

Diss. ETH No. 15580

Supervised Learning in Very High Dimensional Problems with Applications to Microarray Data

A dissertation submitted to the
SWISS FEDERAL INSTITUTE OF TECHNOLOGY
ZURICH

for the degree of
Doctor of Mathematics

presented by
MARCEL DETTLING
Dipl. Math. ETH
born October 21, 1974
citizen of Oberiberg SZ

accepted on the recommendation of
Prof. Dr. Peter Bühlmann, examiner
Prof. Dr. Eckart Zitzler, co-examiner

2004

Acknowledgements

First of all, I would like to thank Prof. Peter Bühlmann for proposing the subject and for supervising this thesis with such great interest. He has optimally supported me with advice and ideas, and I was always welcome to discuss my work and future directions with him. He has been an excellent mentor, introducing me as a scientist, encouraging me to travel and bringing me into contact with many important researchers.

I also thank Prof. Eckart Zitzler for refereeing this thesis and for accepting to act as the co-examiner.

Further thanks go to all my colleagues at the “Seminar für Statistik” for their motivating support in statistics and computer science, as well as for the good atmosphere, the many enjoyable events and inspiring discussions.

Finally, I would like to thank my family and my friends for their constant support and the great time we spent outside of the office.

Contents

Abstract	ii
Zusammenfassung	iv
Introduction	1
Supervised Clustering of Genes	20
Finding Predictive Gene Groups	60
Boosting for Tumor Classification	97
BagBoosting for Tumor Classification	120
Outlook	146
Bibliography	147
Curriculum Vitae	156

Abstract

In the past decade, the advent of efficient genome sequencing tools and high-throughput experimental biotechnology has led to enormous progress in the life sciences. Among the most important innovations is the microarray technology. It allows to quantify the expression for thousands of genes simultaneously by measuring the hybridization from a tissue of interest to probes on a small glass or plastic slide. The characteristics of these data include a fair amount of random noise, a predictor dimension in the thousands, and a sample size in the dozens.

A particular application of the microarray technology is in cancer research, where the goal is a precise and early diagnosis of tumorous malignancies, allowing for a tailored treatment with less side-effects and higher cure rates. The challenge for statistical research is the development and adaptation of class prediction tools that reliably work in this very-high dimensional situation. The problem may be relaxed to some extent by the fact that the true underlying signal may be sparse, meaning that only a few genes significantly contribute to the outcome variation.

This thesis contributes with two papers that pursue the novel concept of finding predictive gene groups from microarray data. It is motivated from the biological assumption that a few latent gene expression signatures are most accurate for phenotype discrimination. We present two algorithms that are based on non-exhaustive, but efficient greedy search heuristics, plus two statistically motivated, likelihood-based objective functions. The competitive classification power of these parsimonious prediction models has been carefully evaluated and empirically confirms the benefit of these supervised grouping techniques.

Two further chapters of this thesis focus on statistically motivated machine learning methods for class prediction with gene expression data. The first contribution is a tailored boosting algorithm that contradicts and clarifies the statement that boosting methods do not work well for microarray data, as observed in several earlier publications. The second paper suggests a completely novel hybrid approach between the two ensemble methods bagging and boosting. This modification results in an algorithm performing among the best within the machine learning methods. Moreover, the second paper presents some innovative ideas about measuring the influence of single genes for a biological interpretation of the prediction models. The validity of these machine learning approaches has been confirmed by application on many real datasets and several simulation models.

Zusammenfassung

Die vergangenen zehn Jahre brachten die Entwicklung von effizienten, Sequenzierungs-Werkzeugen für die Genomik, sowie die Entwicklung von experimentellen Biotechnologien mit grosser Bandbreite. Dies führte zu einem grossen Fortschritt in den sogenannten Life Sciences. Eine der wichtigsten Innovationen ist die Microarray-Technologie. Sie erlaubt eine Quantifizierung der Expression für Tausende von Genen gleichzeitig, durch die Messung der Hybridisierung auf einer kleinen Glas- oder Plasticscheibe. Die Charakteristik dieser Daten umfasst einen ansehnlichen Anteil von zufälliger Variabilität, mehrere Tausend Prädiktor-Variablen, aber nur einige Dutzend Experimente.

Eine populäre Anwendung der Microarray-Technologie liegt in der Krebsforschung. Das Ziel ist eine präzise Früherkennung von Tumorkrankheiten, die eine massgeschneiderte Behandlung mit einem Minimum an Nebenwirkungen und besseren Genesungsraten ermöglichen soll. Die Herausforderung an die Statistik besteht in der Entwicklung und Anpassung von Methoden für die Klassifizierung auf solch hoch-dimensionalen Datensätzen. Die Schwierigkeiten werden auf eine gewisse Weise erleichtert, da das unterliegende biologische Signal dünn besetzt sein kann, d.h. nur eine begrenzte Anzahl von Genen ist für die Variabilität in der Zielvariablen verantwortlich.

Der Beitrag dieser Arbeit sind zwei Kapitel, welche das neuartige Konzept des Auffindens von prädiktiven Gen-Gruppen verfolgen. Die Motivation kommt von der biologischen Vermutung dass einige wenige, latente Gen-Expressions-Signaturen den stärksten Einfluss auf die Phänotyp-Diskriminierung haben. Es werden zwei Algorithmen präsentiert, die auf einer nicht ausschöpfenden, jedoch effizienten, schrittweisen Such-Heuristik basieren. Dazu kommen zwei statistisch motivierte Zielfunktionen, die auf dem Prinzip der maximalen Likelihood basieren. Die

Eignung solcher dünn besetzter Modelle für die Klassifizierung wird sorgfältig evaluiert und bestätigt empirisch den Nutzen solcher Gruppierungs-Techniken.

Zwei weitere Kapitel dieser Arbeit konzentrieren sich auf statistisch motiviertes, maschinelles Lernen für die Vorhersage von Krebsklassen aufgrund von Microarray-Daten. Der erste Beitrag ist ein speziell angepasster Boosting-Algorithmus, welcher der früher publizierten Aussage widerspricht, dass Boosting-Methoden für Microarray-Daten nicht geeignet seien. Ein zweiter Teil präsentiert einen völlig neuartigen Hybriden, der die beiden Methoden Bagging und Boosting vereint. Diese Kombination ergibt einen schlagkräftigen Algorithmus, der unter den besten seines Fachs ist. Zudem präsentiert diese zweite Arbeit auch einige innovative Ideen zur Messung des Gen-Einflusses auf die Klassifizierung und deren biologische Interpretation. Der Nutzen von solchen maschinellen Lern-Methoden wird durch Anwendung auf zahlreichen realen Datensätzen und simulierten Modellen bestätigt.

Introduction

The present doctoral thesis has been motivated by statistical learning problems that arise from the analysis of gene expression microarray data. Recent developments in biotechnology allow for measuring the activity *or* expression of many thousand genes simultaneously in a snapshot like manner, and is attested an enormous potential for progress in virtually every field of biomedical research and daily clinical practice. The ultimate goal of microarray experiments is to understand the processes within the cell and to learn the functional units in the genome.

Due to the time-consuming experimental protocol, the cost and the often limited access to biological tissue of interest, the majority of microarray experiments are performed on a small number of samples only. Together with the large-scale capacity of the technology, this creates a very difficult high-dimensional situation for statistical learning. We are facing many thousands of genes as predictor variables, while the sample size is only in the dozens. This renders much of the existing data analysis tools not working well or even infeasible, and requires their adaptation or the development of novel statistical methods. In the next few sections, the biological background is revisited and the route from the basics of the microarray technology to the challenges in statistical learning is presented.

1 Biological Background

The basic unit of all living organisms is the cell, containing the nucleus where the hereditary information is stored on the chromosomes, see Figure 1. The chromosomes are macro-molecules, made up from *desoxyribonucleic acid*. This chemical substance, abbreviated as DNA, codes the hereditary information in a double-stranded helix of the four bases A, T, G and C. Each of the chromosomes contains between several hundred and many

thousands of genes. A gene corresponds to a specific DNA fragment, and can be interpreted as a construction manual for a protein.

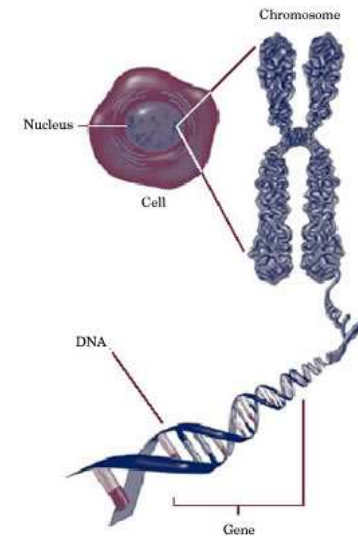


Figure 1: *An overview of cell, nucleus, chromosomes and DNA.*

To first approximation, all the cells in an organism contain the same set of chromosomes, and thus also the same set of genes. Despite the equal information, there is a wide variety in appearance: for example between skin, hair, blood, muscle or fat cells. Also cancerous and healthy cells emerge from the same information. How can this happen? The answer is given by the central dogma of molecular biology, illustrated in Figure 2. It claims that the way from gene to protein consists of two steps. First, the gene is transcribed into *messenger ribonucleic acid*, abbreviated as mRNA or shorter, RNA. Second, the mRNA is translated into a protein. It is important to note that there is huge variation in abundance and efficiency of transcription and translation among

different cell types, and that finally, the protein distribution is responsible for the appearance and the state of a cell.

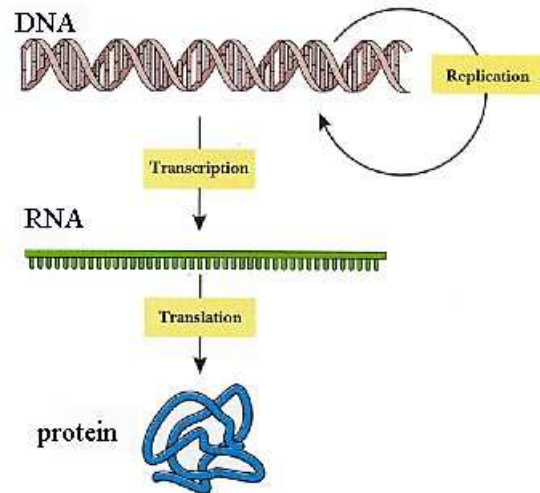


Figure 2: *DNA-RNA-Protein: the central dogma of molecular biology.*

For detecting differences between cells, it would be helpful to monitor their protein composition. This is a technically very demanding task that is not satisfactorily solved yet. The central dogma of molecular biology suggests that an alternative is to measure changes in the mRNA levels. Although most proteins undergo modification after translation and before becoming functional, the mRNA level bears a close relationship to the state of a cell, and thus makes the transcriptome worthy of systematic measurement.

2 Microarray Technology

There are several techniques for quantifying the amount of transcribed mRNA, all of them relying on the fundamental property of complementary base pairing. The most prominent tool are gene expression microarrays, allowing for a snapshot of the entire genome. Such a microarray is an approximately thumb-nail sized glass or plastic slide with many thousands of addressable spots, on each of which a separate experiment takes place: a genomic sequence is placed in over-abundance as a probe, and the amount of mRNA that hybridizes to it (i.e. chemically binds through complementary base pairing) during an experimental process can be measured. The work-flow in a microarray experiment encompasses the production of the arrays, the extraction of mRNA from tissue samples of interest, hybridization, scanning, data pre-processing, statistical analysis and biological verification. These steps are discussed below.

2.1 Experiment and Preprocessing

The two most prevalent microarray technologies are cDNA microarrays and high-density oligonucleotide arrays. They both rely on hybridization through complementary base pairing, but they differ in the manner of placement of the DNA sequences on the array. Accordingly, the experimental approach and the data pre-processing differ as well.

cDNA Arrays

The two most important characteristics of cDNA *or* spotted arrays are a), that each gene is represented by a DNA clone of full length, and b), that they rely on competitive hybridization. The latter means that mRNA is extracted from two different biological samples (i.e. a sample of interest and a control sample). It is then reverse transcribed into cDNA, labeled with red and green fluorescent dyes, and distributed on the microarray, where the cDNA competitively hybridizes to the corresponding DNA

clones. The remaining material is washed off and the amount of chemically bound cDNA is quantified by the intensity of the fluorescence in each spot, as measured by a laser scanner. The basic, raw data from a microarray experiment constitute of the pixel intensities from the scanner. These need to be segmented (i.e. the spots need to be defined) and the pixelized intensities need to be summarized into one value. As a first step, the data are then usually visually inspected by searching for artifacts such as print tip effects, spatially varying bias, edge effects, scratches, dust or saturation effects.

For each spot on the array, scanning and imaging yield a foreground value F_g for the green and F_r for the red channel. Due to the comparably large spaces between the spots, the background fluorescence affects the spot intensities and is also measured as B_g for the green and B_r for the red channel. The naive approach for obtaining an unbiased estimate of the green/red intensity is to subtract the background fluorescence B from the spot intensity F . As these are both estimated quantities, the variability of $F - B$ tends to be larger than the one of F alone. This has led to a controversial scientific debate on whether to perform background subtraction or not. The current situation is that background corrections are only recommended when major spatial differences are present. Finally, the (possibly) background corrected intensities are transformed into a gene expression measure by taking log-ratios, i.e.

$$x_g = \log_2 \left(\frac{F_{g,g} - B_{g,g}}{F_{r,g} - B_{r,g}} \right) \text{ for each gene } g \in \{1, \dots, p\}.$$

After this log-transformation, unchanged expression is zero, and both up-regulated and down-regulated genes can take values from zero to infinity. Taking log-ratios usually also yields a fairly symmetric distribution of the gene expressions.

High-density oligonucleotide chips

The most widely used high-density oligonucleotide microarray is the GeneChip, commercially distributed by Affymetrix. Its

main characteristics are a), that gene expression is quantified by non-competitive hybridization, meaning that only one biological sample (the sample of interest) is fluorescently labeled and hybridized to the microarray, and b), the expression of each gene is measured by comparing the hybridization to a set of 20 probe pairs, each of which is 25 base pairs long and synthesized *in situ* by photo-lithography. The first type of probe in each pair is the perfect match PM which exactly corresponds to the gene sequence, whereas the second is the mismatch MM , created by changing the middle (the 13th) base of the original sequence. The idea of this construction is to provide a control mechanism for random variation and cross-hybridization. The arrays are then scanned, and the output for each gene sequence of interest are two 20-dimensional vectors of intensity readings

$$\begin{aligned} PM_g &= (PM_{g,1}, \dots, PM_{g,20}), \\ MM_g &= (MM_{g,1}, \dots, MM_{g,20}), \end{aligned}$$

which are to be transformed into a gene expression. The default implementation is to rely on the log-transformed, direct average of the perfect match and mismatch differences,

$$x_g = \log_2 \left(\frac{1}{20} \sum_{j=1}^{20} (PM_{g,j} - MM_{g,j}) \right)$$

for each gene $g \in \{1, \dots, p\}$. Improved suggestions for summarizing the PM and MM vectors to a gene expression include the multiplicative model of Li and Wong [1] or the robust multi-array analysis of Irizarry et al [2]. More recent approaches propose to omit the MM -values at all [3], or to incorporate the binding affinity of the spotted PM sequences into the summarization [4].

2.2 Why using statistical methods?

After appropriate preprocessing, both cDNA arrays and oligo chips yield data with similar characteristics. Our notion of a

microarray experiment is given by a random vector

$$\mathbf{x} = (x_1, \dots, x_p),$$

containing the transcriptional activity for up to 25'000 genes. These gene expression measurements are subject to random variation, mainly because of the following four sources:

1) *Undesired biological variability*

When comparing the gene expression between tumors and normal tissue, the expression profile could be different if another patient had been analyzed, or even if another portion of tissue from the same patient had been used.

2) *Microarray manufacturing*

Especially cDNA arrays suffer from differences in the preparation of the DNA clones which are spotted, as well as from various artifacts which are caused by their placement on the microarray.

3) *mRNA preparation*

Variation is caused by the extraction of the mRNA, its subsequent amplification and purification. Additionally, the fluorescent dye binds to the genes with different efficiency.

4) *Hybridization*

An important error source is cross-hybridization, this is mRNA that wrongly binds to non-corresponding probes on the array. Moreover, extraneous factors such as temperature, exposure time and characteristics of the mRNA solution can have an effect on the binding ability.

Thus, the gene expression vector \mathbf{x} cannot be regarded as an exactly measured quantity, and data analysis has to rely on statistical methods. Finally, a microarray experiment is only complete when the gene expression profile \mathbf{x} comes along with a detailed description of the genes which are represented on the array, with information about the experimental protocol, the preprocessing and a vector of covariates. The latter may include the sample

phenotype such as its tumor class, the design variables of controlled experiments (for example drug treatment), plus further clinical parameters about the individual from which the sample was obtained. Examples include simple parameters such as age and sex, results from screening techniques like tumor size, lymph node involvement and metastasis, the response to initial medication or histological findings after invasive treatment.

2.3 Replication and Normalization

Due to the stochastic nature of gene expression data, it is now widely becoming accepted that microarray experiments need to be replicated. The question is how to compare and normalize the gene expression measures across these replicates, since besides the biological signal, they are also subject to variation from the manufacturing and hybridization process. This creates a need for normalization across arrays. Visualization, artifact detection and the exclusion of obviously bad arrays are the first steps. Subsequently, the basic formal approach is to center and scale the gene expressions with respect to an artificial reference array, obtained for example by taking the median expression for each gene across arrays. However, this is often inadequate when nonlinear distortions are present. An alternative is to use quantile normalization, where it is enforced that all the arrays have matching expression values at specified quantiles. It is important to note that such across-array normalization always bears a trade-off between improving the comparability of multiple arrays and obscuring the biological signal by making them too similar.

