each gene $g = 1, \dots, p$ repeat: Leave groups $\mathcal{G}_1, \dots, \mathcal{G}_{(q-1)}$ un-

changed, build temporary candidate groups $\mathcal{G}_q^{+g}$ and $\mathcal{G}_q^{-g}$ by augmenting $\mathcal{G}_q$ with gene $g$ and polarity parameter $\alpha_g \in \{-1, +1\}$. The group centroid is updated as

$$\widetilde{x}_q^{+g} = \frac{|\mathcal{G}_q| \cdot \widetilde{x}_q + 1 \cdot x_g}{|\mathcal{G}_q| + 1} \quad \text{and} \quad \widetilde{x}_q^{-g} = \frac{|\mathcal{G}_q| \cdot \widetilde{x}_q + (-1) \cdot x_g}{|\mathcal{G}_q| + 1}.$$

Fit penalized logistic regression with predictors $\widetilde{\mathbf{x}}^{+g} = (1, \widetilde{x}_1, \ldots, \widetilde{x}_q^{+g})$ and $\widetilde{\mathbf{x}}^{-g} = (1, \widetilde{x}_1, \ldots, \widetilde{x}_q^{-g})$ to obtain the parameter vectors $\theta^{+g}$ and $\theta^{-g}$, as well as conditional probabilities $p_{\theta^{+g}}(\widetilde{\mathbf{x}}^{+g})$ and $p_{\theta^{-g}}(\widetilde{\mathbf{x}}^{-g})$. Compute the penalized negative log-likelihoods $S^{+g}, S^{-g}$ as in (5). Let $S^g = \min(S^{+g}, S^{-g})$.

5. Identify the winning gene $g^* = \arg\min_g S^g$. Compare it to $S^{old}$, the criterion value before gene $g^*$ was added.

6.a) IF not improved, i.e. $S^{g^*} > S^{old}$: Do not accept the gene, terminate the group, continue with groups $\mathcal{G}_1, \ldots, \mathcal{G}_q$ and their centroids. If $q < q_{final}$, increment $q$ and return to step 3 to start a new group.

6.b) IF improved, i.e. $S^{g^*} < S^{old}$: Accept the gene, determine the its polarity parameter $\alpha_{g^*}$ and update group, group centroid and criterion value to

$$\begin{array}{lllll} \alpha_{g^*} & \leftarrow & sign(S^{-g^*} - S^{+g^*}), & \mathcal{G}_q & \leftarrow \mathcal{G}_q \cup \{g^*\}, \\ \widetilde{x}_q & \leftarrow & \frac{|\mathcal{G}_q| \cdot \widetilde{x}_q + \alpha_{g^*} \cdot x_{g^*}}{|\mathcal{G}_q|+1}, & S^{old} & \leftarrow S^{g^*}. \end{array}$$

7. FOR each gene $g = 1, \ldots, \tilde{p}$ in group $\mathcal{G}_q$ repeat: Leave groups $\mathcal{G}_1, \ldots, \mathcal{G}_{(q-1)}$ unchanged and build the temporary candidate group $\mathcal{G}_q^g$ by excluding gene $g$ from group $\mathcal{G}_q$. Update the group centroid,

$$\widetilde{x}_q^g = \frac{1}{|\mathcal{G}_q| - 1} \sum_{g' \in \mathcal{G}_q \setminus \{g\}} \alpha_{g'} x_{g'}.$$

Fit penalized logistic regression with predictor $\widetilde{\mathbf{x}}^g = (1, \widetilde{x}_1, \ldots, \widetilde{x}_q^g)$ to obtain the parameter vector $\theta^g$ and conditional probabilities $p_{\theta^g}(\widetilde{\mathbf{x}}^g)$. Compute the penalized negative log-likelihood $S^g$ as in (5).

8. Identify the gene $g^* = \arg\min_g S^g$, whose exclusion minimizes the grouping criterion and compare it to $S^{old}$.

9.a) IF not improved, i.e. $S^{g^*} > S^{old}$: Do not delete the gene, continue with groups $\mathcal{G}_1, \ldots, \mathcal{G}_q$ (note that $\mathcal{G}_q$ was augmented in step 6) and their centroids. Try to add another gene by restarting at step 4.

9.b) IF improved, i.e. $S^{g^*} < S^{old}$: Exclude gene $g^*$ and update group, group centroid and criterion value by

$$\mathcal{G}_q \leftarrow \mathcal{G}_q \setminus \{g^*\}, \quad \tilde{x}_q \leftarrow \tilde{x}_q^{g^*}, \quad S^{old} \leftarrow S^{g^*}.$$

Now try to add another gene by restarting at step 4.

In summary, our supervised algorithm is a one-step procedure for variable selection, variable grouping and formation of new features by averaging the gene expression within a group, including potential sign-flipping. Variable selection and grouping are done with a stepwise forward search, where we try all genes and augment the group by the gene which optimizes the criterion $S$ from (5). After each forward search, we continue with a backward pruning step to root out genes that have been added wrongly to the group at earlier forward stages. Again, we try all genes and decide on removal by optimizing the criterion $S$. Our grouping procedure is supervised, since all decisions are based on optimizing the criterion $S$ that measures the ability of the groups for explaining the response variable $y$.

The number of groups $q_{final}$ can be set according to previous knowledge, it can be chosen data-adaptively by cross validation, or it can be estimated by techniques such as proposed in [9,10]. The computing time for finding $q = 10$ groups in the AML/ALL leukemia dataset with $n = 72$ observations and $p = 3,571$ genes on a Linux PC with an Intel Pentium IV 1.6 GHz processor is about 560 seconds. Software for our supervised grouping algorithms is available under GNU public license as an R-package called `supclust` from our webpage `http://stat.ethz.ch/~dettling/supervised.html`. In the next sections, we discuss how *Pelora* can be extended to non-dichotomous response, to a forward selection procedure based on single genes, and how additional clinical covariates can be embedded into the grouping.

### *3.4.3 How to Deal with Multiclass Problems*

Polytomous response problems will be handled by reformulating them as multiple binary problems. This approach has been successful for a wide variety of machine learning methods on many datasets. With microarray data, according to our experience from [11], it often improves substantially upon simultaneous multiclass versions, especially when variable selection is involved. The reason is that it is hard to come up with single genes that accurately discriminate polytomous response.

Various approaches for reducing a $K$-class problem with $y \in \{0, \ldots, K-1\}$ to binary problems exist, see [12] for a thorough discussion. We observed good empirical prediction results already with the most simple solution, the one-against-all approach. It works by defining

$$y^{(k)} = \begin{cases} 1, \text{ if } y = k, \\ 0, \text{ else} \end{cases}$$

for $k = 0, \ldots, K-1$, and running the supervised grouping algorithm $K$ times on the dichotomous-response datasets $(\mathbf{x}_1, y_1^{(k)}), \ldots, (\mathbf{x}_n, y_n^{(k)})$ as explained above. For each binary problem, this finally yields $q$ group centroids $\tilde{x}_1^{(k)}, \ldots, \tilde{x}_q^{(k)}$ that can be used as features for polytomous classification. Instead of considering each class against all the other classes, more complex or problem dependent strategies that utilize deeper knowledge about the biological relation between the response classes could be even more accurate for reducing multi-category to multiple binary problems.

### *3.4.4 How to Incorporate Clinical Covariates*

Cancer prognosis is traditionally done on the basis of clinical covariates such as gender, patient age, tumor size, metastasis, cytogenetic aberrations and many more. Some of these are easy to record and it is thus a waste of useful information if modern cancer prognosis just relies on microarray data without regarding the clinical status of a patient. We present here an approach for cancer prognosis that combines microarray gene expression data with clinical covariates. We also address the question of statistical inference in section 4.4. Instead of the random pair $(\mathbf{x}, y)$, we now have a random triple $(\mathbf{x}, \mathbf{u}, y)$, where $\mathbf{u} \in R^m$ are the $m$ clinical covariates. These can either be continuous, polytomous or binary, even a mixture of all three types is allowed. We assume to have complete clinical data for all $n$ patients.

For model selection, we apply our algorithm *Pelora*, still based on optimizing the log-likelihood from (5) with penalized logistic regression. The idea is to identify a combination of gene groups and clinical variables that is optimally predictive for the response $y$. In particular, the predictor $\tilde{\mathbf{x}}$ can now both contain group centroids $\tilde{x}_j$ and clinical covariates $u_k$. To allow this, we just need to formulate step 3.c) from our grouping procedure a bit more precisely:

3.c) Fit penalized logistic regression with the augmented predictor $\tilde{\mathbf{x}}^{+g} = (\tilde{\mathbf{x}}, 1 \cdot x_g)$ for every gene $g$ and with $\tilde{\mathbf{x}}^{+k} = (\tilde{\mathbf{x}}, 1 \cdot u_k)$ for every clinical covariate $k$ to obtain estimated parameter vectors $\theta^{+g}$ and $\theta^{+k}$, as well as conditional class probabilities $p_{\theta^{+g}}(\tilde{\mathbf{x}}^{+g}), p_{\theta^{+k}}(\tilde{\mathbf{x}}^{+k})$. Compute ...

3.c) ...the penalized negative log-likelihoods $S^{+g}, S^{+k}$ as in (5). Determine the winning gene $g^* = \arg\min_g S^{+g}$ and the best covariate $k^* = \arg\min_k S^{+k}$. If $\min(S^{+g^*}, S^{+k^*}) = S^{+g^*}$, start a new group, set $\widetilde{x}_q = x_{g^*}$ and continue with step 4. Else, if $\min(S^{+g^*}, S^{+k^*}) = S^{+k^*}$, pick up covariate $k^*$ into the predictor, set $\widetilde{x}_q = u_{k^*}$ and restart at step 3 to identify the next predictor variable.

Thus, if a clinical covariate optimizes the grouping criterion $S$ in step 3, it is directly incorporated into the model without any grouping or averaging, and we proceed by incrementing the current number $q$ of predictors and restart at step 3 to find the next starting gene or the next clinical covariate. On the other hand, if a gene leads to the lowest value of $S$ in step 3, we set the initial group centroid equal to this gene and continue with step 4 to build a group.

### *3.4.5 Forward Search Without Averaging*

As pointed out by a referee, the *Pelora* algorithm can also be run as a forward variable selection tool based on penalized logistic regression. Each predictor variable $\widetilde{x}_j$ consists of one single gene and neither any grouping nor any averaging takes place. Thus, the gene that optimizes the grouping criterion $S$ in step 3 of our algorithm is incorporated into the model and the algorithm proceeds by incrementing the current number of predictor variables $q$ and restarts at step 3 to find the next gene. When performing such a forward selection, steps 4-9 of the algorithm are obsolete. This *for*ward *sel*ection *a*pproach will be called *Forsela*.

### *3.4.6 Pelora in Comparison to Forsela*

From a modeling point of view, both *Pelora* and *Forsela* perform gene selection and fit a penalized linear logistic model with the selected genes. In *Pelora*, an additional constraint comes in, this is, that the regression parameters are the same for all genes within the same group. Thus, *Pelora*'s constraint can be viewed as a further regularization, besides the $\ell_2$-penalty in the objective function $S$. In view of the ridge-type $\ell_2$-penalty, *Forsela* penalizes every gene (standardized to variance one) by the same amount while the matrix $P$ for *Pelora*, appearing in (5), implies a *variable* ridge penalty for the gene groups, which is inversely proportional to the group size $1/|\mathcal{G}|$. Intuitively, this is the right notion since large groups have low-variance centroids, as motivated in section 2.3.

It is important to point out that *Pelora* does a more drastic dimensionality reduction, by reducing to the group centroids, than *Forsela* which reduces to the selected single genes. Moreover, the group centroids in *Pelora* have lower variance than single genes which often results in lower variablity in out-of-sample predictions. The usefulness of such low-variance features, also known as meta- or super-genes, has been recognized by others, see for example [13]. Thus, *Pelora* can be viewed as a supervised method to construct good class-discriminatory meta-genes.