Because microarray experiments are costly and the accessibility of biological samples is often limited, the number of arrays n is usually small, leaving us with data

$$\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^{n \times p}.$$

Together with the very high predictor dimension p , this has raised challenging questions to computational statistics, but has also made this area to one of the most vibrant and exciting fields of

mathematical research. The list of successful applications is very long and encompasses work from nearly all fields of bio-medical research. The tools which are used for statistical evaluation of such high-throughput data vary according to the specific question of a particular experiment. The most important and widely used methods are briefly presented below.

2.4 Unsupervised Statistical Analysis

Microarrays produce an enormous amount of data which cannot be overlooked without summarizing and rearranging them in sensible ways. The purpose of unsupervised statistical analysis is to structure the data into a form which is readable by the human eye.

Heatmaps

The most prominent tool for visualizing gene expression data are so-called heatmaps, introduced in the seminal work of Eisen et al. [5]. In a heatmap, each row corresponds to a gene and each column to an array (or vice versa). The respective expression value is depicted as a colored rectangle, see Figure 3.

The color range usually goes from green for under-expressed genes, over black to red, indicating genes with higher expression. Both the genes and samples are often rearranged in a way that facilitates the detection of structures. This can for example be achieved with hierarchical clustering (see below), or with biological background information about genes and experiments. The scope of heatmap analyses is mainly limited to illustration, and the human eye has difficulties to detect more than the most obvious structures.

Clustering

Clustering algorithms organize objects into a small number of groups, where the objects within one group are more alike than the objects across groups. With microarray data, these objects

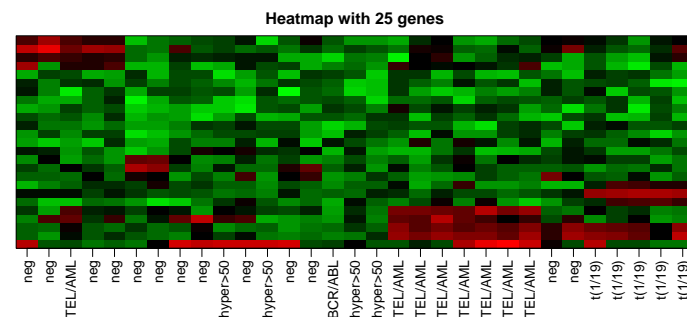


Figure 3: A heatmap showing 25 genes and 31 expression profiles from children with acute lymphoblastic leukemia. The samples have been rearranged according to a hierarchical clustering, the labels on the x-axis stand for the cytogenetical entity. Green means under-expression, red stands for over-expression.

can either be genes or samples. Clustering techniques have become a frequently used standard in the genomic literature, but they represent a very explorative form of data analysis and should be seen as the start, rather than as the end of a microarray study. The scope of clustering is often the visualization of relationship and distance among the objects. The interpretation of a clustering output usually employs the guilt-by-association principle: objects with unknown characteristics are assumed to share the properties of known objects they are grouped with.

Clustering techniques can be divided into hierarchical and non-hierarchical ones. The hierarchical clustering algorithm [5] is usually started bottom-up, with all objects as individual clusters, which are then merged. In each step, the algorithm joins the two closest clusters according to a distance matrix and a linkage method. This produces a nested series of clusters which is visualized by the so-called dendrogram, see Figure 4. The branch length between two objects represents their degree of similarity.

The most prevalent non-hierarchical clustering techniques for

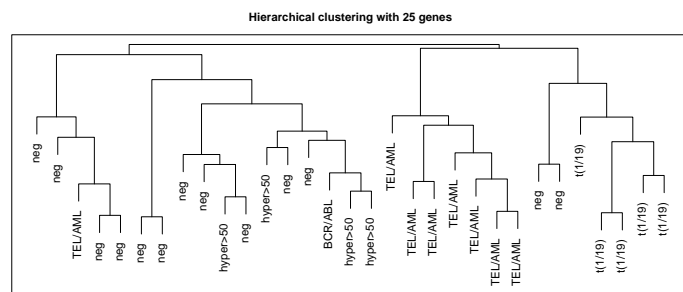


Figure 4: A dendrogram showing the hierarchy among 31 gene expression profiles from different cytogenetical entities, obtained from children with acute lymphoblastic leukemia.

microarray data are k -means clustering [6] and self-organizing maps [7]. Both require the number of clusters to be pre-specified by the user. The k -means algorithm tries to achieve maximum purity within and minimal similarity across its clusters. It is started by assigning the samples to random clusters and iteratively moving them around until the ratio of across-cluster variance to within-cluster variance can no longer be improved. Self-organizing maps are a technique which is related to neural networks. They usually begin with a dimension reduction step and detect similarities among objects by iteratively mapping similar input to similar regions of the output space.

Dimension Reduction

Dimension reduction tools try to summarize the massive amount of data from the thousands of genes into a few representative variables. While this often greatly improves the overview and the handling of the data, it usually still retains most of the information. The most prominent tool is principal component analysis [8]: its goal is to find a sequence of linear combinations that are uncorrelated and capture as much of the variability in the full

dataset as possible. While principal component analysis can be fairly successful in excluding redundant information, the biological interpretation of linear combinations obtained from (possibly) thousands of genes is most often difficult.

Gene Regulatory Networks

The detection of molecular signaling pathways and causal relations between genes recently drew a lot of attention. Inferring regulatory networks from gene expression data is a conceptually and computationally very demanding task. The dependencies between genes are often illustrated by a graph $G(V, E)$, with a vertex set V consisting of the genes, and an edge set E .

An undirected edge between genes i and j exists if V_i and V_j are conditionally dependent given all other variables $\{V_1, \dots, V_p\} \setminus \{V_i, V_j\}$. The notion of conditional stochastic dependence is important for excluding edges between two variables that are caused by a third, confounding variable V_k . The major problem with gene expression data is that the number of vertices is so much larger than the sample size. This requires structural assumptions on the network and/or simplifications in the statistical learning process.

2.5 Supervised Statistical Analysis

Supervised methods rely on a known, preliminary grouping of the samples, which guides through the statistical learning process. The two most important tasks under this heading are the identification of differentially expressed genes between two conditions or phenotypes, as well as class prediction, i.e. the assignment of samples based on their gene expression patterns into known categories.

Screening for Differentially Expressed Genes

A question of primary interest in biology is the identification of genes that change their expression in different populations, under

different conditions or for different phenotypes. The purpose of such analyses is to screen genes for further wet-lab experimentation, with the goal of selecting the most promising candidates that are affected by a grouping, for example generated from a malignant transformation, a knock-out experiment or a drug treatment.

When genes are compared across two conditions, we are in the situation of two-sample testing. There is a very well-developed statistical machinery for this task, the most prominent tools include Student's t -test and its non-parametric companion, the Wilcoxon test. For microarray data, a special focus needs to be laid on the multiplicity issue, since thousands of tests are performed, rendering the non-adjusted p -values systematically too low. A very popular way out is the concept of the false discovery rate [9]. It is an estimate of the fraction of truly altered genes among a set that is declared as significantly differential.

Class Prediction

The task of class prediction focuses on the case where a learning set of n experiments monitor gene expression profiles \mathbf{x}_i of different individuals or tissue samples, and where each experiment is equipped with an additional categorical outcome variable y_i , describing its cancer (sub)type. The data can then be written as

$$\mathcal{L} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$$

with values in $\mathbb{R}^p \times \{0, 1, \dots, K - 1\}$, where y_i codes for one of the K cancer types. In such a supervised setting, the goal is to estimate the conditional probability function

$$\mathbb{P}y = k|\mathbf{x}$$

given the past experience from \mathcal{L} . Using the plug-in principle and a maximum-likelihood argument, $\mathbb{P}\cdot, \cdot$ can subsequently be used for predicting the unknown class label \hat{y} of a new instance on the basis of its gene expression profile. A precise and early diagnosis of cancer is often crucial for successful treatment. Given

the large-scale, high-throughput gene expression technology and accurate statistical methods, bio-molecular information could become as, or even more important than traditional clinical factors. The challenge is to deal with the very high dimensionality of the gene expression datasets, and to exploit them efficiently. This has been stimulating the development of statistical tools and has also been in the main focus of this thesis.

3 Thesis Overview

While optimally, the entire process from microarray design over preprocessing to normalization and data analysis would be defined specifically for each experimental setup, this is hardly ever the case in practice. Due to the complexity of all these elements, most of the scientific work focuses on one particular area. For statistical research, the preprocessed and normalized gene expression measurements are often taken at face-value. This is as well the case for the research that has been conducted during this thesis.

3.1 The Question of the Thesis

In clinical practice, cancer diagnosis and prognosis is daily routine. Enormous efforts have been made on improving it, in the hope for achieving better cure rates and fewer treatment-related side-effects. Tumor prediction is usually based on markers derived from parameters such as patient age or sex, results from screening techniques like tumor size, lymph node involvement and metastasis, the response to initial medication, or histological findings after invasive treatment. Promising results have also been achieved with microarray data, which could, given accurate statistical methods, have their breakthrough in clinical practice within the next years. Mathematically, we are in a supervised learning situation, where the task is to reconstruct the probability structure $\mathbb{P}y = k|\mathbf{x}$ as a function of the gene expression profile \mathbf{x} from past experience, i.e. from a learning set of patients with

known outcome y . Due to the very high-dimensional structure of gene expression data with thousands of predictors and only dozens of samples, this is in no way a simple task, but there is a potential of coming up with a reasonably accurate estimate when the true underlying signal is sparse. This seems to be the case, because most of the genes do not significantly contribute to class discrimination. The big open question that was pursued in this thesis is how to construct good classifiers that are reasonably predictive in such a sparse situation.

3.2 Supervised Gene Grouping

A promising approach is to use a parsimonious representation of the data, meaning that the classifier is a function of a few, highly predictive features only. Mathematically, this is represented as

$$\mathbf{P}y = k|\mathbf{x} = f(\tilde{\mathbf{x}}), \text{ where } \tilde{\mathbf{x}} = (\tilde{x}_1, \dots, \tilde{x}_q) \text{ and } q \ll p.$$

The low-dimensional predictor $\tilde{\mathbf{x}}$ is made up of transformed features $\tilde{x}_1, \dots, \tilde{x}_q$. This is inspired by that such a representation could correspond with knowledge about signaling pathways, i.e. with biologically motivated rules such as “if gene 534 is overexpressed, this causes gene 285 to be underexpressed, which is responsible for gene 739 to be overexpressed, and this finally leads to a molecular product indicative of cancer subtype 0”. Oftentimes, such biological knowledge is not available, which creates the need for statistical tools that are capable of exploratory learning. This was the main focus of the first two papers in this thesis, “Supervised Clustering of Genes” [10] and “Finding Predictive Gene Groups from Microarray Data” [11]. Both address the question of constructing sparse, but highly predictive gene signatures from gene expression data in a supervised manner. This was a novel idea to statistics, without much work being around. Hence, it required the development of new methodology and efficient search heuristics. Critical aspects include that the identified gene signatures are highly predictive, reasonably stable and easily interpretable. Being such, they are useful for class prediction,

and could as well yield insight about the biological processes in the genome. The supervised grouping approach seemed attractive enough that some people mixed in similar ideas later as well, for example Diaz [12], Jörnsten and Yu [13], as well as Bair and Tibshirani [14].

Paper 1: Supervised Clustering of Genes [10]

For tackling the quest on co-regulated genes, unsupervised methods are frequently used in microarray analysis. Most prevalent are hierarchical and k -means clustering, self-organizing maps and principal component analysis. All these methods group genes according to unsupervised similarity measures computed from the gene expressions only. If one considers a single representative value for each cluster, they lead to a parsimonious representation of the gene expression data, which could be used for class prediction. The weakness of these well-established tools is that the clustering happens without regarding the y -values, making the features of mediocre interest in classification.

Supervised clustering differs in that its primary goal is to reveal gene groups that are strongly predictive for the response y , rather than being made up of tightly co-expressed genes. We achieve this by grouping genes according to information from both the gene expressions and the response. However, the practical implementation is thorny, due to the tremendous combinatorial complexity. Even the comparably simple task of finding the most predictive group consisting of 10 genes out of 5'000 genes leaves more than $2 \cdot 10^{30}$ possibilities. Despite the powerful computers that are available, an exhaustive search would take millions of years. Thus, we have to rely on a fast clustering heuristic that incorporates gene selection and gene grouping in a single step. Paper 1 presents a grouping criterion that tries to achieve maximal class discrimination, along with a minimization algorithm that is feasible for screening thousands of genes. It also provides empirical evidence that the method provides good predictive potential, has reasonably stable output and produces features that are beyond the noise level.

Paper 2: Finding Predictive Gene Groups [11]

The second publication entitled “Finding Predictive Gene Groups from Microarray Data” further pursues the parsimonious model representation from equation 1 and presents Pelora, a novel approach for finding low-dimensional, highly predictive features. It brings supervised grouping into a sound statistical framework, using an objective function that is based on penalized likelihood. Furthermore, it provides deeper theoretical insight, more mathematical rigor and a refined algorithmical approach, while sharing the good empirical results of the first publication.

Pelora improves upon the grouping heuristic from paper 1 in many ways. First, it allows for overlapping gene signatures. This is motivated from biology, because some genes are supposed to operate in more than one signaling pathway. Pelora also yields better interaction between the gene groups, since they are built in a multivariate model environment. Its penalized objective function, besides being well familiar to statistics, is more robust, may avoid overfitting and leads to better empirical results in difficult classification problems with substantial Bayes risk. Finally, the Pelora algorithm allows for including additional clinical covariates to refine the grouping process, and it can easily be adapted to continuous response problems.

3.3 Machine Learning for Tumor Classification

An alternative route for estimating sparse structures from high-dimensional gene expression data are boosting methods. These flexible class prediction tools have been proposed by Freund and Schapire [15] in the machine learning literature and have shown remarkable success in a wide variety of applications. The initial notion of boosting was that in each of its iterations, the cases that were misclassified in the previous round get their weights increased, whereas the weights are decreased for cases that were correctly classified. More mathematically speaking, boosting methods work according to the gradient descent principle and perform gene selection as well as model fitting by iterative optimization

of an empirical risk function

$$R(\mathcal{L}, \hat{p}(\mathbf{x}), L) = \frac{1}{n} \sum_{i=1}^n L(y_i, \hat{p}(\mathbf{x}_i))$$

from a learning set \mathcal{L} via constrained functional gradient descent, where $\hat{p}(\mathbf{x}_i)$ denotes the current boosting probability estimate for the i th instance and $L(\cdot, \cdot)$ is a statistically motivated loss function. An attractive choice for the latter is the the binomial log-likelihood

$$L(y, \hat{p}(\mathbf{x})) = y \log(\hat{p}(\mathbf{x})) + (1 - y) \log(1 - \hat{p}(\mathbf{x})),$$

a continuous surrogate for the 0/1-misclassification loss. It is easy to show that the resulting LogitBoost algorithm from [16] yields an approximation to half of the log-odds ratio. Hence, logistic boosting corresponds to a linear expansion in a set of weak learners on the logit scale, obtained by stagewise optimization of the binomial log-likelihood. Obviously, the choice of the loss function and the weak learner are crucial for the boosting result.

The strength of boosting methods for class prediction with gene expression data is that yield a sparse solution by proceeding in a cautious forward way and by performing multivariate feature selection. This is advantageous compared to the most often chosen approach of preliminary univariate feature selection, because it opens the opportunity to select a powerful committee of genes and is not exposed to the danger of preferring individually strong predictors that all re-explain the same phenomenon.

Paper 3: Boosting for Tumor Classification [17]

Because of their empirical success on especially high-dimensional problems, it seemed attractive to use boosting methods for tumor classification with gene expression data. This had first been tried by Ben-Dor et al. [18] and Dudoit et al. [19] with moderate success: empirically, boosting could at best keep up with much simpler methods such as the nearest neighbor rule. Paper 3 clarifies that the performance of boosting for classification of gene

expression data can often be drastically improved by modifying the algorithm.

Without being key for the success, it is shown that boosting also profits from tailored preliminary feature selection according to the Wilcoxon test statistic. Relying on the LogitBoost procedure instead of the original AdaBoost is advantageous on noisy problems or when there are misspecifications among the class labels in the training data, both of which is frequently the case with microarray gene expressions. Finally, if discrimination has to be done for more than two tumor types, we suggest to reduce multiclass to multiple binary problems, allowing for different gene subsets and different model complexity. Empirical results provide sound evidence that these modifications make boosting a competitive player for predicting gene expression data.

Paper 4: BagBoosting for Tumor Classification [20]

The fourth publication presents a completely novel type of boosting algorithm which performs among the best methods in class prediction with microarray data. The new algorithm is called *BagBoosting* and follows a hybrid approach of ensemble methods, where a bagged weak learner is employed in the tailored boosting algorithm from paper 3.

The rationale for this combination is that the boosting committee yields low-bias high-variance predictions, whereas bagging results in high-bias low-variance output. The paper confirms that BagBoosting combines the advantages of both methods, and consistently results in lower mean squared error as well as superior empirical class prediction results. Taking care of the concerns that such sophisticated “blind” machine learning techniques hamper the biological interpretation of the prediction models, this paper also present completely novel methodology for measuring the influence of single genes on the output.