#### *3.4.7 Extension to Continuous Response Problems*

If the interest is in finding gene groups whose collective expression is informative for continuous responses such as tumor size or drug response, *Pelora* can be easily adapted. The grouping algorithm is still supervised and follows the description from section 3.4.2, but it differs in the objective function $S$ and does no longer rely on penalized logistic regression as a learner. Instead, we may use the $\ell_2$-penalized residual sum of squares

$$S = \sum_{i=1}^{n} (y_i - m_\theta(\tilde{\mathbf{x}}_i))^2 + \frac{n}{2}\lambda\theta^T P\theta, \tag{8}$$

based on $m_\theta(\tilde{\mathbf{x}}_i)$ from (9), where $\theta$ is the parameter vector, $\lambda$ is the tuning parameter and $P$ is the non-unit penalty matrix from equation (7). The $(q+1)$-dimensional predictor is $\tilde{\mathbf{x}} = (1, \tilde{x}_1, \ldots, \tilde{x}_q)$. For continuous response $y$, the residual sum of squares is the 'natural' loss criterion and we rely on the classical linear model

$$m_\theta(\tilde{\mathbf{x}}_i) = \sum_{j=0}^{q} \theta_j \tilde{x}_{ij} = \tilde{\mathbf{x}}_i\theta, \text{ for observations } i = 1, \ldots, n. \tag{9}$$

The notion behind ridge regression [14] is to estimate the parameter vector $\theta$ by minimizing $S$ from (8) with respect to $\theta$. Setting derivatives to zero leads to $(q+1)$ linear equations, which can be solved as

$$\hat{\theta} = (\widetilde{X}^T\widetilde{X} + \frac{n}{2}\lambda P)^{-1} \cdot \widetilde{X}^T y,$$

representing an explicit solution for minimizing $S$ in 8. Thus, the Newton-Raphson approximation is not necessary, and we directly obtain the exact solution. Software for treating continuous response problems is also contained in our R-package `supclust`.

## 4 Numerical Results

We evaluated our supervised grouping algorithms on several different datasets, all describing the gene expression of cancer patients. In particular, we analyzed:

- *The leukemia dataset of Golub et al. [15]:*
  This dataset contains gene expression levels of $n = 72$ patients either suffering from acute lymphoblastic leukemia (ALL, 47 cases) or acute myeloid leukemia (AML, 25 cases) and was obtained from Affymetrix oligonucleotide microarrays. Available at `http://www.genome.wi.mit.edu/MPR` are a training set of 38 observations and a test set of 34 samples. Following the protocol in [16], we preprocess the data by thresholding, filtering, a base 10 log-transformation and standardization, so that the data finally comprise the expression values of $p =$3,571 genes.
- *The estrogen and nodal datasets of West et al. [17]:*
  These datasets monitor $p =$7,129 genes in 49 breast tumor samples and were obtained by applying the Affymetrix technology. They are available at `http://mgm.duke.edu/genome/dna_micro/work/`. After thresholding to a floor of 100 and a ceiling of 16,000 expression units, we applied a base 10 log-transformation and standardized each experiment to zero mean and unit variance. Two response variables are available: one describing the status of the estrogen receptor and the other coding for the lymph node involvement. The two datasets are referred to as estrogen and nodal.
- *The colon cancer dataset of Alon et al. [18]:*
  This dataset was obtained from the Affymetrix technology and shows expression levels of 40 tumor and 22 normal colon tissues for a selection of 2,000 genes with highest minimal intensity across the samples. It is available at `http://microarray.princeton.edu/oncology/`. We process these data further by a base 10 log-transformation and standardization of each experiment to zero mean and unit variance across genes.
- *The prostate cancer dataset of Singh et al. [19]:*
  Available at `http://www-genome.wi.mit.edu/MPR/prostate`, these data comprise the expression of 52 prostate tumor and 50 non-tumor prostate samples, obtained from the Affymetrix technology. We use normalized and thresholded data as described in [19], leaving us with the base 10 log-transformed expression of $p =$6,033 genes, for each experiment standardized to zero mean and unit variance across genes.
- *The lymphoma dataset of Alizadeh et al. [20]:*
  This dataset contains cDNA microarray gene expression levels of the $K = 3$ most prevalent adult lymphoid malignancies. The sample size is $n = 62$, the data are available at `http://llmpp.nih.gov/lymphoma/data/figure1`. The expression of 4,026 accurately measured genes, either preferentially expressed in lymphoid cells or with known immunological or oncological importance

is documented. We imputed missing values and standardized the data as described in [16].

### 4.1 Typical Output

Generally, the output of *Pelora* looks very promising. In two-class datasets, each group centroid $\tilde{x}_j$, for $j = 1, \dots, q_{final}$, perfectly discriminates the two response classes. As an example, the 2-dimensional projection in figure 3 impressively shows how well the group centroids separate between the three different tissue types of the lymphoma dataset. The plot suggests that our group centroids are very suitable to predict the tissue types. Indeed, they allow error-free classification of training data and as shown in section 4.2, they also yield good results on independent test data.

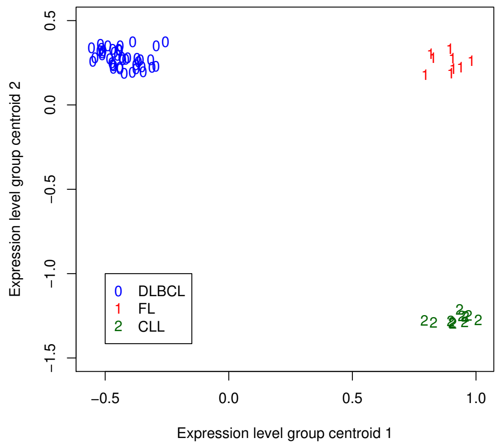


Fig. 3. 2-dimensional projection of lymphoma data: group centroid $\tilde{x}_1^{(0)}$ for discrimination of class 0 versus the classes 1 and 2 is on the $x$-axis, and $\tilde{x}_1^{(2)}$ for separation of class 2 versus classes 0 and 1 is on the $y$-axis.

The typical group size with *Pelora* is between 10-20 genes, table 1 reports average and standard deviation of the number of grouped genes for the first $q = 10$ groups in each dataset, obtained from *Pelora* with $\lambda = 1/32$. Note that the choice of the parameters $q$ and $\lambda$ is discussed in section 4.2 on page 21. The group size slightly diminishes with stronger penalization (increasing $\lambda$), but the differences are not very big. Note that with *Wilma*, our supervised algorithm from [3], the groups were smaller and contained on average only between 5-7 genes. This may be caused by the fact that *Wilma* is running under stronger supervision and has a grouping criterion which is less smooth than the one of *Pelora*.

It is beyond the scope of our paper to judge the functional relevance and the

| *Group size* | Colon | Leuke | Estro | Nodal | Prost | Lymph |
|---|---|---|---|---|---|---|
| mean | 14.0 | 12.1 | 15.4 | 14.8 | 17.9 | 15.8 |
| standard dev. | 5.3 | 3.2 | 4.2 | 4.3 | 9.0 | 3.5 |

Table 1
Group size: average and standard deviation of $q = 10$ groups from *Pelora* with $\lambda = 1/32$, for colon, leukemia, estrogen, nodal, prostate and lymphoma data.

biological meaning of *Pelora's* output. Instead, we collect empirical evidence that the group centroids are very informative for sample classification and perform at least as good as established methods based on single genes.

### *4.2 Predictive Potential*

By our supervised grouping algorithm *Pelora*, sample classification is straightforward, as it comprises a built-in classifier. In general, a classifier is a function that assigns a class label, based on observed features $x$. Here, these features will be the group centroids $\widetilde{x}_1, \ldots, \widetilde{x}_q$ and class label prediction is done with *Pelora's* conditional probabilities $p_\theta(\widetilde{\mathbf{x}})$ via

$$\widehat{y}(\widetilde{\mathbf{x}}) = \begin{cases} 0, & \text{if } p_\theta(\widetilde{\mathbf{x}}) \leq 1/2 \\ 1, & \text{if } p_\theta(\widetilde{\mathbf{x}}) > 1/2. \end{cases}$$

In multiclass problems, when using the one-against-all approach from section 3.4.3, the built-in classifier works by a maximum-likelihood principle. We obtain conditional class probabilities $p_\theta(\widetilde{\mathbf{x}}^{(k)})$ for every binary problem $k = 0, \ldots, K-1$ and assign the class label

$$\widehat{y}\left(\widetilde{\mathbf{x}}^{(0)}, \ldots, \widetilde{\mathbf{x}}^{(K-1)}\right) = \arg\max_k p_\theta\left(\widetilde{\mathbf{x}}^{(k)}\right).$$

Instead of working with the built-in classifier, we could also use the group centroids $\widetilde{x}_1, \ldots, \widetilde{x}_q$ as input for alternative methods like the nearest-neighbor rule [16], (possibly restricted) linear or quadratic discriminant analysis [16] or support vector machines [21], and many more. However, extensive experimentation (data not shown) yielded no improvement with these alternative methods compared to the built-in classifier.

In practice, the supervised groups and the built-in classifier are fitted on a learning set of tissues whose class labels are known. Subsequently, they can be used to predict the class labels of new tissues with unknown outcome. Since all the methodology for the grouping and the built-in classifier have been described earlier, we focus now on the only issue that remains, the choice of

| | $q = 2$ | $q = 4$ | $q = 6$ | $q = 8$ | $q = 10$ |
|---|---|---|---|---|---|
| $\lambda = 1$ | 23.54% | 16.62% | 14.15% | 13.54% | 12.77% |
| $\lambda = 1/2$ | 16.31% | 13.69% | 12.62% | 11.08% | 10.62% |
| $\lambda = 1/4$ | 13.85% | 10.77% | 9.54% | 8.77% | 8.00% |
| $\lambda = 1/8$ | 9.08% | 8.31% | 7.23% | 7.54% | 7.23% |
| $\lambda = 1/16$ | 7.08% | 7.54% | 7.54% | 7.54% | 6.77% |
| $\lambda = 1/32$ | 8.77% | 6.92% | 6.77% | 6.31% | 5.69% |
| $\lambda = 0$ | 9.54% | 10.00% | 10.00% | 10.00% | 10.00% |

Table 2
Misclassification rates for *Pelora's* built-in classifier with different parameter values $\lambda$ and $q_{final}$, based on 50 random splits of the leukemia training dataset into learning sets of 25 observations and validation sets of 13 tissues.

*Pelora's* two free parameters: the number of groups $q_{final}$ and the penalty parameter $\lambda$. For a fair evaluation of the predictive potential, tuning parameters should not be chosen such that the prediction results on the test data are optimized. This often leads to a considerable selection bias and does not reflect the practical situation where we have to predict the class labels of new patients' samples with unknown outcome.

As an example, we show here how to tune $q_{final}$ and $\lambda$ in a honest manner on the leukemia training dataset comprising 38 observations. The idea is to mimic out-of-sample classification by randomly splitting the training data into a learning set of 25 observations and a validation set of 13 observations. We fit *Pelora* on the learning set using all combinations of parameter values $q_{final} \in \{1, 2, \ldots, 10\}$ and $\lambda \in \{1, \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \frac{1}{16}, \frac{1}{32}, 0\}$, and then estimate the prediction accuracy by computing the fraction of misclassified individuals on the validation set. We repeat the splitting 50 times and average the misclassification rates, see table 2 and figure 4. The optimal parameter values, leading to the lowest error-rates on the leukemia training data, are $q_{final} = 10$ and $\lambda = \frac{1}{32}$. We now use *Pelora's* groups and the built-in classifier with these parameters to predict the original leukemia test dataset comprising 34 observations. We observe that only 1 sample is wrongly classified, a result which meets the state-of-the-art reported in the literature. Note that penalized logistic regression without any variable selection as in [6] yielded 3 false predictions, whereas the the combination of penalized logistic regression and recursive feature elimination proposed in [7] also achieved our result of 1 misallocation.