Supervised Clustering of Genes

Marcel Dettling, Peter Bühlmann
ETH Zürich

Abstract

Background: We focus on microarray data where experiments monitor gene expression in different tissues and where each experiment is equipped with an additional response variable such as a cancer type. Although the number of measured genes is in the thousands, it is assumed that only a few marker components of gene subsets determine the type of a tissue. Here we present a new method for finding such groups of genes by directly incorporating the response variables into the grouping process, yielding a supervised clustering algorithm for genes.

Results: An empirical study on eight publicly available microarray datasets shows that our algorithm identifies gene clusters with excellent predictive potential, often superior to classification with state-of-the-art methods based on single genes. Permutation tests and bootstrapping provide evidence that the output is reasonably stable and more than a noise artifact.

Conclusions: In contrast to other methods such as hierarchical clustering, our algorithm identifies several gene clusters whose expression levels clearly distinguish the different tissue types. The identification of such gene clusters is potentially useful for medical diagnostics and may at the same time reveal insights into functional genomics.

Software: Is available as an R-package under GNU public license from the webpage <http://stat.ethz.ch/~dettling/supercluster.html>

1. Introduction

Microarray technology allows the measurement of expression levels of thousands of genes simultaneously and is expected to contribute significantly to advances in fundamental questions of biology and medicine. We focus on the case where the experiments monitor the gene expression of different tissue samples, and where each experiment is equipped with an additional categorical outcome variable, describing for example a cancer type. An important problem in this setting is to study the relation between gene expression and tissue type. While microarrays monitor thousands of genes, it is assumed that only a few underlying marker components of gene subsets account for nearly all of the outcome variation, that is, determine the type of a tissue. The identification of these functional groups is crucial for tissue classification in medical diagnostics, as well as for understanding how the genome as a whole works.

As a first approach, unsupervised clustering techniques have been widely applied to find groups of co-regulated genes on microarray data. *Hierarchical clustering* [5, 21] identifies sets of correlated genes with similar behavior across the experiments, but yields thousands of clusters in a tree-like structure. This makes the identification of functional groups very difficult. In contrast, *self-organizing-maps* [22] require a prespecified number and an initial spatial structure of clusters, but this may be hard to come up with in real problems. These drawbacks were improved by a novel graph theoretical clustering algorithm [23], but as all other unsupervised techniques, it usually fails to reveal functional groups of genes that are of special interest in tissue classification. This is because genes are clustered by similarity only, without using any information about the experiment's response variables.

We focus here on supervised clustering, defined as grouping of variables (genes), controlled by information about the Y variables, that is, the tumor types of the tissues. Previous work in this field encompasses *tree harvesting* [24], a 2-step method which consists first of generating numerous candidate groups by unsupervised hierarchical clustering. Then, the average expression

profile of each cluster is considered as a potential input variable for a response model and the few gene groups that contain the most useful information for tissue discrimination are identified. Only this second step makes the clustering supervised, since the selection process relies on external information about the tissue types. An interesting supervised clustering approach that directly incorporates the response variables Y in the grouping process is the *partial least squares* (PLS) procedure [25, 26], an often applied tool in the chemometrics literature, which in a supervised manner constructs weighted linear combinations of genes that have maximal covariance with the outcome. PLS has the drawback that the fitted components involve all (usually thousands of) genes, which makes them very difficult to interpret.

Here we present a promising new method for searching functional groups, each made up of only a few genes whose consensus expression profiles provides useful information for tissue discrimination. Like PLS, it is a 1-step approach that directly incorporates the response variables Y into the grouping process and is thus an algorithm for supervised clustering of genes. Because of the combinatorial complexity when clustering thousands of genes, we rely on a greedy strategy. It optimizes an empirical objective function that quickly and efficiently measures the cluster's ability for phenotype discrimination. Inspired by [27], we choose Wilcoxon's test statistic for two unpaired samples [28], refined by a novel second criterion, the margin function. Our supervised algorithm can be started with or without initial groups of genes, and then clusters genes in a stagewise forward and backward search, as long as their differential expression in terms of our objective function can be improved. This yields clusters typically made up of 3–9 genes, whose coherent average expression levels allow perfect discrimination of tissue types. In an empirical study, the clusters show excellent out-of-sample predictive potential, and permutation and randomization techniques show that they are reasonably stable and clearly more than just a noise artifact. The output of our algorithm is thus potentially beneficial for cancer-type diagnosis. At the same time it is very accessible

for interpretation, as the output consists of a very limited number of clusters, each summarizing the information about a few genes. Thus, it may also reveal insights into biological processes and give hints on explaining how the genome works.

We first describe our new algorithm for supervised clustering of gene expression data and then apply the procedure to eight publicly available microarray datasets and test the results for their predictive potential, stability and relevance.

2. Supervised Clustering of Genes

This section presents an algorithm for supervised learning of similarities and interactions among predictor variables for classification in very high dimensional spaces, and hence is predestinated for searching functional groups of genes on microarray expression data.

2.1 The Partitioning Problem

Our basic stochastic model for microarray data equipped with categorical response is given by a random pair

$$(\mathbf{X}, Y) \text{ with values in } \mathbb{R}^p \times \mathbb{Y},$$

where $\mathbf{X} \in \mathbb{R}^p$ denotes a log-transformed gene expression profile of a tissue sample, standardized to mean zero and unit variance. Y is the associated response variable, taking numeric values in $\mathbb{Y} = \{0, 1, \dots, K-1\}$. A usual interpretation is that Y codes for one of K cancer types. For simplicity and a concise description of the algorithm, we first assume that $K = 2$, so that the response is binary. A generalization of the setting for multi-categorical response ($K > 2$) is given below in section 2.3.

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 and thus the type of a tissue, we model the conditional probability as

$$\mathbf{P}[Y = 1 | \mathbf{X}] = f(\mathbf{X}_{\mathcal{G}_1}, \mathbf{X}_{\mathcal{G}_2}, \dots, \mathbf{X}_{\mathcal{G}_q}), \quad (1)$$

where $f(\cdot)$ is a nonlinear function mapping from \mathbb{R}^q to $[0, 1]$, $\{\mathcal{G}_1, \dots, \mathcal{G}_q\}$ with $q \ll p$ are functional groups or clusters of genes which form a disjoint and usually incomplete partition of the index set: $\{\cup_{i=1}^q \mathcal{G}_i\} \subset \{1, \dots, p\}$ and $\mathcal{G}_i \cap \mathcal{G}_j = \emptyset$, $i \neq j$. Finally, $\mathbf{X}_{\mathcal{G}_i} \in \mathbb{R}$ denotes a ‘‘representative’’ expression value of gene cluster \mathcal{G}_i . There are many possibilities to determine such group values $\mathbf{X}_{\mathcal{G}_i}$, but since we would like to shape clusters that contain similar genes, a simple linear combination is an accurate choice (see [24, 29]):

$$\mathbf{X}_{\mathcal{G}_i} = \frac{1}{|\mathcal{G}_i|} \sum_{g \in \mathcal{G}_i} \alpha_g \mathbf{X}_g \text{ with } \alpha_g \in \{-1, 1\}. \quad (2)$$

Because of the use of log-transformed, mean centered and standardized expression data, we, as a novel extension allow the contribution of a particular gene g to the group value $\mathbf{X}_{\mathcal{G}_i}$ also to be given by its ‘‘sign-flipped’’ expression value $-\mathbf{X}_g$. This means that we treat under- and overexpression symmetrically and it prevents the differential expression of genes with different polarity (that is, one with low expression for class 0 and the other with low expression for class 1) from canceling out when they are averaged. But even by using such simple cluster expression values as in (2), finding a partition of the index set $\{1, \dots, p\}$ into subsets or clusters $\{\mathcal{G}_1, \dots, \mathcal{G}_q\}$ that virtually determine the probability structure is still highly nontrivial and the design of a procedure which reveals the exact partition according to (1) is too ambitious. Thus, we develop a computationally intensive procedure that approximately solves (1) and empirically yields good results.

2.2 Clustering with Scores and Margins

A practical heuristic for gene clustering is the *cluster affinity search technique* (CAST) [23]. Our approach is algorithmically similar and also relies on growing the cluster incrementally by adding one gene after the other. Subsequent cleaning steps help us to remove spurious genes that were incorrectly added to the cluster at earlier stages. As in CAST, we repeat growth and

removal until the cluster stabilizes, and then start a new cluster. The main, and very important difference is that we do not augment (or shorten) the cluster by the gene that suits best (or least) into the current cluster in terms of an unsupervised similarity measure, but base our strategy for supervised clustering of genes on adding (or removing) the gene that improves the differential expression of the current cluster most, according to an empirical objective function for the representative group values from (2). To be more explicit, we assume now that we are given n independent and identically distributed realizations

$$(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n), \text{ with } \mathbf{x}_j \in \mathbb{R}^p \text{ and } y_j \in \{0, 1\}, \quad (3)$$

of the random vector (\mathbf{X}, Y) , whose expression profiles \mathbf{x}_j are centered to mean zero and scaled to unit variance. The objective function needs to be a quantitative and efficiently computable measure of a cluster's ability to discriminate the tissues. Since we aim for subsets of genes with accurate separation in binary problems, we rely on Wilcoxon's test statistic for two unpaired samples [28], which has been also applied as a nonparametric rank-based score function for genes in [27]. The score of a single gene i is computed from its n -dimensional vector of observed values $\xi_i = (x_{i1}, \dots, x_{in})$,

$$\text{Score}(\xi_i) = s(\xi_i) = \sum_{j \in \mathcal{N}_0} \sum_{l \in \mathcal{N}_1} 1_{[x_{ij} \geq x_{il}]}, \quad (4)$$

where x_{ij} is the expression value of gene i for tissue j and \mathcal{N}_k represents the set of the n_k tissues $\in \{1, \dots, n\}$ being of type $k \in \{0, 1\}$. The score uses information about the type of the tissues and is thus a criterion for supervised clustering. It can be interpreted as counting, for each experiment having response value 0, the number of tissues from class 1 that have smaller expression values, and summing up these quantities. Computing the score for a gene cluster \mathcal{G}_i goes likewise via its observed representative values $\xi_{\mathcal{G}_i} = (x_{\mathcal{G}_i,1}, \dots, x_{\mathcal{G}_i,n})$. Viewing the score as Wilcoxon's test statistic, it allows to order genes and clusters according to their potential significance for tissue discrimination.

If the expression values of a particular gene or cluster yield exact separation of the classes, the expression values for all tissue samples having response 0 are uniformly lower than the ones belonging to class 1, or vice versa. In the former case, the score function returns its minimal value $s_{min} = 0$, in the latter case the maximum score $s_{max} = n_0 n_1$ is assigned.

We rely on the use of log-transformed, mean centered and standardized gene expression data and thus need to prevent the averaging of two discriminatory genes with different polarity (that is, one with low expression for class 0 and the other with low expression for class 1) canceling out the differential expression of their mean. Therefore, we aim for low expression values pointing to class 0 for all genes, which is achieved by using the sign-flipped expression $\tilde{\xi}_i$ for all genes $i \in \{1, \dots, p\}$,

$$\tilde{\xi}_i = \alpha_i \xi_i = \begin{cases} (x_{i1}, \dots, x_{in}), & \text{if } s(\xi_i) \leq s_{max}/2, \\ (-x_{i1}, \dots, -x_{in}), & \text{if } s(\xi_i) > s_{max}/2. \end{cases} \quad (5)$$

The sign-flip is equivalent to setting $\alpha_g = -1$ in equation (2) for all genes that tend to have lower expression values for the tissues of type 1 than for tissues of type 0. After the sign-flip, the scores of all individual genes i in the expression matrix are equal to

$$s(\tilde{\xi}_i) = \min(s(\xi_i), s_{max} - s(\xi_i)),$$

and since all genes now have the same polarity, we can safely average them to compute group expression values. It is important to notice that the biological interpretation is not impeded by the sign-flips. Nevertheless for interpretational purposes, the information about them should be recorded.

During the clustering process, we typically come across different gene or cluster expression vectors that have equal score (often zero) and hence the same quality according to our objective function. This is due to the discrete range of the score function. To achieve uniqueness in the decisions which gene or cluster is optimal, we need a refinement of our objective function. We thus introduce the margin function, a continuous and real-valued measure for the strength of tissue discrimination of a sign-flipped gene

expression vector $\tilde{\xi}_i$, where low expression values point towards the tissues of class 0,

$$\text{Margin}(\xi_i) = m(\xi_i) = \min_{l \in \mathcal{N}_1} (x_{il}) - \max_{j \in \mathcal{N}_0} (x_{ij}), \quad (6)$$

where $\mathcal{N}_0, \mathcal{N}_1$ and x_{ij} are as in (22). The margin function is positive if and only if the score is zero and $\tilde{\xi}_i$ then perfectly separate the tissues, otherwise it is negative. It measures the size of the gap between the lowest expression value from tissues with response 1, and the highest gene expression corresponding to class 0. The larger this gap and hence the value of the margin function, the easier and clearer the discrimination of the two classes. The computation of the margin is again likewise for clusters via $\xi_{\mathcal{G}_i}$. Whenever various gene or cluster expression profiles have equal score, their quality is judged by the margin function. Our objective function thus has two components. The score function is regarded as highest priority, whereas the margin function serves as the next highest priority criterion to achieve uniqueness.

Our clustering algorithm is detailed in Figure 5. It begins with the sign-flip operation described in (5) to bring all genes to the same polarity. The clustering process can be started with or without initial gene clusters. If none are given, we start the procedure with the single gene that optimizes the objective function. Otherwise, the representative value of the starting cluster is determined. We then proceed by constructing the cluster incrementally. By searching among all genes, we merge and average the current cluster with one single gene, such that the augmented cluster optimizes our objective function, that is, has the lowest score, or (in case of ‘‘ties’’) the largest margin. The merging process is repeated until the objective function can no longer be improved. To remove spurious elements out of the current cluster, we then continue with a backward pruning stage, where genes are excluded step by step so that the objective function is optimized by every single removal. This cleaning stage aims to root out genes that were wrongly added to the cluster before.

Supervised Clustering Algorithm

1. Start with the entire $p \times n$ expression matrix X . Its rows are genes, and its columns are observations of 2 different tissue types, having zero mean and unit variance.
2. Determine the score of every gene i , that is, every n -dimensional row of observed expression values $\xi_i = (x_{i1}, \dots, x_{in})$ in X as in (22). Flip the sign of each gene expression vector ξ_i that has score $s(\xi_i) > s_{max}/2$ by multiplying it with (-1) ,

$$\tilde{\xi}_i = \alpha_i \xi_i = \begin{cases} \xi_i, & \text{if } s(\xi_i) \leq s_{max}/2, \\ -\xi_i, & \text{if } s(\xi_i) > s_{max}/2. \end{cases}$$

This operation changes the score to $s(\tilde{\xi}_i) = \min(s(\xi_i), s_{max} - s(\xi_i))$.

3. Composition of the starting values
 - a) If no initial cluster \mathcal{G} is given, identify the gene i^* having the lowest score $s(\tilde{\xi}_i)$. If more than one is found, the gene i^* with the largest margin $m(\tilde{\xi}_i)$ as in (6) is chosen. Set the initial cluster mean $\xi_{\mathcal{G}}$ equal to the expression vector $\tilde{\xi}_{i^*}$ of the chosen gene.
 - b) If an initial cluster \mathcal{G} is given, average the expression of the genes therein,

$$\xi_{\mathcal{G}} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \tilde{\xi}_g = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \alpha_g \cdot (x_{g1}, \dots, x_{gn})$$

4. Forward Search

Average the current cluster expression profile $\xi_{\mathcal{G}}$ with each individual gene i ,

$$\xi_{\mathcal{G}+i} = \frac{1}{|\mathcal{G}|+1} \left(\tilde{\xi}_i + \sum_{g \in \mathcal{G}} \tilde{\xi}_g \right), \quad i = 1, \dots, p.$$

4. ... Identify the winning gene i^* as $\arg \min_i s(\xi_{\mathcal{G}+i})$, that is, the gene that leads to the lowest score. If not unique, identify the winning gene i^* as the one that optimizes score *and* margin, that is, $i^* = \arg \min_i s(\xi_{\mathcal{G}+i})$ as well as $i^* = \arg \max_i m(\xi_{\mathcal{G}+i})$.
5. Repeat step 4 until the identified gene i^* is no longer accepted to enter the cluster. This is said to happen, if the score of the updated cluster expression vector $\xi_{\mathcal{G}+i^*}$ worsens, i.e. $s(\xi_{\mathcal{G}+i^*}) > s(\xi_{\mathcal{G}})$, or if the score remains unchanged and the margin deteriorates, i.e. $s(\xi_{\mathcal{G}+i^*}) = s(\xi_{\mathcal{G}})$ as well as $m(\xi_{\mathcal{G}+i^*}) < m(\xi_{\mathcal{G}})$
6. Backward Search
Exclude each gene i of the current cluster \mathcal{G} separately and average the expression vectors of the remaining genes,

$$\xi_{\mathcal{G}-i} = \frac{1}{|\mathcal{G}|-1} \left(\sum_{g \in \mathcal{G} \setminus \{i\}} \tilde{\xi}_g \right), \quad i \in \mathcal{G}.$$

Compute score and margin of each $\xi_{\mathcal{G}-i}$. Identify (as in step 4) that gene i^* whose exclusion optimizes the score, or if not unique, optimizes score and margin.

7. Repeat step 6 until the exclusion of the identified gene i^* is (according to the formulation in step 5) no longer accepted.
8. Repeat steps 4 – 7 until the cluster converges and the objective function is optimal.
9. If more than one cluster \mathcal{G} is desired, discard the genes in the former clusters from X and restart the algorithm at step 3 with the reduced, sign-flipped expression matrix.