Figure 4 contains a graphical overview of the results we obtained for different parameter values. We observe that the predictive potential is poor with very few groups, then improves with increasing number of groups and stabilizes when more than 6 groups are used. Of course, a much larger number of groups would exhibit overfitting and result in poor prediction. Moreover, the

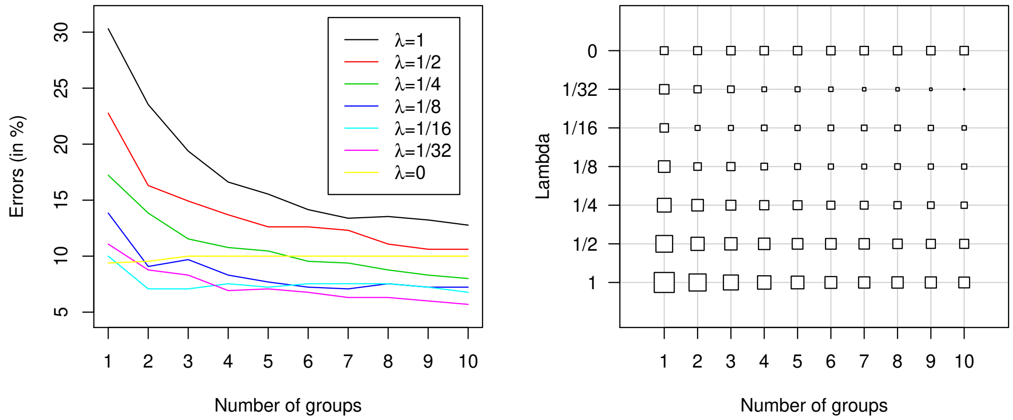


Fig. 4. Graphical representation of misclassification rates for *Pelora's* built-in classifier with different parameter values $\lambda$ and $q_{final}$, based on 50 random splits of the leukemia training dataset into learning sets of 25 observations and validation sets of 13 tissues. In the right panel, the size of the squares corresponds to the magnitude of the misclassification error.

correct amount of penalization drastically improves the classification. Without penalization ($\lambda = 0$), the error-rates are almost twice as high as with moderate $\lambda \in [\frac{1}{32}, \frac{1}{8}]$. Too strong penalization with $\lambda \geq \frac{1}{4}$ again degrades the classification. In general, the choice of the parameters is not too difficult, as the misclassification rates do not fluctuate wildly and are close to optimal over a larger range of $q_{final}$ and $\lambda$.

Tables and figures for all the other datasets cannot be displayed here due to space constraints. However, the full information is available from our webpage `http://stat.ethz.ch/~dettling/supervised.html`. The results for the other datasets are qualitatively equivalent, and the conclusions drawn from table 2 and figure 4 also hold there. After extensive experimentation, we determine the parameters $q_{final} = 10$ and $\lambda = \frac{1}{32}$ as default values, with which we will run *Pelora* on datasets where no independent test sets are available.

### *4.3 Comparison to Other Methods*

In this section, we compare the predictive potential of *Pelora's* built-in classifier with our former supervised grouping algorithm *Wilma* [3], the forward selection approach *Forsela* as presented in section 3.4.5, and three classifiers that are working with single genes as input. Since, except for the leukemia dataset, no genuine test sets are available, we base this comparison on repeated random splits into learning sets comprising two thirds, and validation sets containing one third of the training data. We do not run out-of-sample tuning to optimize the prediction results, but instead rely on fixed default parameters. For *Pelora*, we use the built-in classifier with default values $q_{final} = 10$ and $\lambda = \frac{1}{32}$. Our supervised grouping algorithm *Wilma* from [3], which does not comprise an internal classifier, is used with $q = 10$ group centroids as input for the 1-nearest-neighbor rule. Extensive experimentation (data not shown) with *Forsela* showed that $\lambda = \frac{1}{32}$ and $q = 30$ predictor variables (single genes) are reasonable default parameters for this technique. Finally, we compare the predictive potential of the group centroids with benchmark classification methods based on single genes.

For the benchmark methods, we select the 200 individually most predictive genes by the Wilcoxon statistic on the learning data (for each random split into training and validation data). In multiclass problems, this gene preselection consists of selecting the 200 most predictive genes for every binary discrimination. Note that this number has been recognized as a reasonable value in the broad evaluation of Dudoit et al. [16], and that *Pelora* is working with a similar number of genes, as it relies on 10 groups containing on average around 20 genes. The classifiers that are used with these 200 genes as input are the default 1-nearest-neighbor rule and diagonal linear discriminant analysis, which were the best classifiers in Dudoit et al.'s comparison study on microarray data [16]. As the state-of-the-art in modern classification, we also employ a support vector machine (from the R-package `e1071`) with radial basis kernel. We here rely on its default settings, although this flexible classifier may yield better results after sophisticated fine tuning.

| | Colon | Leuke | Estro | Nodal | Prost | Lymph |
|---|---|---|---|---|---|---|
| Pelora | 15.71% | 5.69% | 11.50% | 27.88% | 8.94% | 0.76% |
| Wilma | 16.48% | 2.62% | 8.75% | 35.88% | 8.06% | 0.57% |
| Forsela | 13.81% | 4.15% | 11.88% | 35.25% | 8.24% | 0.48% |
| NNR 200 | 15.90% | 2.46% | 15.38% | 43.25% | 12.82% | 0.67% |
| DLD 200 | 13.33% | 2.62% | 9.50% | 36.12% | 15.82% | 0.67% |
| SVM 200 | 17.62% | 0.92% | 11.12% | 36.88% | 8.35% | 0.48% |

Table 3
Misclassification rates for our supervised grouping algorithms *Pelora* and *Wilma*, the forward selection approach *Forsela* based on penalized logistic regression, as well as for the 1-nearest-neighbor rule (NNR), diagonal linear discriminant analysis (DLD) and support vector machines (SVM) with the 200 individually most predictive genes for 6 different datasets. All error-rates are means from 50 random splits into learning set ($\frac{2}{3}$ of data) and validation set ($\frac{1}{3}$ of data).

According to table 3 and figure 5, the predictive potential of supervised groups' centroids is convincing. We observe that our former implementation *Wilma* has an edge over *Pelora* in the four "easier" datasets leukemia, estrogen, prostate and lymphoma, but performs worse on the colon and nodal data. The improvement with our new method is thus not just on the methodological side, but also with regard to the prediction results in classification problems with substantial Bayes risk. This is most likely due to more robustness in *Pelora*, that is, weaker influence of the response $y$ in gene grouping.

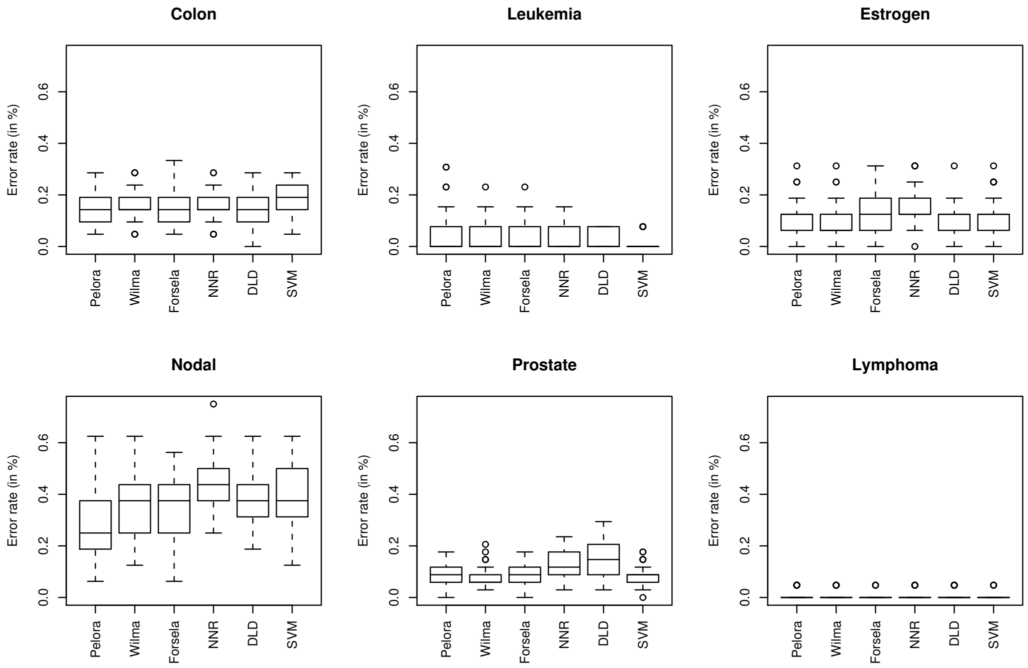


Fig. 5. Box and whisker plots, showing the variation of the misclassification rates over 50 random splits into learning set ($\frac{2}{3}$ of data) and validation set ($\frac{1}{3}$ of data) for 6 different classifiers: *Pelora* and *Wilma* with $q = 10$ groups, *Forsela* with $q = 30$ single genes, as well as the 1-nearest-neighbor rule (NNR), diagonal linear discriminant analysis (DLD) and a support vector machine (SVM), based on 200 single genes.

The forward selection approach *Forsela*, based on penalized logistic regression without any averaging, compares surprisingly favorably against *Pelora* and all the other methods. It yields low error-rates throughout, except for the leukemia and nodal data. The observation that *Pelora* is better than *Forsela* on the difficult nodal data set is probably due to the fact that the group centroids in *Pelora* are low-variance predictors yielding smaller variability in out-of-sample predictions; see also section 3.4.6.

The benchmark methods, diagonal linear discriminant analysis, the 1-nearest-neighbor-rule and support vector machines, perform similarly as *Pelora*, but slightly worse than *Wilma* and *Forsela*. This means that we have collected quite a bit of empirical evidence that our supervised grouping approaches yield gene groups which are valuable for sample classification. But both *Wilma* and *Pelora* should not only be seen as pure prediction tools. They partition thousands of genes into a few small groups that contain very useful information for explaining the outcome $y$. This is certainly an interesting dimensionality reduction and the gene groups may yield a clue on how the genome works with respect to certain diseases, and they can be used as a starting point to reveal functional gene groups or regulatory gene sub-networks.

### *4.4 Significance of Group Centroids and Clinical Variables*

For obtaining a prediction model that combines microarray data and clinical covariates, we described in section 3.4.4 how *Pelora* incorporates clinical variables into the grouping process. Here, we analyze how much prediction information is contained in the group centroids and the covariates. For illustration, we rely on the breast cancer dataset of van't Veer et al. [22]. Its training dataset contains expression values of 5,408 genes from red/green cDNA microarrays for 78 patients: 34 who developed metastases within 5 years, and 44 who remained disease-free during this period. Furthermore, information about 6 covariates is provided, which in clinical practice is used to decide upon therapy. In particular, these variables are the tumor grade $\in \{1, 2, 3\}$, the estrogen receptor status $\in [0, 100]$, the progesteron receptor status $\in [0, 100]$, the tumor size in millimeters, the patient age and angioinvasion $\in \{0, 1\}$.

When using *Pelora* with default $\lambda = \frac{1}{32}$ on the combined breast cancer expression and clinical data, we observe that none of the clinical variables entered the model, even if the number of predictors was raised to $q_{final} = 30$. This is in line with the findings in van't Veer et al. [22] and can be interpreted that the clinical covariates, compared to the expression profile, do not contain much useful information for class prediction.

Note that in other datasets, where more strongly predictive clinical variables are available, we may observe a mixture of group centroids and covariates already among the first 10 predictors identified by *Pelora*. To simulate this situation and to exemplify how one can determine which predictors contribute significantly to sample classification, we artificially reduced the breast cancer dataset to 1141 arbitrarily chosen genes. Then, among the first 10 predictors *Pelora* selected, are the intercept, six gene groups and 3 clinical variables. In order of selection, the latter are tumor grade, patient age and angioinvasion.