Figure 5: Algorithm for supervised clustering of binary problems with scores and margins.

Accordingly, the forward and backward stages are repeated until the cluster converges, i.e. no further improvement of the objective function by adding or removing single genes is possible.

If one wishes to have more than $q = 1$ cluster for a binary class distinction, the genes forming the first cluster are discarded from the expression matrix, and the clustering process is restarted, again with or without an initial cluster. The algorithm's computations are feasible for dimensions p and sample sizes n which are clearly beyond today's common orders and hence also applicable for microarray experiments from the future. The computing time for searching $q = 5$ clusters in the binary 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 5 seconds only. Software for the supervised clustering algorithm is available for free as an R-Package at <http://stat.ethz.ch/~dettling/supercluster.html>.

In summary, our cluster algorithm is a combination of variable (gene) selection for cluster membership and forming a new predictor by possible sign-flipping and averaging the gene expressions within a cluster as in (2). The cluster membership is determined with a forward and backward searching technique that optimizes the predictive score and margin criteria in (22) and (6), which both involve the supervised response variables from the data.

2.3 Generalization for Multiclass Problems

Here we explain the extension of the supervised clustering algorithm to multi-categorical ($K > 2$) problems, where the response comprises more than two tissue types. We recommend comparing each response class separately against all other classes. This *one-against-all* approach for reduction to K binary problems is very popular in the machine learning community, since many algorithms are solely designed for binary response. It works by defining

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

and running K times the supervised clustering algorithm on $(x_1, y_1^{(k)}), \dots, (x_n, y_n^{(k)})$ as explained above. The interpretation is that we, as in (1), model the conditional probability for discrimination of the k th class versus all the other response categories as depending on a few gene subsets only,

$$\mathbb{P}[Y^{(k)} = 1 | \mathbf{X}] = f_k \left(\mathbf{X}_{\mathcal{G}_1^k}, \mathbf{X}_{\mathcal{G}_2^k}, \dots, \mathbf{X}_{\mathcal{G}_q^k} \right)$$

for $k = 0, \dots, K-1$, where $f_k(\cdot)$ are nonlinear functions mapping from \mathbb{R}^q to $[0, 1]$. $\mathcal{G}_1^k, \dots, \mathcal{G}_q^k$ are the $q \ll p$ functional groups of genes and $\mathbf{X}_{\mathcal{G}_1^k}, \dots, \mathbf{X}_{\mathcal{G}_q^k}$ are their representative group values, defined as in (2). When the supervised clustering algorithm is applied to each of the K binary class distinctions, this results in totally $K \cdot q$ clusters, which can then be used to model the conditional probability for the K -class response,

$$\mathbb{P}[Y = k | \mathbf{X}] = f \left(\mathbf{X}_{\mathcal{G}_1^0}, \dots, \mathbf{X}_{\mathcal{G}_q^0}, \dots, \mathbf{X}_{\mathcal{G}_1^{K-1}}, \dots, \mathbf{X}_{\mathcal{G}_q^{K-1}} \right).$$

It is important to notice that instead of considering each class against all the other classes, many more ways to reduce a multi-class problem to multiple binary problems exist, see [30, 31] for a thorough discussion. We assume that problem dependent solutions which utilize deeper knowledge about the biological relation among the tissue types could be even more accurate for reducing multi-categorical to binary problems.

3. Numerical Results

3.1. Data

Leukemia dataset

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. For more information see [32]; the data are available at <http://www.genome.wi>.

mit.edu/MPR. Following exactly the protocol in [19], we preprocess them by thresholding, filtering, a logarithmic transformation and standardization, so that the data finally comprise the expression values of $p = 3,571$ genes.

Breast cancer dataset

This dataset, described in [33], monitors $p = 7,129$ genes in 49 breast tumor samples. The data were obtained by applying the Affymetrix technology and are available at http://mgm.duke.edu/genome/dna_micro/work/. We thresholded the raw data with a floor of 100 and a ceiling of 16,000 before applying a base 10 logarithmic transformation. Finally, each experiment was standardized to zero mean and unit variance. The response variable describes the status of the estrogen receptor (ER). According to [33], two samples failed to hybridize correctly and were excluded from their analysis. In five cases, two different clinical tests for determination of the ER status yielded conflicting results. These five plus another four randomly chosen samples were also separated from the rest of the data, so that a dataset of $n = 38$ samples remained, of which 18 were ER+ and 20 ER-.

Colon cancer dataset

In this dataset, expression levels of 40 tumor and 22 normal colon tissues for 6,500 human genes are measured using the Affymetrix technology. A selection of 2,000 genes with highest minimal intensity across the samples has been made in [34]. The data are available at <http://microarray.princeton.edu/oncology/>. As for all other datasets, we process these data further by carrying out a base 10 logarithmic transformation and standardizing each tissue sample to zero mean and unit variance across the genes.

Prostate cancer dataset

The raw data are available at <http://www-genome.wi.mit.edu/MPR/prostate> and comprise the expression of 52 prostate tumors and 50 non-tumor prostate samples, obtained from the Affymetrix technology. We use normalized and thresholded data as described in [35]. We also excluded genes whose expression varied less than 5-fold relatively, or less than 500 units absolutely

between the samples, leaving us with the expression of $p=6,033$ genes. Finally, we applied a base 10 logarithmic transformation and standardized each experiment to zero mean and unit variance across the genes.

SRBCT dataset

This dataset was described in [36] and contains gene expression profiles for classifying small round blue cell tumors of childhood into 4 classes (Neuroblastoma, Rhabdomyosarcoma, Non-Hodgkin-Lymphoma, Ewing Family of Tumors) and was obtained from cDNA microarrays. A training set comprising 63 SRBCT tissues, as well as a test set consisting of 20 SRBCT and 5 non-SRBCT samples are available at <http://www.nhgri.nih.gov/DIR/Microarray/Supplement>. Each tissue sample is associated with thoroughly preprocessed expression profile of $p=2,308$ genes, already standardized to zero mean and unit variance across genes.

Lymphoma dataset

This dataset is available at <http://11mpp.nih.gov/lymphoma/data/figure1> and contains gene expression levels of the $K=3$ most prevalent adult lymphoid malignancies: 42 samples of diffuse large B-cell lymphoma (DLBCL, class 0), 9 observations of follicular lymphoma (FL, class 1), and 11 cases of chronic lymphocytic leukemia (CLL, class 2). The total sample size is $n=62$, and the expression of $p=4,026$ well-measured genes, preferentially expressed in lymphoid cells or with known immunological or oncological importance is documented. More information on these data can be found in [37]. We imputed missing values and standardized the data as described in [19].

Brain tumor dataset

This dataset, presented in [38], contains $n=42$ microarray gene expression profiles from $K=5$ different tumors of the central nervous system, i.e. 10 medulloblastomas, 10 malignant gliomas, 10 AT/RTs, 8 PNETs and 4 human cerebella. The raw data originated from the Affymetrix technology and are publicly available at <http://www.genome.wi.mit.edu/MPR/CNS>. For data prepro-

cessing, we followed the protocol in the supplementary information to [38]. After thresholding, filtering, a logarithmic transformation and standardization of each experiment to zero mean and unit variance, a dataset comprising $p=5,597$ genes remained.

NCI dataset

This dataset comprises gene expression levels of $p=5,244$ genes for $n=61$ human tumor cell lines which can be divided in $K=8$ classes: 7 breast, 5 central nervous system, 7 colon, 6 leukemia, 8 melanoma, 9 non small cell lung carcinoma, 6 ovarian and 9 renal tumors. A more detailed description of the data can be found on the website <http://genome-www.stanford.edu/nci60> and in [39]. We work with preprocessed data as in [19].

3.2 Results with Supervised Clustering

In this section we briefly describe the results obtained by applying the supervised clustering algorithm to the above datasets. Generally, the output of the clustering procedure is very promising. In all eight datasets we analyzed, comprising a total of 24 binary class distinctions, the average cluster expression \mathbf{x}_G always perfectly discriminates the two response classes (in multiclass problems, this is one class against the rest). Hence, the scores of all clusters are equal to zero. Moreover, the clusters have strongly positive margins, indicating that the different tissue types are clearly separated. As an example, Figure 6 shows impressively how well the average cluster expression vectors x_{G_1} and x_{G_2} discriminate between the three response classes of the lymphoma dataset. It is intuitively clear from Figure 6 that our cluster expression vectors \mathbf{x}_G are very suitable as predictor variables for the tissue types and they indeed allow for error-free classification on the training data and also yield good results on independent test datasets, see section 3.4.

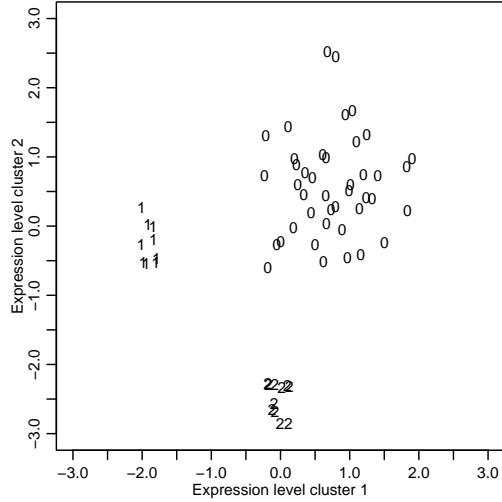


Figure 6: *Lymphoma data.* Average cluster expression \mathbf{x}_{G_1} shaped for the separation of response class 1 (FL), versus response classes 0 & 2 (DLBCL & CLL) on *x*-axis, and \mathbf{x}_{G_2} formed for discrimination of class 2 versus classes 0 & 1 on *y*-axis.

3.3 Permutation Test

This section is concerned with assessing relevance and addresses the question whether or not the promising output of the clustering procedure is a noise artifact. For this purpose, we explore quality measures of clusters generated from random noise gene expression data and compare them to the results obtained with the original data. Since the distributions of the score function $s(\cdot)$ and the margin function $m(\cdot)$ on noise are not known, we rely on simulations. Let (y_1, \dots, y_n) be the original set of re-

sponses. Then,

$$(y_1^{*(\ell)}, \dots, y_n^{*(\ell)})$$

is a “shuffled” set of responses, constructed from the original response set by a random permutation for each $\ell = 1, \dots, L$. We then allocate an element of the permuted response to each of the (fixed) gene expression profiles \mathbf{x}_i , giving us independent and identically distributed pairs

$$(\mathbf{x}_1, y_1^{*(\ell)}), (\mathbf{x}_2, y_2^{*(\ell)}), \dots, (\mathbf{x}_n, y_n^{*(\ell)}) \text{ for each } \ell = 1, \dots, L$$

as in (3). The supervised clustering procedure is then applied $L = 1000$ times on such data with randomly permuted responses. For every permuted set of responses, a single cluster ($q = 1$) was formed on the entire dataset and both its final score $s^{*(\ell)}$ and margin $m^{*(\ell)}$ were recorded.

We explored the empirical distribution of the scores and margins from permuted data to judge whether the clusters found on the original datasets are of better quality than we would expect by chance. The results given in Figure 7 and in Tables 1 and 2 for a representative selection of data (see the caption to Table 1 for details on data selection) are very satisfactory. As outlined in section 3.2, the scores $s^{(0)}$ on the original datasets altogether are equal to zero, with clearly positive margins $m^{(0)}$. The parameters on the randomly permuted data are worse: The final score $s^{*(\ell)}$ reached the minimal value of zero in 11% to 98% of the shuffling trials in different datasets (for example, 41% in Figure 7). These frequencies represent a non-significant result in our permutation test for the score function. However, this is not very troubling, since the final margins $m^{*(\ell)}$ for the permuted data were at best slightly positive, not indicating a clear separation of the randomly shuffled response classes. Values in the range of the margin in the original data were never achieved with any of the permuted data. This corresponds to a p -value of zero in the permutation test for our entire objective function consisting of score *and* margin. We thus can “for surely” reject the hypothesis that the clusters found

<i>Margins</i>	$m^{(0)}$	$\max(m^{*(\ell)})$	$\text{med}(m^{*(\ell)})$	$\min(m^{*(\ell)})$
Leukemia	0.20	0.05	-0.01	-2.41
Breast	1.29	0.23	0.04	-0.82
Prostate	0.05	0.02	-0.04	-0.90
Colon	0.08	0.05	-0.12	-1.39
SRBCT	1.00	0.11	-0.06	-1.16
Lymphoma	1.65	0.14	0.01	-1.16
Brain	1.03	0.32	0.09	-0.29
NCI	2.52	0.44	0.12	-0.91

Table 1: Margins $m^{(0)}$ from the original datasets, as well as maximal, median and minimal margins $m^{*(\ell)}$ from 1000 permuted replicates, for leukemia data (AML/ALL distinction), breast cancer data (ER+/ER- distinction), prostate data (tumor/normal-distinction), colon data (tumor/normal-distinction), SRBCT data (distinction of the Ewing family of tumors versus three other tumor types), lymphoma data (distinction of DLBCL versus FL and CLL), brain tumor data (separation of atypical teratoid/rhabdoid tumors (AT/RTs) against 4 other tumor types) and NCI data (distinction of leukemia against seven other cancers).

on the original data by our supervised algorithm are irrelevant and just a noise artifact. Moreover, we observed that the clusters from permuted data were much larger in size, exceeding the typical size (see also section 3.5, Table 11) on non-permuted data of between 3 to 9 genes clearly, e.g. with a mean of 12.5 and a standard deviation of 3.2 genes for the AML/ALL-distinction on the leukemia dataset.

The fact that the score has highly non-significant p-values is at first sight surprising. The reason for this is that the cluster expression values $x_{\mathcal{G},j}$ in (2) are highly dependent among the samples $j = 1, \dots, n$ via the responses y_j in the supervisedly estimated cluster $\mathcal{G} = \mathcal{G}(y_1, \dots, y_n)$ and the sign coefficients $\alpha_g = \alpha_g(y_1, \dots, y_n)$. This strong interdependence causes the

<i>Scores</i>	$s^{(0)}$	$\min(s^{*(\ell)})$	$\max(s^{*(\ell)})$	$\#(s^{*(\ell)} = 0)/L$
Leukemia	0	0	279	0.41
Breast	0	0	43	0.91
Prostate	0	0	566	0.17
Colon	0	0	164	0.11
SRBCT	0	0	148	0.26
Lymphoma	0	0	78	0.67
Brain	0	0	11	0.98
NCI	0	0	13	0.95

Table 2: Scores $s^{(0)}$ from the original dataset, maximal and minimal scores $s^{*(\ell)}$ from $L = 1000$ permuted replicates, and proportion of shuffled bootstrap trials where score 0 was achieved. The selection of data was as in table 1.

unusual phenomenon that the null-distribution, assuming no association between the expression values X and the response Y , has a substantial probability to score zero. The margin statistics in (6) has much better power properties than the score.

3.4 Predictive Potential

In this section, we will evaluate the predictive potential of the supervised clustering algorithm’s output to see if it could successfully reveal functional groups of genes. A predictor or classifier for K different tissue types is a function $C(\cdot)$ that assigns a class label \hat{y} , based on an observed feature vector \mathbf{x} . More precisely, the classification rule here will be based on average cluster expression values $\mathbf{x} = (x_{\mathcal{G}_1^0}, \dots, x_{\mathcal{G}_q^{K-1}})$ as $K \cdot q$ features,

$$\hat{y} = C(\mathbf{x}) = C\left(\mathbf{x}_{\mathcal{G}_1^0}, \dots, \mathbf{x}_{\mathcal{G}_q^0}, \dots, \mathbf{x}_{\mathcal{G}_1^{K-1}}, \dots, \mathbf{x}_{\mathcal{G}_q^{K-1}}\right),$$

with values in $\{0, \dots, K - 1\}$. In practice, the classifier is built from a learning set of tissues whose class labels are known. Subsequently it can be used to predict the class labels of new tissues

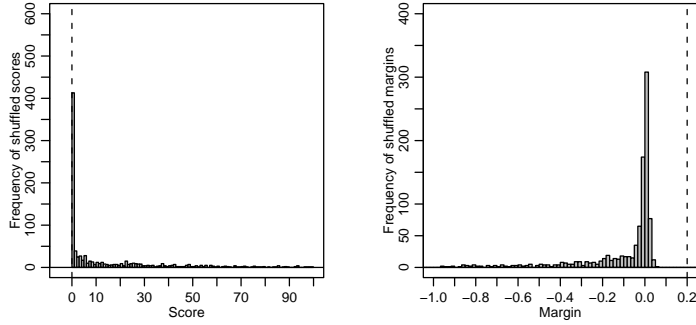


Figure 7: Histograms, showing the empirical distribution of scores (left) and margins (right) for the leukemia dataset (AML/ALL distinction), based on 1000 bootstrap replicates with permuted response variables. The dashed vertical lines mark the values of score and margin with the original response variables.

with unknown outcome. There are various methods to build classification rules based on past experience and we restrict here on two relatively simple methods which are well suited for our purpose.

Nearest Neighbor Classification

An easy to implement and, compared to more sophisticated methods, impressively competitive classifier for microarray data is the k -nearest neighbor rule [40]. It is based on a distance function $d(\cdot, \cdot)$ for pairs \mathbf{x} and \mathbf{x}' of feature vectors. Since we consider here standardized gene expression data, the Euclidean distance function

$$d(\mathbf{x}, \mathbf{x}') = \sqrt{\sum_{i=1}^{K \cdot q} (x_i - x'_i)^2}$$

is a reasonable choice. Then, for each new feature vector, the k closest feature vectors from the tissues in the learning data are identified and the predicted class is given by majority vote of the associated responses of these k closest neighbors. We found a choice of $k = 1$ neighbors to be appropriate, but more data driven approaches via cross validation for the determination of k would be possible.