To answer the question whether some of these clinical covariates, and which of the group centroids, contribute significantly to sample classification, we do bootstrap-based statistical inference on an independent breast cancer test dataset, which contains the expression values and clinical data of 19 additional patients: 7 who remained metastasis-free for 5 years and 12 who experienced disease progression. By using only the model-structure from the training data, we fitted penalized logistic regression as in section 3.4.1 on the test dataset

| predictor | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| variable | intercept | clinical | group | clinical | group |
| $p$-value | 0.012 | 0.000 | 0.000 | 0.000 | 0.136 |
| predictor | 5 | 6 | 7 | 8 | 9 |
| variable | group | group | group | clinical | group |
| $p$-value | 0.084 | 0.008 | 0.146 | 0.024 | 0.022 |

Table 4
Bootstrap $p$-values for the coefficients of *Pelora's* prediction model on the breast cancer data with 1141 arbitrarily chosen genes. Variables 2, 4, 5-7 and 9 are group centroids, variable 1 is the tumor grade, variable 3 is the patient age and variable 8 is angioinvasion.

and obtained the parameter vector $\widehat{\theta}^{test} = (\widehat{\theta}_0^{test}, \dots, \widehat{\theta}_q^{test})$. To get an impression about the distribution and variability of these coefficients, we generate 1,000 non-parametric bootstrap samples from the test data by drawing with replacement: every run $b \in \{1, \dots, 1000\}$ yields an estimated parameter vector $\widehat{\theta}^{(b)} = (\widehat{\theta}_0^{(b)}, \dots, \widehat{\theta}_q^{(b)})$. For quantifying the significance of each predictor variable, we computed the $(1-\alpha)$-bootstrap confidence intervals

$$[2 \cdot \widehat{\theta}_j^{test} - q_{j,(1-\frac{\alpha}{2})}; 2 \cdot \widehat{\theta}_j^{test} - q_{j,\frac{\alpha}{2}}],$$

where $q_{j,\alpha}$ is the $\alpha$-quantile of the bootstrap distribution. Inverting these intervals leads to the $p$-values reported in table 4. For the reduced breast cancer dataset with 1141 genes, all fitted predictor variables except for 3 group centroids turned out to be significant at the 5%-level.

## 5 Conclusions

We have presented methodology for finding predictive molecular gene signatures from microarray data by using supervised grouping techniques. This is potentially beneficial in medical diagnostics and prognostics, as the identified signature groups are made up of interacting genes whose expression centroids have high explanatory power for the response variable. These groups of genes and their centroids can in turn be used to accurately predict the outcome of new samples. But supervised grouping should not be seen as a pure prediction tool: it partitions thousands of genes into a few small gene groups which amounts to a drastic dimensionality reduction. Moreover, groups of genes may yield more important biological insights than single genes, for example as valuable first information about gene function and regulation.

From a more technical viewpoint, our novel supervised grouping algorithm *Pelora* combines supervised gene selection, gene grouping and optional sample classification in a single-step approach. Its goal is to find groups of genes whose centroids render the discrimination of the outcome $y$ as simple as possible. We solve this by building the groups incrementally in a combination of forward steps and regularly recurring cleaning steps. All grouping operations are based on an empirical objective function that includes information from the $y$-values and is based on conditional class probabilities computed from penalized logistic regression analysis. By using these probability estimates, *Pelora* also comprises a built-in classifier that exploits the gene group centroids.

*Pelora* improves many of the limitations of *Wilma*, our first implementation of supervised grouping. It also allows to capture genes operating in multiple pathways, as it does not require disjointness of its groups. By using a grouping criterion that is based on multiple groups, we can expect to find a team of interacting groups instead of a cohort of individual players as with *Wilma*. Moreover, we have proposed extensions of *Pelora* to polytomous and continuous response problems, to a forward selection technique for genes without any averaging, as well as a combination with additional clinical covariates. But *Pelora* does not only convince by its neat features or its coherent algorithm which is based on sound statistical methodology within the likelihood framework: with an extensive empirical study on a variety of microarray gene expression datasets, we provide empirical evidence that *Pelora*'s predictive potential can keep up with established classifiers and state-of-the-art machine learning methods, and has a great potential to improve them on difficult datasets with high misclassification risk. Although *Pelora* was specifically developed for the analysis of microarray data, it may be useful for other data that are subject to the "large $p$, small $n$" problem and where a few underlying groups of explanatory variables are expected to determine most of the outcome variation.

## A Proof: Penalized Logistic Regression with Non-Unit Penalty

Here, we prove that penalized logistic regression with non-standardized predictor $\tilde{\mathbf{x}} = (1, \tilde{x}_1, \ldots, \tilde{x}_q)$ and the non-unit penalty matrix $P$ from (7) yields equivalent parameter estimates and the same fitted values as when working with the unit penalty matrix $Q = diag(0, 1_{q \times q})$ and standardized predictor $\tilde{\mathbf{u}} = (\frac{1}{s_0}, \frac{\tilde{x}_1}{s_1}, \ldots, \frac{\tilde{x}_q}{s_q})$, where $s_0 = 1$ per definition and $s_j$, for $j = 1, \ldots, q$, is the (empirical) standard deviation of $\tilde{x}_j$. The classical logistic model can then be formulated equivalently as

$$\log\left(\frac{p_\theta(\tilde{\mathbf{x}}_i)}{1 - p_\theta(\tilde{\mathbf{x}}_i)}\right) = \sum_{j=0}^{q} \theta_j \tilde{x}_{ij} = \sum_{j=0}^{q} \gamma_j \tilde{u}_{ij} = \log\left(\frac{p_\gamma(\tilde{\mathbf{u}}_i)}{1 - p_\gamma(\tilde{\mathbf{u}}_i)}\right), \tag{A.1}$$