Aggregated Trees

Another approach which proved to be very fruitful in our setting is as follows: When knowing conditional probabilities $p_k(\mathbf{x}) = \mathbf{P}[Y^{(k)} = 1 | \mathbf{X} = \mathbf{x}]$, which specify how likely a tissue with feature vector \mathbf{x} belongs to the k th or one of the other classes, the classifier function is

$$\hat{y} = C(\mathbf{x}) = \arg \max_{k \in \{0, \dots, K-1\}} p_k(\mathbf{x}_{G_1^k}, \dots, \mathbf{x}_{G_q^k}), \quad (7)$$

meaning that a tissue is assigned to the class with highest probability. In practice, of course, we have to rely on estimated probabilities $\hat{p}_k(\mathbf{x})$. An often applied method for this task is the CART algorithm for fitting classification trees [41]. Its drawback when using it with our supervised clusters as input is that in case of perfect separation of the tissues in the training data, it only uses one (the first) component $x_{G_1^k}$ of the feature vector \mathbf{x} to determine conditional probabilities $\hat{p}_k(\mathbf{x})$, and does not take into account any of the useful information about the remaining $(q-1)$ input variables $\mathbf{x}_{G_2^k}, \dots, \mathbf{x}_{G_q^k}$. To improve the tree-based probability estimates, we design a novel technique based on plurality voting with classification trees, called *aggregated trees*. The idea is to fit q trees, one each with the q cluster expression profiles (components of the feature vector \mathbf{x}) that have been found by our supervised algorithm for a particular binary class distinction. Each tree casts a weighted vote $\hat{p}_{ki}(x_{G_i^k}), i = 1, \dots, q$, for

response class k against the rest. Averaging then yields

$$\hat{p}_k(\mathbf{x}) = \hat{p}_k(\mathbf{x}_{G_1^k}, \dots, \mathbf{x}_{G_q^k}) = \frac{1}{q} \cdot \sum_{i=1}^q \hat{p}_{ki}(\mathbf{x}_{G_i^k}).$$

as estimated conditional probabilities, which can be plugged into (7) for maximum likelihood classification.

Empirical Study

Because, except for the leukemia and SRBCT data, no genuine test sets are available, our empirical study for exploring the classification potential is based on random divisions into learning and test set as well as leave-one-out cross validation. For the latter, we set aside the i th tissue and carry out cluster identification and classifier fitting by considering only the remaining $(n - 1)$ data points. We then honestly predict \hat{y}_i , the class label of the i th tissue sample and repeat this process for all data we have. Each observation is held out and predicted exactly once. We can determine the test set error by calculating the fraction of predicted class labels which differ from the true class labels. Results for the nearest neighbor and the aggregated tree classifier and varying number of clusters q are given in Tables 3 and 4.

It is known from theory (see, for example [42], p.71) that error rates from leave-one-out cross validation have low bias but large variance. Estimating error rates by repeated random splitting of the data into training and (larger) test sets may be better in terms of mean squared error. In Tables 5 and 6 we report misclassification rates which are based on $N = 100$ random divisions into a learning set comprising two thirds, and a test set containing the remaining third of all n data. We took care that the class proportions were roughly identical in learning and test set. Also here in every run here, both cluster identification and classifier construction are carried out on the training data, followed by honestly predicting the class labels \hat{y}_i for the test data with the two classifiers and various number of clusters q . The misclassification rate is then calculated as the averaged fraction

<i>Leukemia</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	5.56%	5.56%	4.17%	2.78%
Aggregated Trees	5.56%	5.56%	1.39%	1.39%
<i>Breast</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	0.00%	0.00%	0.00%	0.00%
Aggregated Trees	0.00%	0.00%	0.00%	0.00%
<i>Prostate</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	13.73%	7.84%	4.90%	6.86%
Aggregated Trees	13.73%	13.73%	6.86%	8.82%
<i>Colon</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	27.42%	22.58%	22.58%	19.35%
Aggregated Trees	27.42%	29.03%	19.35%	19.35%
<i>SRBCT</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	0.00%	0.00%	0.00%	0.00%
Aggregated Trees	3.17%	0.00%	0.00%	0.00%
<i>Lymphoma</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	3.23%	1.61%	1.61%	1.61%
Aggregated Trees	3.23%	1.61%	1.61%	1.61%
<i>Brain</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	30.95%	23.81%	19.05%	16.67%
Aggregated Trees	42.86%	23.81%	21.43%	19.05%
<i>NCI</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	40.98%	40.98%	36.07%	29.51%
Aggregated Trees	49.18%	47.54%	39.34%	29.51%

Table 3: Misclassification rates for out-of-sample classification with $q \in \{1, 2, 3, 5\}$ gene clusters as features, based on leave one out cross validation.

of predicted class labels which differ from the true one.

We observe that the error estimates obtained from random splitting are on a slightly higher level than the ones from leave-one-out cross validation. We also see that introducing some redundancy for the discrimination process by using additional

	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	2.78%	2.78%	2.78%
Aggregated Trees	2.78%	2.78%	2.78%
<i>Breast</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	0.00%	0.00%	0.00%
Aggregated Trees	0.00%	0.00%	0.00%
<i>Prostate</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	4.90%	4.90%	5.88%
Aggregated Trees	6.86%	5.88%	5.88%
<i>Colon</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	16.13%	17.74%	19.35%
Aggregated Trees	16.13%	17.74%	17.74%
<i>SRBCT</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	0.00%	0.00%	1.59%
Aggregated Trees	1.59%	1.59%	1.59%
<i>Lymphoma</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	0.00%	0.00%	0.00%
Aggregated Trees	0.00%	0.00%	0.00%
<i>Brain</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	19.05%	16.67%	16.67%
Aggregated Trees	14.29%	11.90%	11.90%
<i>NCI</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	24.59%	27.87%	26.23%
Aggregated Trees	21.31%	21.31%	19.67%

Table 4: Misclassification rates for out-of-sample classification with $q \in \{10, 15, 20\}$ gene clusters as features, based on leave one out cross validation.

clusters, that is, increasing q , yields better performance; but of course, a too large value of q would exhibit overfitting.

	$q = 1$	$q = 2$	$q = 3$	$q = 5$
<i>Leukemia</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	6.58%	4.62%	4.21%	3.75%
Aggregated Trees	6.58%	6.12%	3.71%	3.54%
<i>Breast</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	1.00%	0.75%	0.75%	1.00%
Aggregated Trees	1.00%	1.58%	1.67%	2.33%
<i>Prostate</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	14.47%	11.68%	9.62%	7.97%
Aggregated Trees	14.47%	16.47%	10.32%	8.79%
<i>Colon</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	23.35%	20.35%	19.10%	16.95%
Aggregated Trees	23.35%	21.80%	19.70%	18.10%
<i>SRBCT</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	1.33%	0.48%	0.43%	0.48%
Aggregated Trees	5.76%	0.95%	0.71%	1.10%
<i>Lymphoma</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	2.15%	2.20%	1.50%	0.85%
Aggregated Trees	3.45%	2.45%	1.40%	0.80%
<i>Brain</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	31.21%	27.50%	26.36%	24.71%
Aggregated Trees	35.43%	28.43%	24.43%	22.14%
<i>NCI</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	45.25%	40.25%	37.90%	34.80%
Aggregated Trees	51.85%	42.35%	38.05%	34.05%

Table 5: Misclassification rates for out-of-sample classification with $q \in \{1, 2, 3, 5\}$ gene clusters as features, based on $N = 100$ random divisions into learning set (two thirds of the data) and test set (one third of the data).

Comparison to Classification with Single Genes

Does the use of averaged cluster expression profiles from our supervised algorithm improve the classification results compared to

	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	3.33%	3.38%	3.25%
Aggregated Trees	2.79%	2.71%	2.62%
<i>Breast</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	0.83%	1.00%	1.00%
Aggregated Trees	2.58%	2.42%	3.00%
<i>Prostate</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	7.26%	6.94%	6.91%
Aggregated Trees	8.12%	8.00%	7.79%
<i>Colon</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	16.45%	16.05%	15.95%
Aggregated Trees	16.95%	16.20%	16.45%
<i>SRBCT</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	0.76%	0.95%	1.05%
Aggregated Trees	1.76%	1.90%	2.14%
<i>Lymphoma</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	0.65%	0.50%	0.50%
Aggregated Trees	0.25%	0.20%	0.30%
<i>Brain</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	23.86%	23.71%	23.36%
Aggregated Trees	19.64%	18.29%	16.86%
<i>NCI</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	32.10%	30.50%	29.65%
Aggregated Trees	29.30%	27.75%	26.50%

Table 6: Misclassification rates for out-of-sample classification with $q \in \{10, 15, 20\}$ gene clusters as features, based on $N = 100$ random divisions into learning set (two thirds of the data) and test set (one third of the data).

non-averaged, individual genes? To answer this important question, we also classified our datasets with exactly the same genes that were contained in the clusters, but did not average them.

Instead of q average expression profiles we then have roughly five times as many single genes as predictor variables. Misclassification rates from repeated random splitting are given in Tables 7 and 8. We observe that the aggregated tree classifier yields in 54 of 56 cases better results with cluster averages than with individual genes as input. Also the nearest neighbor classifier is in 43 of 56 cases better when used in conjunction with clusters than with single genes. Note that since the events are not independent, we cannot employ a binomial test for the null hypothesis of equal performance between clusters and single genes. An analysis of score and margin of the individual genes that were used in the clusters shows that most of them are not the strongest individually for predicting the tissue types, that is, they individually often only have mediocre scores and margins, but have very good predictive power as a group. So far, we gained evidence that our algorithm really identifies functional groups of genes whose average expression level has high explanatory power for the response classes.

Comparison to Other Studies

We now classify the breast cancer validation sample of [33], which contains four randomly chosen tissues plus five instances where two different clinical tests for determination of the ER status yielded conflicting results. We choose the nearest neighbor method with $q = 3$ clusters to be our classifier for the validation sample, as it had the best predictive potential on the $n = 38$ training data. Our predictions, shown in Table 9 always agree with the class label provided on the PNAS supporting information website, which corresponds to the outcome of the immunoblot assay method.

Not only the results on the validation sample are very convincing, but the cross validation on the $n = 38$ training tissues is also error free. This is different from the results in [33] with precedent feature selection, singular value decomposition and Bayesian binary regression, where 7 of 9 tissues in the validation sample and 36 of 38 tissues in the training sample were accurately predicted.

<i>Leukemia</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	6.33%	4.79%	4.50%	4.08%
Aggregated Trees	8.50%	6.04%	4.54%	3.92%
<i>Breast</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	1.08%	0.83%	0.92%	1.17%
Aggregated Trees	5.42%	2.50%	1.83%	2.42%
<i>Prostate</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	13.24%	10.68%	9.15%	8.44%
Aggregated Trees	25.47%	21.29%	18.56%	17.44%
<i>Colon</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	23.40%	21.95%	20.15%	18.90%
Aggregated Trees	30.95%	29.70%	30.20%	31.20%
<i>SRBCT</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	1.76%	0.86%	0.81%	1.05%
Aggregated Trees	4.38%	2.00%	2.62%	3.95%
<i>Lymphoma</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	2.43%	2.29%	1.76%	1.05%
Aggregated Trees	4.38%	2.81%	2.10%	1.00%
<i>Brain</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	30.79%	29.07%	29.50%	27.57%
Aggregated Trees	40.14%	35.29%	34.64%	33.50%
<i>NCI</i>	$q = 1$	$q = 2$	$q = 3$	$q = 5$
Nearest Neighbor	39.63%	34.89%	32.84%	31.95%
Aggregated Trees	56.58%	49.53%	44.84%	42.42%

Table 7: Benchmark misclassification rates for out-of-sample classification with the very same but non-averaged genes from $q \in \{1, 2, 3, 5\}$ clusters as features, based on $N = 100$ random divisions into learning set (two thirds of the data) and test set (one third of the data).

Moreover, our result confirms that the breast cancer expression matrix contains a strong signal for discriminating the ER status.

	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	3.67%	3.75%	3.79%
Aggregated Trees	4.83%	6.79%	8.46%
<i>Breast</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	1.33%	1.50%	1.58%
Aggregated Trees	4.17%	5.42%	8.33%
<i>Prostate</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	7.76%	8.18%	7.85%
Aggregated Trees	16.65%	17.65%	18.94%
<i>Colon</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	16.65%	16.25%	15.70%
Aggregated Trees	33.55%	34.15%	34.90%
<i>SRBCT</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	1.19%	1.43%	1.48%
Aggregated Trees	6.48%	6.95%	8.43%
<i>Lymphoma</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	0.81%	0.81%	0.86%
Aggregated Trees	0.81%	1.05%	1.24%
<i>Brain</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	28.50%	28.00%	27.50%
Aggregated Trees	34.36%	34.79%	35.29%
<i>NCI</i>	$q = 10$	$q = 15$	$q = 20$
Nearest Neighbor	30.68%	29.74%	28.95%
Aggregated Trees	39.21%	39.05%	37.79%

Table 8: Benchmark misclassification rates for out-of-sample classification with the very same but non-averaged genes from $q \in \{10, 15, 20\}$ clusters as features, based on $N = 100$ random divisions into learning set (two thirds of the data) and test set (one third of the data).

Next, we use our method to classify the original 34 test samples in the leukemia dataset. We applied the supervised cluster-

Tumor	14	31	33	44	45
Status	Neg?	Neg?	Neg?	Neg	Pos?
Prediction	Neg	Neg	Neg	Neg	Pos
Tumor	46	47	48	49	
Status	Pos?	Pos	Pos	Neg	
Prediction	Pos	Pos	Pos	Neg	

Table 9: *Classification of the breast cancer validation sample with $q = 3$ cluster expression profiles based on the training sample with 38 tumors as features and aggregated trees as predictor. The status of the tumors is according to the information provided on the PNAS-website. The question mark means that two clinical tests yielded conflicting results. Displayed here is the outcome of the immunoblot assay method.*

ing algorithm on the $n = 38$ training data, where we also fit the best predictor from our random splitting study (aggregated trees with $q = 20$ clusters as input features) as classifier for the independent sample. Our predictions turned out to be error-free, a result which can be directly compared to [32], where 29 of 34 observations were classified correctly by a weighted voting scheme. With support vector machines, results ranging between 30 to 32 correct classifications were reported [43]. Moreover, a full leave-one-out cross validation on the $n = 38$ training data (results not shown) resulted in perfect classification for various q values; also the performance for cross validation on the entire dataset with $n = 72$ observations is competitive, compared, for example, to [18].

The SRBCT data contains an additional test set of 20 SRBCT and 5 non-SRBCT samples. We first classified the 20 SRBCT tissues with the best classifier from the random splitting study on the $n = 63$ training samples, the nearest neighbor method with $q = 3$ clusters as input. The predictions turned out to be error-free, approving the perfect classification with artificial neural networks and principal components as in [36], as well as the correct

diagnosis obtained with multi-category support vector machines in [44]. As aggregated trees and the one-nearest-neighbor classifier with $q = 3$ clusters as input are not well suited for assessing prediction strengths on the 5 non-SRBCT samples, we applied logistic discrimination and rejected every classification that was done with a probability lower than 0.95. All 5 non-SRBCT’s did not exceed this threshold and were thus correctly rejected, whereas 3 of the 20 SRBCT tissues did not exceed it and could not confidently be classified either, though they were predicted correctly. Also, this result, as well as our error-rate from leave-one-out cross validation on the training data, which achieves the benchmark error-rate of 0%, are consistent with [36, 44]. This provides more evidence that our method can at least keep up with state-of-the-art classifiers such as neural networks or support vector machines.

The five remaining microarray studies do not contain genuine test sets and we thus compare our error-rates from cross validation and random splitting against the literature. The classification of tumor versus normal prostate tissue has been evaluated with leave-one-out cross validation [35]. After precedent feature selection, an accuracy of “greater than 90%” was obtained, a result that can be beaten by our error-rate of 4.90%, which corresponds to 5 misclassifications in totally 102 samples. The colon cancer datasets has already been considered by various authors, for example in [18], with classifiers based on single genes such as nearest neighbors and boosting in a cross validation study. Our method does not clearly improve their results, although it seems to have an edge over them. However, we could not accomplish a cross validation error-rate of 9.68%, as reported in [43] with support vector machines. The error-rates on the lymphoma, brain tumor and NCI data provide evidence that our method, based on a one-against-all approach, does a good job in multiclass problems as well. On the lymphoma data we observe perfect classification, thus achieve the non-to-improve benchmark. On the brain tumor data, our minimal cross validation error-rate of 11.90% is superior to the 16.67% obtained in [38] with a weighted voting

algorithm. Many more misclassifications occur on the NCI than on the other datasets, due to the large number of classes and their heterogeneity. However, when comparing our predictions to the results in a broad evaluation of classifiers on the NCI data [19], they prove to be very valuable. We consistently obtained mean error-rates of less than 30% with random splitting, the optimum is 26.50% using aggregated trees with $q = 20$ clusters, whereas the best median error-rates reported in [19] are in a range around 35% and higher.

	Leuke	Breast	Prost	Colon
Sup. Cl.	1.39%	0.00%	4.90%	16.13
Literature	1.39%	5.26%	9.80%	9.68
	SRBCT	Lymph	Brain	NCI*
Sup. Cl.	0.00%	0.00%	11.90%	26.50%
Literature	0.00%	?	16.67%	≈35%

Table 10: Best leave-one-out cross validation error-rates from our supervised clustering procedure compared to best reported results from the literature where directly comparable, references are given in the main text. *The mean error-rate on the NCI data is based on random divisions into training and test set, and compared against the median error-rate obtained under the same framework in [19]

In summary, our predictions from simple classifiers based on the supervised clustering’s output can easily keep up with sophisticated methods that are based on single genes, and as Table 10 shows, our supervised clusters beat the best reported results from the literature in four out of eight datasets. On three further datasets, we achieve the benchmark of perfect classification. The success of our method may be because the averaging of genes according to (2) has a variance reducing effect and yields more stable and accurate features for classification. Besides its good predictive potential, the cluster structure provided by our method is very accessible for biological interpretation and can be benefi-

cial for functional genomics.

3.5 Stability

The stability of the gene clusters detected by our supervised clustering algorithm is a critical issue. The output is much more useful for functional genomics if it remains unchanged for “similar” input data. We use the bootstrap as a tool for assigning statistical significance, see [45]. We assume n pairs of observations (\mathbf{x}_i, y_i) with binary response $y_i \in \{0, 1\}$, from which we form a resampled gene expression dataset

$$(\mathbf{x}_1, y_1)^*, \dots, (\mathbf{x}_n, y_n)^*$$

of length n by drawing with replacement from the original data pairs. We can then apply our supervised algorithm to extract clusters $\mathcal{G}_1^*, \dots, \mathcal{G}_q^*$ out of these resampled data. For an empirical study, we generated $L = 1000$ resampled gene expression datasets of size n to explore the compositional variability of the first cluster \mathcal{G}_1^* in eight binary problems as detailed in the caption of table 11.

We first analyze the variability in cluster size. The results, summarized in table 11, show surprising stability across the eight different datasets. We observe that quite small clusters, typically made up of 3-9 genes, were found. The standard deviation in cluster size was fairly low in all eight datasets. As a next and more difficult step, we try to explore the compositional variability of the clusters. To give a rough overview which proportion of genes is actively present in the clustering process, we assess a confidence level to each individual gene i , which measures how likely it is to be clustered,

$$\pi_i = \frac{N_i}{L} = \frac{1}{L} \cdot \sum_{\ell=1}^L 1_{[gene\ i \in \mathcal{G}_1^{*(\ell)}]}, \quad i = 1, \dots, p, \quad (8)$$

where N_i is the number of the L clusters that contain gene i . The numerical results given in table 12 show that except for the

Clustersize	Mean	Stdev	Min	Max
Leukemia	5.855	2.910	1	23
Breast Cancer	4.344	2.062	1	13
Prostate	6.327	2.373	2	17
Colon	6.642	2.733	2	20
SRBCT	4.739	1.816	1	14
Lymphoma	5.485	2.679	1	16
Brain	6.094	2.751	1	19
NCI	6.174	2.930	1	20

Table 11: Variability in size of clusters that have been shaped with the supervised algorithm, based on 1000 bootstrap replicates. Leukemia stands for distinction between AML and ALL; in the breast cancer data, the separation of the ER receptor status has been analyzed; prostate and colon stand for discrimination of normal versus tumorous tissue; in the SRBCT dataset, the Ewing family of tumors was separated against three other phenotypes; for the lymphoma dataset discrimination of DLBCL against FL & CLL was considered; in the brain tumor dataset AT/RTs were discriminated from four further malignancies; and in the NCI dataset, leukemia was separated against seven other cancers. The presented figures for the four multiclass datasets are representative for all their binary distinctions between a tumor type against all others.

colon tumor data, only a minority of genes ever entered a cluster. Also for the prostate and leukemia data this proportion was somewhat bigger, but still most of the genes never took part in the clustering process. More importantly, only a very small part of the genes is used frequently, that is, more than 50 times in the 1000 clusters. We conjecture that our supervised algorithm discriminates phenotypes with a small core of genes only, and in this sense it is reasonably stable.

We continue by assessing confidence levels to pairs of genes which gives a clue about pairwise interactions. We count the

Active Genes	$\#(\pi_i > 0)$	in %	$\#(\pi_i > \frac{1}{20})$	in %
Leukemia	624	17.474%	18	0.504%
Breast	128	1.803%	9	0.130%
Prostate	949	15.730%	16	0.265%
Colon	1028	51.400%	12	0.600%
SRBCT	68	2.946%	11	0.477%
Lymphoma	279	6.930%	19	0.472%
Brain	345	6.164%	21	0.375%
NCI	227	4.329%	23	0.439%

Table 12: Number and proportion of genes that ever have been used in the first cluster \mathcal{G}_1^* (first two columns), as well as number and proportion of genes that have been used for cluster \mathcal{G}_1^* in more than 50 out of the 1000 bootstrap trials (last two columns). The selection of data is identical to table 11.

number N_{ij} of clusters \mathcal{G}_1^* found with our bootstrapped gene expression datasets that both contain the genes i and j , and then divide by the number of replicates L ,

$$\pi_{ij} = \frac{N_{ij}}{L} = \frac{1}{L} \cdot \sum_{\ell=1}^L 1_{[gene\ i \in \mathcal{G}_1^{*(\ell)}]} \cdot 1_{[gene\ j \in \mathcal{G}_1^{*(\ell)}]}, \quad (9)$$

for $i, j \in \{1, \dots, p\}$. These confidence levels not only give an idea how likely the pairs are, but also provide information for functional genomics, as we can now analyze whether pairs of genes preferentially enter clusters simultaneously or not. The number of hits N_i for individual genes i follows a Binomial(L, π_i) distribution (given the data), and for pairs (i, j) we have that N_{ij} is Binomial(L, π_{ij}) (we ignore here the fact that π_i in (8) and π_{ij} in (9) are computed with $L = 1000$ replicates instead of the theoretical $L = \infty$). If there were no attraction or repulsion between genes, the joint probability π_{ij} would be given by the product $\pi_i \pi_j$ of the marginal probabilities. By calibrating the observed number of hits N_{ij} with the Binomial($L, N_i N_j / L$) distribution

under independence, we can test the hypothesis

$$H_0 : \pi_{ij} = \pi_i \pi_j,$$

and compute the associated p-values. Low p-values indicate significant pairs of genes. Moreover, we also distinguish between two genes which are attracting (with N_{ij} larger than expected under the null hypothesis), and which are repelling (with N_{ij} lower than expected under H_0). We implemented an empirical analysis based on $L = 1000$ bootstrap trials, for pairs made up of the five genes with the highest confidence levels π_i in the discrimination of lymphoma class 0 (DLBCL) against the other two phenotypes. Numerical results are summarized in table 13, clone numbers and function of the genes are given in table 14. Among the 10 pairs, several significant gene pairs which are strongly attracting or repelling are present; for example, the genes 3786 and 3804 strongly attract each other. Moreover, 78% of the clusters that contained gene 3804 also included gene 3786, again signifying a special relation between these two. An interpretation of such facts in the framework of functional genomics is beyond the scope of this paper.

It is now tempting to extend this kind of analysis from pairs to tuples of third and higher orders. But estimating higher-order interactions will become very unreliable due to the limited amount of sample size n .

3.6 Additional Modifications

Our supervised clustering procedure can be understood as a generic method and allows alteration of various details according to the users' choice and specific demands. We also tried to improve the supervised clustering procedure ourselves with additional modifications, the most important of which are described here. The averaging of the gene expression in (2) is specified by the arithmetic mean plus sign-flips, a very simple linear combination of genes, since it is impractical to repeatedly optimize a

<i>Numbers</i>	Gene 3786	Gene 3804	Gene 761	Gene 780
Gene 3763	184 (301)	68 (220)	144 (155)	173 (133)
Gene 3786		289 (187)	153 (132)	72 (113)
Gene 3804			136 (96)	60 (83)
Gene 761				40 (58)
<i>p-values</i>	Gene 3786	Gene 3804	Gene 761	Gene 780
Gene 3763	(-) 0.000	(-) 0.000	(-) 0.359	(+) 0.001
Gene 3786		(+) 0.000	(+) 0.055	(-) 0.000
Gene 3804			(+) 0.000	(-) 0.007
Gene 761				(-) 0.015

Table 13: Top: numbers of observed and expected (under hypothesis of independence, numbers in parentheses) gene pairs of the five most frequently clustered genes in the distinction of DLBC-lymphoma against the other two phenotypes, based on 1000 bootstrap replicates. Bottom: p-values for attraction (+) and repulsion (-) of gene pairs from two-sided binomial tests that compare the joint probability against the product of the marginals.

Sign	Gene	Clone	Function
-	3763	769861	CD63 (melanoma 1) antigen
-	3786	345538	Cathepsin L
-	3804	343867	Allograft-inflammatory factor-1
+	761	1341294	Unknown
+	780	1334411	Unknown UG Hs.32553 ESTs

Table 14: Clone numbers and function description of the five genes that have been clustered most frequently in the discrimination of DLBC-lymphoma against the other two phenotypes in the lymphoma dataset.

general linear combination such as

$$\mathbf{X}_{G_i} = \sum_{g \in G_i} \beta_g \mathbf{X}_g \quad \text{with} \quad \sum_g |\beta_g| = 1$$

during the clustering process. But theoretically, once the cluster algorithm has done its work, we could try to improve the discriminatory power of the actual cluster by numerically optimizing a weighted linear combination as above with respect to score and margin. In practice, we recognized that the numerical optimization was very difficult. If we started it with equal weights, they only changed slightly and the objective function (this is, the margin) did not improve much. Because of this we favor the more simple method.

Since the margin function in (6) is not scale-invariant, we also considered clustering with an adjusted margin. This means that we optimized the quotient of margin and within group variation for a gene expression vector $\xi_i = (x_{i1}, \dots, x_{in})$,

$$\text{Adjusted margin}(\xi_i) = \frac{\text{Margin}(\xi_i)}{\sqrt{s_0^2/n_0 + s_1^2/n_1}}.$$

Here, n_k is the size and s_k^2 is the sample variance of class $k \in \{0, 1\}$. While theoretically, the size of the gap between the two response classes is meaningful only in relation to the within-group variance, the adjustment of the margin proved not to be very important in practice, owing to the use of standardized gene expression data. It did not improve the predictive performance of the clusters and slightly deteriorated their stability. Since it is common practice to standardize expression data, we recommend to work with the non-adjusted margin.

Our algorithm, as described in section 2, yields disjoint clusters of genes. To account for the fact that genes may function in multiple pathways one could modify as follows. First, run the clustering algorithm on the data, producing a first cluster; second, compute a probability estimate for $\mathbb{P}[Y = 1|\mathbf{X}]$ for a two-class problem, for example with probability based classification methods or in a logistic model; third, reweight the data with weights as in the Real AdaBoost algorithm ([16]; algorithm 2, p.340); then, return to the first step but now with reweighted data. Doing the loop q times produces q clusters which are allowed to be non-disjoint.

We also explored the improvement of the supervised clustering algorithm by biasing it towards larger clusters. Specifically, we did not stop the forward search when score and/or margin first worsened, but continued as long as the objective function remained within a factor of the best. Our intention was that the objective function could improve again and reach even better values. As soon as the objective function once dropped below the tolerance (a factor times the best ever achieved value), we stopped the forward search and continued the algorithm with the cluster that yielded the best parameters ever. While our first guess was that the biasing could result in larger clusters with clearer separation, it rarely ever had any effect in practice.

4. Conclusions

We have proposed an algorithm for supervised clustering of genes from microarray experiments. Our procedure is potentially useful in the context of medical diagnostics, as it identifies groups of interacting genes that have high explanatory power for given tissue types, and which in turn can be used to accurately predict the class labels of new samples. At the same time, such gene clusters may reveal insights into biological processes and may be valuable for functional genomics.

In summary, our algorithm tries to cluster genes such that the discrimination of different tissue types is as simple as possible. It builds the clusters incrementally and relies on a fast, stepwise strategy that allows an efficient, non-exhaustive search among thousands of genes. More specifically, the aim is to identify sparse linear combinations of genes, whose average expression level is uniformly low for one response class and uniformly high for the other class(es).

In empirical studies, the average cluster expression profiles showed superior classification potential compared to other techniques where unclustered genes had been used. The clusters showed reasonable stability and there are several reasons that point towards their biological significance: a) they do not only

contain the genes which are individually good, but groups of genes whose consensus expression profile is best with respect to the objective function; b) the predictive potential of the very same, non-averaged genes cannot keep up with the one of the corresponding cluster means; and c) an application of our algorithm to randomly permuted data points out that the identified structure is more than just a noise artifact.

An important task which remains to be addressed in future research is the generalization of the supervised clustering algorithm to quantitative response variables and to censored survival data. The fundamental idea of supervised clustering can be pursued again, but needs alternative objective functions that rank individual genes and gene clusters based on their explanatory power for non-categorical response variables.

Finding Predictive Gene Groups from Microarray Data

Marcel Dettling, Peter Bühlmann
ETH Zürich

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 *penalized logistic regression analysis*, 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.

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 [25], 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 [24], 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 [10]. Another single-step approach based on Rissanen’s minimum description length principle was pursued by Jörnsten and Yu [13].

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 [10], which is called *Wilma*, since its grouping criterion is based on the *Wilcoxon* and *margin* statistics. Then, we present *Pelora*, a novel approach to supervised grouping of genes, using an objective function based on *penalized logistic regression analysis*. 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. Motivation for Supervised Grouping

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,

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

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

$$\begin{aligned} \mathbf{x}_1, \dots, \mathbf{x}_{n_0} & \text{ iid with cdf } F(\cdot - \mu^{(0)}) \text{ in pop. 0,} \\ \mathbf{x}_{n_0+1}, \dots, \mathbf{x}_n & \text{ iid with cdf } F(\cdot - \mu^{(1)}) \text{ in pop. 1,} \end{aligned} \quad (10)$$

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 (10) becomes a simple two-population group model if

$$\begin{aligned} \mu^{(0)} &= (\mu_{\mathcal{G}_1}^{(0)}, \dots, \mu_{\mathcal{G}_1}^{(0)}, \dots, \mu_{\mathcal{G}_q}^{(0)}, \dots, \mu_{\mathcal{G}_q}^{(0)}), \\ \mu^{(1)} &= (\mu_{\mathcal{G}_1}^{(1)}, \dots, \mu_{\mathcal{G}_1}^{(1)}, \dots, \mu_{\mathcal{G}_q}^{(1)}, \dots, \mu_{\mathcal{G}_q}^{(1)}), \end{aligned} \quad (11)$$

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

The magnitude of the difference $|\mu_{\mathcal{G}}^{(0)} - \mu_{\mathcal{G}}^{(1)}|$ 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 (10), with the cumulative distribution function $F(\cdot)$ chosen as the $\mathcal{N}_{10}(0, I)$ -distribution. Figure 8 shows the scatterplot of two genes from this group \mathcal{G} with $\mu_{\mathcal{G}}^{(0)} = -3$, and $\mu_{\mathcal{G}}^{(1)} = 3$, 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 8. 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.

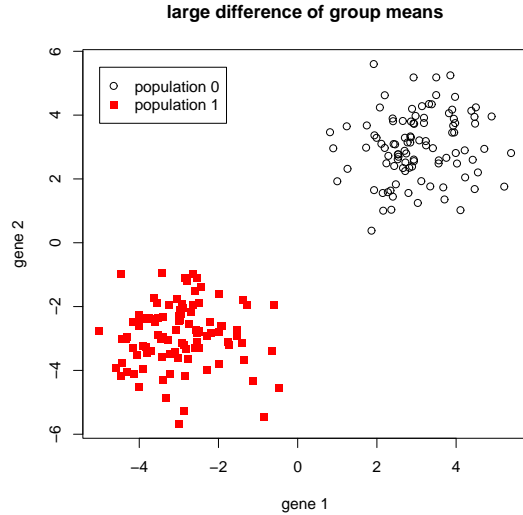


Figure 8: Scatterplot of two genes from a group \mathcal{G} with $\mu_{\mathcal{G}}^{(0)} = -3$, $\mu_{\mathcal{G}}^{(1)} = 3$.

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 9. The correlation of 0.53, which is low in the context of microarray 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

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

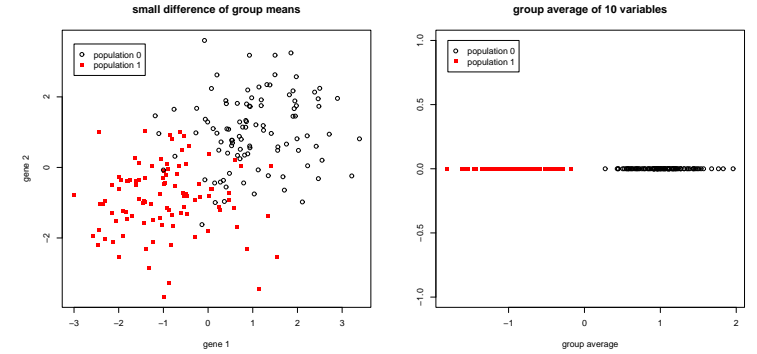


Figure 9: Left: scatterplot of two genes from a group \mathcal{G} with $\mu_{\mathcal{G}}^{(0)} = -1$ and $\mu_{\mathcal{G}}^{(1)} = 1$. Right: average expression \tilde{x} of a group with size $|\mathcal{G}| = 10$.

and check how well the group average \tilde{x} separates the two population groups. This is demonstrated in the right panel of Figure 9 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 9 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.

$$\max_{i, y_i=0} \tilde{x}_i < \min_{i, y_i=1} \tilde{x}_i \quad \text{or} \quad \min_{i, y_i=0} \tilde{x}_i > \max_{i, y_i=1} \tilde{x}_i. \quad (12)$$

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 9. 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,

$$\sqrt{\text{Var}(\tilde{x}|y)} \sim C_y / \sqrt{|\mathcal{G}|}$$

for some constant $C_y > 0$ as $|\mathcal{G}| \rightarrow \infty$, which will be small relative to $|\mu_{\mathcal{G}}^{(0)} - \mu_{\mathcal{G}}^{(1)}|$ if $|\mathcal{G}|$ is large.