with parameters $\theta = (\theta_0, \dots, \theta_q)^T$ and $\gamma = (\gamma_0, \dots, \gamma_q)^T$, where $\gamma_j = \theta_j s_j$ for $j = 0, \dots, q$. From (A.1) it follows that $p_\theta(\tilde{\mathbf{x}}_i) = p_\gamma(\tilde{\mathbf{u}}_i)$. Estimates of the parameters are then obtained by penalized maximum likelihood via

$$\widehat{\theta} = \arg\min_\theta - \sum_{i=1}^n (y_i \cdot \log p_\theta(\tilde{\mathbf{x}}_i) + (1 - y_i) \cdot \log(1 - p_\theta(\tilde{\mathbf{x}}_i))) + n\frac{\lambda}{2}\theta^T P\theta$$
$$\widehat{\gamma} = \arg\min_\gamma - \sum_{i=1}^n (y_i \cdot \log p_\gamma(\tilde{\mathbf{u}}_i) + (1 - y_i) \cdot \log(1 - p_\gamma(\tilde{\mathbf{u}}_i))) + n\frac{\lambda}{2}\gamma^T Q\gamma.$$

Now, by using $p_\theta(\tilde{\mathbf{x}}_i) = p_\gamma(\tilde{\mathbf{u}}_i)$ and the equality $\theta^T P\theta = \gamma^T Q\gamma$, we obtain $\widehat{\gamma}_j = \widehat{\theta}_j s_j$, from which the claim follows.

## References


[1] D. Nguyen, D. Rocke, Tumor classification by partial least squares using microarray gene expression data, Bioinformatics 18 (2002) 39–50.

[2] T. Hastie, R. Tibshirani, D. Botstein, P. Brown, Supervised harvesting of expression trees, Genome Biology 2 (2001) research 0003.1–0003.12.

[3] M. Dettling, P. Bühlmann, Supervised clustering of genes, Genome Biology 3 (2002) research 0069.1–0069.15.

[4] R. Jörnsten, B. Yu, Simultaneous gene clustering and subset selection for sample classification via MDL, Bioinformatics 19 (2003) 1100–1109.

[5] S. Le Cessie, J. Van Houwelingen, Ridge estimators in logistic regression, Applied Statistics 41 (1990) 191–201.

[6] P. Eilers, J. Boer, G.-J. Van Ommen, H. Van Houwelingen, Classification of microarray data with penalized logistic regression, in: Proceedings of SPIE: Progress in Biomedical Optics and Imaging, Vol. 2, 2001, pp. 187–198.

[7] J. Zhu, T. Hastie, Classification of gene microarrays by penalized logistic regression, Tech. rep., Department of Statistics, University of Stanford (2002).

[8] P. Bickel, C. Klaassen, Y. Ritov, J. Wellner, Efficient and Adaptive Estimation for Semiparametric Models, Johns Hopkins University Press, 1993.

[9] S. Dudoit, J. Fridlyand, A prediction-based resampling method to estimate the number of clusters in a dataset, Genome Biology 3 (2002) research 0036.1–0036.21.

[10] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a dataset via the gap statistic, Tech. Rep. 208, Department of Statistics, University of Stanford (2000).

[11] M. Dettling, P. Bühlmann, Boosting for tumor classification with microarray data, Bioinformatics 19 (2003) 1061–1069.

[12] E. Allwein, R. Schapire, Y. Freund, Reducing multiclass to binary: A unifying approach for margin classifiers, Journal of Machine Learning Research 1 (2000) 113–141.

[13] E. Huang, S. Chen, H. Dressman, J. Pittman, M. Tsou, C. Hong, A. Bild, E. Iversen, M. Liao, C. Chen, M. West, J. Nevins, A. Huang, Gene expression predictors of breast cancer outcomes, The Lancet 361 (2003) 1590–1596.

[14] A. Hoerl, R. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, Technometrics 12 (1970) 55–67.

[15] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gassenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caliguri, C. Bloomfield, E. Lander, Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring, Science 286 (1999) 531–538.

[16] S. Dudoit, J. Fridlyand, T. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data, Journal of the American Statistical Association 97 (2002) 77–87.

[17] M. West, C. Blanchette, H. Dressman, E. Huang, S. Ishida, R. Spang, H. Zuzan, J. Olson, J. Marks, J. Nevins, Predicting the clinical status of human breast cancer by using gene expression profiles, Proceedings of the National Academy of Science 98 (2001) 11462–11467.

[18] U. Alon, N. Barkai, D. Notterdam, K. Gish, S. Ybarra, D. Mack, A. Levine, Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, Proceedings of the National Academy of Science 96 (1999) 6745–6750.

[19] D. Singh, P. Febbo, K. Ross, D. Jackson, J. Manola, C. Ladd, P. Tamayo, A. Renshaw, A. D'Amico, J. Richie, E. Lander, M. Loda, P. Kantoff, T. Golub, W. Sellers, Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2002) 203–209.

[20] A. Alizadeh, M. Eisen, E. Davis, C. Ma, I. Lossos, A. Rosenwald, J. Boldrick, H. Sabet, T. Tran, X. Yu, J. Powell, L. Yang, G. Marti, T. Moore, J. Hudson, L. Lu, D. Lewis, R. Tibshirani, G. Sherlock, W. Chan, T. Greiner, D. Weisenburger, J. Armitage, R. Warnke, R. Levy, W. Wilson, M. Grever, J. Byrd, D. Botstein, P. Brown, L. Staudt, Distinct types of diffuse large b-cell-lymphoma identified by gene expression profiling, Nature 403 (2000) 503–511.

[21] T. Furey, N. Cristianini, N. Duffy, D. Bednarski, M. Schummer, D. Haussler, Support vector machine classification and validation of cancer tissue samples using microarray expression data, Bioinformatics 16 (2000) 906–914.

[22] L. Van't Veer, H. Dai, M. Van de Vijver, Y. He, A. Hart, M. Mao, H. Peterse, K. Van der Kooy, M. Marton, A. Witteveen, G. Schreiber, R. Kerkhoven, C. Roberts, P. Linsley, R. Bernards, S. Friend, Gene expression profiling predicts clinical outcome of breast cancer, Nature 415 (2002) 530–535.