2.2 Beyond the Two-Population Group Model

The two-population group model in (11) 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 (12) if

$$|\bar{\mu}_{\mathcal{G}}^{(0)} - \bar{\mu}_{\mathcal{G}}^{(1)}|$$

is large, relative to

$$\max\left(\sqrt{\text{Var}(\tilde{x}|y=0)}, \sqrt{\text{Var}(\tilde{x}|y=1)}\right),$$

where

$$\bar{\mu}_{\mathcal{G}}^{(y)} = \frac{1}{|\mathcal{G}|} \sum_{g \in \mathcal{G}} \mu_g^{(y)} \text{ and } y \in \{0, 1\}.$$

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 $\text{Cov}(x_g, x_{g'}|y)$ are low for all genes $g, g' \in \mathcal{G}$, yielding small conditional variances $\text{Var}(\tilde{x}|y)$.

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 $|\bar{\mu}_{\mathcal{G}}^{(0)} - \bar{\mu}_{\mathcal{G}}^{(1)}|$ will be smaller (which is worse) for any superset group $\mathcal{G} \supset g^*$, while the conditional variances $\text{Var}(\tilde{x}|y)$ 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 $\tilde{x}_1, \dots, \tilde{x}_q$, at least as long as these q group representatives are not too strongly conditionally dependent given the binary response $y \in \{0, 1\}$.

2.3 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 $\tilde{x}_1, \dots, \tilde{x}_q$ 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

$$\mathbf{P}[y = 1|\mathbf{x}] = f(\tilde{\mathbf{x}}) \text{ with } \tilde{\mathbf{x}} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_q), \quad (13)$$

where $f(\cdot)$ is an unknown nonlinear function and \tilde{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

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

as the representative group value. The unknown discrete parameter α_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 (13) 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" [10]. The name *Wilma* is an acronym for the *Wilcoxon* and *margin* 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 ℓ_2 -penalized negative log-likelihood function

$$S = - \sum_{i=1}^n (y_i \log p_\theta(\tilde{\mathbf{x}}_i) + (1 - y_i) \log(1 - p_\theta(\tilde{\mathbf{x}}_i))) + n \frac{\lambda}{2} \theta^T P \theta, \quad (14)$$

based on the estimated probabilities $p_\theta(\tilde{\mathbf{x}}) = \mathbb{P}_\theta[y = 1|\tilde{\mathbf{x}}]$ from penalized logistic regression analysis, hence the name *Pelora*.

Note that θ is the parameter vector, λ 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 $\tilde{\mathbf{x}} = (1, \tilde{x}_1, \dots, \tilde{x}_q)$, 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 [46] has been used as a stand-alone for classification of microarray gene expression data with single genes. Eilers et al. [47] as well as Zhu and Hastie [48] 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

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

for observations $i = 1, \dots, n$, with parameter vector $\theta^T = (\theta_0, \theta_1, \dots, \theta_q)$ and $x_{i0} = 1$. The idea of penalized logistic regression is to estimate θ by a ℓ_2 -penalized maximum likelihood

principle. We minimize

$$S(\theta) = -\sum_{i=1}^n (y_i \log p_\theta(\tilde{\mathbf{x}}_i) + (1 - y_i) \log(1 - p_\theta(\tilde{\mathbf{x}}_i))) + n \frac{\lambda}{2} \theta^T P \theta \quad (15)$$

for fixed $\tilde{\mathbf{x}}_i$ with respect to θ . Note that (14) and (15) are identical, but the goal in (15) is to estimate the parameter vector θ by minimizing S for fixed predictors, whereas for supervised grouping, we try to find the (possibly overlapping) partition whose centroid-predictors optimize S in (14) with optimal parameter θ from (15). P is the penalty matrix, defined as

$$P = \begin{pmatrix} 0 & 0 & \dots & 0 & 0 \\ 0 & \text{Var}(\tilde{x}_1) & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & \text{Var}(\tilde{x}_{(q-1)}) & 0 \\ 0 & 0 & \dots & 0 & \text{Var}(\tilde{x}_q) \end{pmatrix} \quad (16)$$

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 (15), we take derivatives with respect to θ ,

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

where $\tilde{X} = (1, \tilde{x}_{i1}, \dots, \tilde{x}_{iq})_{i=1, \dots, n}$ 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 θ^{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

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

where the matrix W_θ is a diagonal weight matrix, defined as

$$W_\theta = \text{diag}((p_\theta(\tilde{\mathbf{x}}_i)(1 - p_\theta(\tilde{\mathbf{x}}_i)))_{i=1, \dots, n}).$$

Then, we plug in and with

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

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

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

where $\bar{y} = \frac{1}{n} \sum y_i$. This means that $p_{\theta^{(0)}}(\tilde{\mathbf{x}}_i) = \bar{y}$, 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

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

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 λ is chosen appropriately. The second Newton-Raphson iteration typically yields asymptotic efficiency, see [49]. 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

Figure 10 gives the details about our supervised grouping procedure. 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 (14). 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 [50, 51]. 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

Algorithm for Finding Predictive Gene Groups

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

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

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,

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

for $j = 1, \dots, (q-1)$ and $\alpha_g \in \{-1, 1\}$, where $|\mathcal{G}_j|$ is the number of genes in group \mathcal{G}_j . The optional parameter α_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 θ^{+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 (14). 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 $\mathcal{G}_1, \dots, \mathcal{G}_{(q-1)}$ unchanged, 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

$$\begin{aligned} \tilde{x}_q^{+g} &= \frac{|\mathcal{G}_q| \cdot \tilde{x}_q + 1 \cdot x_g}{|\mathcal{G}_q| + 1}, \quad \text{and} \\ \tilde{x}_q^{-g} &= \frac{|\mathcal{G}_q| \cdot \tilde{x}_q + (-1) \cdot x_g}{|\mathcal{G}_q| + 1}. \end{aligned}$$

Fit penalized logistic regression with predictors $\tilde{\mathbf{x}}^{+g} = (1, \tilde{x}_1, \dots, \tilde{x}_q^{+g})$ and $\tilde{\mathbf{x}}^{-g} = (1, \tilde{x}_1, \dots, \tilde{x}_q^{-g})$ to obtain the parameter vectors θ^{+g} and θ^{-g} , as well as conditional probabilities $p_{\theta^{+g}}(\tilde{\mathbf{x}}^{+g})$ and $p_{\theta^{-g}}(\tilde{\mathbf{x}}^{-g})$. Compute the penalized negative log-likelihoods S^{+g}, S^{-g} as in (14). 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 $\mathcal{G}_1, \dots, \mathcal{G}_q$ 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 its polarity parameter α_{g^*} and update group, group centroid and criterion value to

$$\alpha_{g^*} \leftarrow \text{sign}(S^{-g^*} - S^{+g^*}), \quad \mathcal{G}_q \leftarrow \mathcal{G}_q \cup \{g^*\},$$

$$\tilde{x}_q \leftarrow \frac{|\mathcal{G}_q| \cdot \tilde{x}_q + \alpha_{g^*} \cdot x_{g^*}}{|\mathcal{G}_q| + 1}, \quad S^{old} \leftarrow S^{g^*}.$$

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

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

Fit penalized logistic regression with predictor $\tilde{\mathbf{x}}^g = (1, \tilde{x}_1, \dots, \tilde{x}_q^g)$ to obtain the parameter vector θ^g and conditional probabilities $p_{\theta^g}(\tilde{\mathbf{x}}^g)$. Compute the penalized negative log-likelihood S^g as in (14).

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 $\mathcal{G}_1, \dots, \mathcal{G}_q$ (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.

Figure 10: Algorithm for finding predictive gene groups from microarray data

called `supclust`. Please refer to 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 [17], 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 $y \in \{0, \dots, K-1\}$ to binary problems exist, see [31] 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 $k = 0, \dots, K-1$, and running the supervised grouping algorithm K times on the dichotomous-response datasets $(\mathbf{x}_1, y_1^{(k)}), \dots, (\mathbf{x}_n, y_n^{(k)})$ as explained above. For each binary problem, this finally yields q group centroids $\tilde{x}_1^{(k)}, \dots, \tilde{x}_q^{(k)}$ 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 (14) 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, see Figure 11.

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 \tilde{x}_j consists of one single gene

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 θ^{+g} and θ^{+k} , as well as conditional class probabilities $p_{\theta^{+g}}(\tilde{\mathbf{x}}^{+g}), p_{\theta^{+k}}(\tilde{\mathbf{x}}^{+k})$. Compute the penalized negative log-likelihoods S^{+g}, S^{+k} as in (14). 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 $\tilde{x}_q = x_{g^*}$ and continue with step 4. Else, if $\min(S^{+g^*}, S^{+k^*}) = S^{+k^*}$, pick up covariate k^* into the predictor, set $\tilde{x}_q = u_{k^*}$ and restart at step 3 to identify the next predictor variable.

Figure 11: Alterations on the *Pelora* algorithm for incorporating clinical variables into the grouping process.

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 *forward selection* approach 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 ℓ_2 -penalty in the objective function S . In view of the ridge-type ℓ_2 -penalty, *Forsela* penalizes every gene (standardized to variance one) by the same amount while the matrix P for *Pelora*, appearing in (14), 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 2.2.

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 variability 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 [52]. 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 ℓ_2 -penalized residual sum of squares

$$S = \sum_{i=1}^n (y_i - m_\theta(\tilde{\mathbf{x}}_i))^2 + \frac{n}{2} \lambda \theta^T P \theta, \quad (17)$$

based on $m_\theta(\tilde{\mathbf{x}}_i)$ from (18), where θ is the parameter vector, λ is the tuning parameter and P is the non-unit penalty matrix from equation (16). The $(q+1)$ -dimensional predictor is $\tilde{\mathbf{x}} = (1, \tilde{x}_1, \dots, \tilde{x}_q)$. For continuous response y , the residual sum of squares is the 'natural' loss criterion and we rely on the classical linear model

$$m_\theta(\tilde{\mathbf{x}}_i) = \sum_{j=0}^q \theta_j \tilde{x}_{ij} = \tilde{\mathbf{x}}_i \theta, \text{ for observations } i = 1, \dots, n. \quad (18)$$

The notion behind ridge regression [53] is to estimate the parameter vector θ by minimizing S from (17) with respect to θ . Setting derivatives to zero leads to $(q+1)$ linear equations, which can be solved as

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

representing an explicit solution for minimizing S in 17. Thus, the Newton-Raphson approximation is not necessary, and we directly obtain the exact solution.

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. [32]:

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 [19], 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. [33]:

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. [34]:

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. [35]:

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 [35], 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. [37]:

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 [19].

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 12 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.



Figure 12: 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 15 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 λ is discussed in section 4.2 on page 87. The group size slightly diminishes with stronger penalization (increasing λ),

but the differences are not very big. Note that with *Wilma*, our supervised algorithm from [10], 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*.

	Colon	Leuke	Estro	Nodal	Prost	Lymph
mean	14.0	12.1	15.4	14.8	17.9	15.8
st. dev.	5.3	3.2	4.2	4.3	9.0	3.5

Table 15: *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.*

It is beyond the scope of our paper to judge the functional relevance and the 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 $\tilde{x}_1, \dots, \tilde{x}_q$ and class label prediction is done with *Pelora*'s conditional probabilities $p_\theta(\tilde{\mathbf{x}})$ via

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

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

$p_\theta(\tilde{\mathbf{x}}^{(k)})$ for every binary problem $k = 0, \dots, K - 1$ and assign the class label

$$\hat{y}(\tilde{\mathbf{x}}^{(0)}, \dots, \tilde{\mathbf{x}}^{(K-1)}) = \arg \max_k p_\theta(\tilde{\mathbf{x}}^{(k)}).$$

Instead of working with the built-in classifier, we could also use the group centroids $\tilde{x}_1, \dots, \tilde{x}_q$ as input for alternative methods like the nearest-neighbor rule [19], (possibly restricted) linear or quadratic discriminant analysis [19] or support vector machines [43], and many more. However, extensive experimentation (data not shown) yielded no improvement with these alternative methods compared to the built-in classifier.

	$q = 2$	$q = 4$	$q = 6$	$q = 8$	$q = 10$
$\lambda = 1$	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 16: *Misclassification rates for Pelora's built-in classifier with different parameter values λ 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 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 *Pelora*'s two free parameters: the number of groups q_{final} and the penalty parameter λ . 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 λ 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 $q_{final} \in \{1, 2, \dots, 10\}$ 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 16 and Figure 13. 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 [47] yielded 3 false predictions, whereas the combination of penalized logistic regression and recursive feature elimination proposed in [48] also achieved our result of 1 misallocation.

Figure 13 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 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

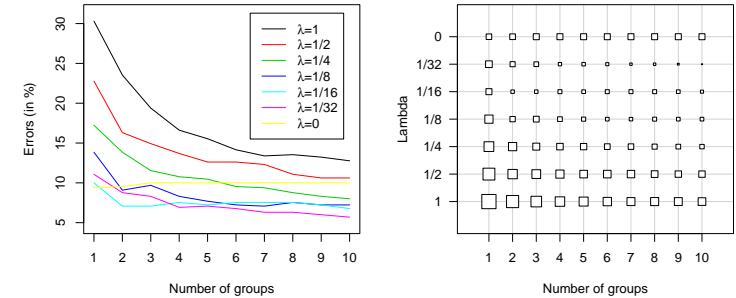


Figure 13: Graphical representation of misclassification rates for *Pelora's* built-in classifier with different parameter values λ and q_{final} , based on 50 random splits of the leukemia training dataset into learning sets of 25 observations and validation sets of 13 tissues. In the right panel, the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

misclassification rates do not fluctuate wildly and are close to optimal over a larger range of q_{final} and λ .

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 16 and Figure 13 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* [10], 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 [10], 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. [19], 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 [19]. 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.

According to Table 17 and Figure 14, the predictive potential of supervised groups’ centroids is convincing. We observe that our former implementation *Wilma* has an edge over *Pelora*

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

Table 17: 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).

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.

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

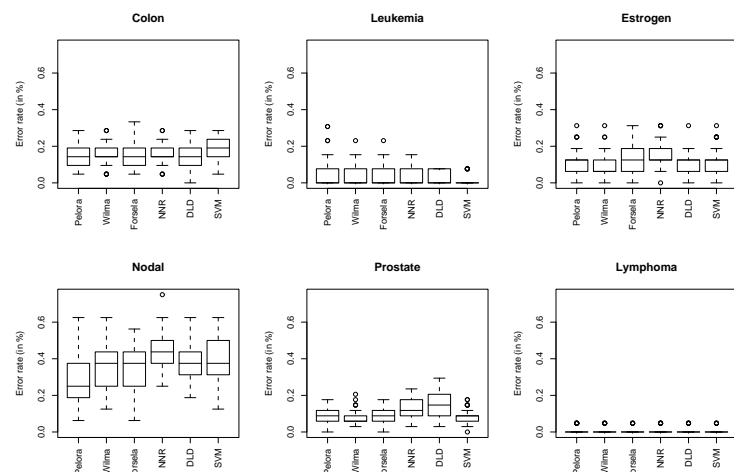


Figure 14: Box and whisker plots, showing the variation of the misclassification rates over 50 random splits into learning set ($\frac{2}{3}$ of data) and validation set ($\frac{1}{3}$ of data) for 6 different classifiers: Pelora and Wilma with $q = 10$ groups, Forsela with $q = 30$ single genes, as well as the 1-nearest-neighbor rule (NNR), diagonal linear discriminant analysis (DLD) and a support vector machine (SVM), based on 200 single genes.

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 Analysis

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. [54]. 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 angiogenesis $\in \{0, 1\}$.

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. [54] 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 angiainvasion.

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

Table 18: *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 angiainvasion.*

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 and obtained the parameter vector $\hat{\theta}^{test} = (\hat{\theta}_0^{test}, \dots, \hat{\theta}_q^{test})$. 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 $\hat{\theta}^{(b)} = (\hat{\theta}_0^{(b)}, \dots, \hat{\theta}_q^{(b)})$. For quantifying the significance of each predictor variable, we computed the

$(1 - \alpha)$ -bootstrap confidence intervals

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

where $q_{j,\alpha}$ is the α -quantile of the bootstrap distribution. Inverting these intervals leads to the *p*-values reported in Table 18. 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.

6. Proofs

Here, we prove that penalized logistic regression with non-standardized predictor $\tilde{\mathbf{x}} = (1, \tilde{x}_1, \dots, \tilde{x}_q)$ and the non-unit penalty matrix P from (16) yields equivalent parameter estimates and the same fitted values as when working with the unit penalty matrix $Q = \text{diag}(0, 1_{q \times q})$ and standardized predictor $\tilde{\mathbf{u}} = (\frac{1}{s_0}, \frac{\tilde{x}_1}{s_1}, \dots, \frac{\tilde{x}_q}{s_q})$, where $s_0 = 1$ per definition and s_j , for $j = 1, \dots, q$, is the (empirical) standard deviation of \tilde{x}_j . The

classical logistic model can then be formulated equivalently as

$$\log \left(\frac{p_\theta(\tilde{\mathbf{x}}_i)}{1 - p_\theta(\tilde{\mathbf{x}}_i)} \right) = \sum_{j=0}^q \theta_j \tilde{x}_{ij} = \sum_{j=0}^q \gamma_j \tilde{u}_{ij} = \log \left(\frac{p_\gamma(\tilde{\mathbf{u}}_i)}{1 - p_\gamma(\tilde{\mathbf{u}}_i)} \right) \quad (19)$$

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 (19) 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

$$\begin{aligned} \hat{\theta} &= \arg \min_{\theta} \left[- \sum_{i=1}^n (y_i \log p_\theta(\tilde{\mathbf{x}}_i) + (1 - y_i) \log(1 - p_\theta(\tilde{\mathbf{x}}_i))) \right. \\ &\quad \left. + n \frac{\lambda}{2} \theta^T P \theta \right] \\ \hat{\gamma} &= \arg \min_{\gamma} \left[- \sum_{i=1}^n (y_i \log p_\gamma(\tilde{\mathbf{u}}_i) + (1 - y_i) \log(1 - p_\gamma(\tilde{\mathbf{u}}_i))) \right. \\ &\quad \left. + n \frac{\lambda}{2} \gamma^T Q \gamma \right]. \end{aligned}$$

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 $\hat{\gamma}_j = \hat{\theta}_j s_j$, from which the claim follows.

Boosting for Tumor Classification

Marcel Dettling, Peter Bühlmann
ETH Zürich

Abstract

Motivation: Microarray experiments generate large datasets with expression values for thousands of genes but not more than a few dozens of samples. Accurate supervised classification of tissue samples in such high-dimensional problems is difficult but often crucial for successful diagnosis and treatment. A promising way to meet this challenge is by using boosting in conjunction with decision trees.

Results: We demonstrate that the generic boosting algorithm needs some modifications to become an accurate classifier in the context of gene expression data. In particular, we present a feature preselection method, a more robust boosting procedure and a new approach for multi-categorical problems. This allows for slight to drastic increase in performance and yields competitive results on several publicly available datasets.

Availability: Software for the modified boosting algorithms, as well as for decision trees is available for free as *R* package at <http://stat.ethz.ch/~dettling/boosting.html>.

1. Introduction

The recently developed microarray technology allows for measuring expression levels of thousands of genes simultaneously. We focus on the case where the experiments monitor gene expression values of different individuals or tissue samples, and where each experiment is equipped with an additional categorical outcome variable describing a cancer (pheno)type. In such a supervised setting, our goal is to predict the unknown class label of a new individual on the basis of its gene expression profile, since precise diagnosis of cancer type is often crucial for successful treatment. Given the availability of efficient classification techniques, bio-molecular information could become as, or even more important than traditional clinical factors.

Classification of different phenotypes, predominantly cancer types, using microarray gene expression data has been considered by numerous authors [18, 19, 32, 33, 34, 43, 55, 56]. The methods used in these studies range from classical discriminant analysis over Bayesian approaches and clustering methods to flexible tools from machine learning such as bagging, boosting and support vector machines. Explicitly, boosting decision trees has been applied for the classification of gene expression data in Ben-Dor et al. [18] and in Dudoit et al. [19]. Both studies compare the original AdaBoost algorithm that was proposed by Freund and Schapire [15] against other classifiers, and both recognize that boosting does not yield very impressive results.

In this paper we demonstrate that the performance of boosting for classification of gene expression data can often be drastically improved by modifying the algorithm as follows: First, we perform feature preselection with the nonparametric scoring method of [27]. Then, we apply the LogitBoost procedure introduced by Friedman et al. [16] instead of the AdaBoost procedure. The former was found to have a slight edge over AdaBoost in a variety of more traditional classification problems [16], and it usually performs better on noisy data or when there are mis-specifications or inhomogeneities of the class labels in the training data, which is frequently the case with microarray gene expression

data. Finally, if discrimination has to be done for more than two tumor types, we reduce multiclass to multiple binary problems so that different gene subsets and different model complexity for discriminating different tumor types are allowed. This multiclass approach turns out to be much more accurate than the direct multiclass LogitBoost algorithm of [16]. On six publicly available datasets and with a simulation study we show that the sum of these modifications leads to a classification procedure which performs very competitively, does not require sophisticated fine tuning and is fairly easy to implement.

2. Methods

2.1 The Stochastic Framework

We assume that we are given n training data pairs

$$(x_1, y_1), \dots, (x_n, y_n),$$

with $x_i \in \mathbb{R}^p$ and $y_i \in \{0, \dots, J-1\}$, which are independent and identically distributed realizations of a random vector (X, Y) . The interpretation is that the feature or input vector X models the p -dimensional gene expression profile and the response or output variable Y denotes the class label. Today, the sample size n is typically in the range of 20 to 80 and the number of monitored genes p varies between 2,000 and 20,000.

In the standard classification problem, the goal is to predict the class label Y , based on the expression vector X . This amounts to construct a classifier function

$$\mathcal{C} : X \mapsto \mathcal{C}(X) \in \{0, \dots, J-1\},$$

which can subsequently be used to predict the unknown class label of a *new* tissue sample based on its expression vector. The optimal classifier is such that the misclassification risk

$$\mathbf{P}[\mathcal{C}(X) \neq Y] \text{ is minimal.} \quad (20)$$

Note that this quantity is most often different from zero. The solution of (20) requires knowledge of the true, but generally inaccessible conditional probability distribution $\mathbf{P}[Y = j|X]$ and is called *Bayes classifier*,

$$\mathcal{C}_{Bayes}(X) = \operatorname{argmax}_{j \in \{0, \dots, J-1\}} \mathbf{P}[Y = j|X]. \quad (21)$$

In practice, it can be constructed via estimated conditional probabilities $\hat{\mathbf{P}}[Y = j|X]$. This is a classical task for $p \ll n$, but expression data with many more features than samples ($p \gg n$) create a new challenge. A promising way to find a good discriminative model is by using boosting in conjunction with decision trees.

2.2 Binary Classification

We focus first on binary problems with response $Y \in \{0, 1\}$. The best way to handle multi-categorical problems is explained later in section 2.3.

Feature Preselection

The intrinsic problem with classification from microarray data is that sample size n is much smaller than the dimensionality of the feature space, i.e. the number of genes p . Many genes are non-differentially expressed across the samples and irrelevant for phenotype discrimination. Dimensionality reduction of the feature space has been performed by many authors, see for example [18], [19] and [32], among others. It drastically eases the computational burden and for many problems improves class prediction due to the reduced amount of noise. Our feature selection is based on scoring each individual gene g , with $g \in \{1, \dots, p\}$, according to its strength for phenotype discrimination. We use a nonparametric method that is based on ranks and was presented by [27]. It is in fact equivalent to the test statistic of Wilcoxon's two sample test,

$$\text{Score}(g) = s(g) = \sum_{i \in \mathcal{N}_0} \sum_{j \in \mathcal{N}_1} 1_{[x_j^{(g)} - x_i^{(g)} \leq 0]},$$

where $x_i^{(g)}$ is the expression value of gene g for individual i and \mathcal{N}_m represents the set of the n_m indices $\in \{1, \dots, n\}$ having response in $m \in \{0, 1\}$. The score function can be interpreted as counting for each individual having response value zero, the number of instances with response one that have smaller expression values, and summing up these quantities. Viewing it as Wilcoxon's test statistic, it allows ordering the genes according to their potential significance. It captures to what extent a gene g discriminates the response categories and it is easy to notice that both values near the minimum score zero and the maximum score $n_0 n_1$ indicate a differentially expressed, informative gene. The quality measure

$$q(g) = \max(s(g), n_0 n_1 - s(g))$$

thus gives the highest values to those genes whose expression levels have the best strength for phenotype discrimination. We then simply take the $\tilde{p} \leq p$ genes with the highest values of $q(g)$ as our top features and restrict the boosting classifier to work with this subset. The number of predictor variables is a tuning parameter whose optimal value varied across different datasets. A formal choice of \tilde{p} is possible via cross validation on the training data or by determining the correct null distribution by bootstrap methods and a decision on significance levels as in [27].

Many more variable selection criteria for gene expression data have been proposed in the literature. We think that our approach based on Wilcoxon's test statistic is more suitable in the context of gene expression data than the t-statistic used in [19], or the TNoM score [18] which corresponds to counting the number of errors made by the best stump, a decision tree with two terminal nodes. The situation is similar to the trade-off between t-, Wilcoxon- and sign-test. It is known from robustness theory that the t-test is highly sensitive to outliers and (even small) deviations from the normal distribution, whereas the sign-test (TNoM score) wastes useful information about the magnitude of gene expression levels. A good compromise is the Wilcoxon-test which has nearly optimal power properties over a large class of data-

generating distributions, see [57].

Binary LogitBoost with Decision Trees

Boosting, first introduced in [15] has been found to be a powerful classification technique with remarkable success on a wide variety of problems, especially in higher dimensions. It aims at producing an accurate combined classifier from a sequence of *weak* (or *base*) classifiers, which are fitted to iteratively reweighted versions of the data. In each boosting iteration m , with $m \in \{1, 2, \dots, M\}$, the observations that have been misclassified at the previous step have their weights increased, whereas the weights are decreased for those that were classified correctly. The m th weak classifier $f^{(m)}$ is thus forced to focus more on individuals that have been difficult to classify correctly at earlier iterations. The combined classifier is equivalent to a weighted majority vote of the weak classifiers for shifted class labels $y_i \in \{-1, 1\}$,

$$\mathcal{C}^{(M)}(X) = \text{sign} \left(\sum_{m=1}^M \alpha_m f^{(m)}(X) \right).$$

Three elements need to be chosen: (i) the type of weak learners $f^{(m)}$, (ii) the reweighting of the data and the aggregation weights α_m , and (iii) the number of boosting iterations M . Regarding issue (i), we exclusively focus on decision trees, see [41]. These are the most popular learners in conjunction with boosting. In fact, we even further restrict here to *stumps*, which are trees with two terminal nodes only, since in the context of gene expression data, this always yielded better or equal performance than boosting larger trees. Concerning issue (ii), the reweighting of the data and the choice of aggregation weights can be coherently motivated by the principle of functional gradient descent [58, 16], from which several versions of boosting for classification emerge. We build here on the LogitBoost introduced in [16]: it relies on the binomial log-likelihood as a loss function, which is a more natural criterion in classification than the exponential criterion underlying the AdaBoost algorithm. Since the former increases

linearly instead of exponentially for strongly negative margins [59], it is more robust in noisy problems where the misclassification risk of equation (20) is substantial, and also in situations where mislabeled training data points or inhomogeneities in the training samples are present, all of which can be the case with gene expression data. Finally regarding (iii), the choice of the stopping parameter is often neglected and the boosting process is stopped at a usually large, but arbitrarily fixed number of iterations. Alternatively, we consider an empirical approach for the choice of M in the next section. The binary LogitBoost with decision stumps as weak learner works then as follows:

Step 1: Initialization

Start with an initial committee function $F^{(0)}(x) \equiv 0$ and initial probabilities $p^{(0)}(x) \equiv 1/2$; $p(x)$ is an abbreviation for $\widehat{\mathbb{P}}[Y = 1|X = x]$.

Step 2: LogitBoost iterations

For $m = 1, 2, \dots, M$ repeat:

A. Fitting the weak learner

- (i) Compute working response and weights for all $i = 1, \dots, n$

$$w_i^{(m)} = p^{(m-1)}(x_i) \cdot (1 - p^{(m-1)}(x_i)),$$

$$z_i^{(m)} = \frac{y_i - p^{(m-1)}(x_i)}{w_i^{(m)}}.$$

- (ii) Fit a regression stump $f^{(m)}$ by weighted least squares

$$f^{(m)} = \operatorname{argmin}_f \sum_{i=1}^n w_i^{(m)} (z_i^{(m)} - f(x_i))^2.$$

B. Updating and classifier output

$$F^{(m)}(x_i) = F^{(m-1)}(x_i) + \frac{1}{2} f^{(m)}(x_i).$$

$$C^{(m)}(x_i) = \operatorname{sign} \left(F^{(m)}(x_i) \right),$$

$$p^{(m)}(x_i) = \left(1 + \exp \left(-2 \cdot F^{(m)}(x_i) \right) \right)^{-1}.$$

To increase understanding of the LogitBoost algorithm, we point out that each committee function $F^{(m)}(x)$ is an estimate of half of the log-odds ratio

$$F(x) = \frac{1}{2} \log \left(\frac{p(x)}{1 - p(x)} \right).$$

LogitBoost thus fits an additive logistic regression model by stagewise optimization of the binomial log-likelihood. More details can be found in [16].

A very useful property of our classification method is that it directly yields probability estimates $\widehat{\mathbb{P}}[Y = j|X = x]$. This is crucial for constructing classifiers respecting non-equal misclassification costs. Moreover, it allows to build classifiers which have the option to assign the label “no class” (or “doubt”) for certain regions in the space of gene expression vectors x , see for example [42].

An important advantage of LogitBoost compared to methods like neural nets or support vector machines is that it works well without fine tuning and no sophisticated nonlinear optimization is necessary. Provided that a decision tree algorithm is available, e.g. versions of CART [41] or C4.5 [60], LogitBoost with trees can be implemented very easily. Software for decision trees is widely available: for example for free as an R-Package called `rpart`, at <http://www.stat.math.ethz.ch/CRAN>.

Choice of the Stopping Parameter

The stopping parameter M is often simply fixed at a large number in the range of dozens or hundreds. This, because boosting is generally quite resistant against overfitting so that the choice of M is typically not very critical, see also Figure 15. An alternative

is to use an empirical approach for estimation of M by leave-one-out cross validation on the training data. The idea is to compute the binomial log-likelihood

$$\begin{aligned} \ell(m) &= \sum_{i=1}^n \log \left(\hat{p}^{(m)}(x_i) \right) \cdot 1_{[Y_i=1]} + \\ &\quad + \log \left(1 - \hat{p}^{(m)}(x_i) \right) \cdot 1_{[Y_i=0]}, \end{aligned} \quad (22)$$

for each boosting iteration m across the samples and to choose the stopping parameter as the m for which $\ell(m)$ is maximal. We observed that $\ell(m)$ usually peaks somewhere between 10 and 100 boosting iterations. However empirically, we could not exploit significant advantages of estimated stopping parameters against a choice of $M = 100$ in the gene expression data we considered.

2.3 Reducing Multiclass to Binary

Here we explain how multi-response problems ($J > 2$) can be handled in conjunction with boosting. We recommend to compare each response class separately against all other classes. This *one-against-all* approach for reduction to J binary problems is very popular in the machine learning community, since many algorithms are solely designed for binary problems. It works by defining the response in the j th problem as

$$Y^{(j)} = \begin{cases} 1, & \text{if } Y = j, \\ 0, & \text{else} \end{cases}$$

and running j times the entire procedure including feature pre-selection, binary LogitBoost and stopping parameter estimation on the data $(x_1, y_1^{(j)}), \dots, (x_n, y_n^{(j)})$. This yields estimates $\hat{\mathbb{P}}[Y^{(j)} = 1|X]$ for $j \in \{0, \dots, J - 1\}$, which can be converted into probability estimates for $Y = j$ via normalization,

$$\hat{\mathbb{P}}[Y = j|X] = \frac{\hat{\mathbb{P}}[Y^{(j)} = 1|X]}{\sum_{k=1}^J \hat{\mathbb{P}}[Y^{(k)} = 1|X]}.$$

This expression can be plugged into the Bayes classifier of equation (21) and it is easy to see that this yields

$$\mathcal{C}(X) = \operatorname{argmax}_{j \in \{0, \dots, J-1\}} \hat{\mathbb{P}}[Y^{(j)} = 1|X]$$

as our final classifier in multiclass problems. More sophisticated and computationally more expensive approaches for reducing multiclass to binary problems also exist, see [30] or [31] for a thorough discussion.

The one-against-all approach allows for different preselected features, different chosen variables for the decision trees in the LogitBoost algorithm, and for different model complexity via different stopping parameters for every class discrimination. This adaption seems to be very important with gene expression data. We observed, that the multiclass LogitBoost of [16], which treats the multiclass problem more simultaneously, performed much worse in our study. In the NCI dataset, comprising $J = 8$ different tumor types, it yielded an error rate of 36.1%, whereas with the one-against-all method, the error-rate was only 22.9%. For the Lymphoma dataset with $J = 3$ response classes, the one-against-all approach is also superior with 1.61% versus 8.06%.

3. Results

3.1 Real Data

We explored the performance of our classification technique on six publicly available datasets.

Leukemia

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. More information can be found in [32]; the raw data are available at <http://www-genome.wi.mit.edu/cancer>. Following the protocol in [19], we preprocess them by thresholding, filtering, a log-

arithmetic transformation and standardization, so that the data finally comprise the expression values of $p = 3,571$ genes.

Colon

In this dataset, expression levels of 40 tumor and 22 normal colon tissues for 6,500 human genes are measured using the Affymetrix technology. A selection of 2,000 genes with highest minimal intensity across the samples has been made in [34], and these data are publicly available at <http://molbio.princeton.edu/oncology/>. As for the leukemia dataset, we process these data further by carrying out a base 10 logarithmic transformation and standardizing each tissue sample to zero mean and unit variance across the genes.

Estrogen & Nodal

These datasets were first presented in recent papers of West et al. [33] and Spang et al. [61]. Their common expression matrix monitors 7,129 genes in 49 breast tumor samples. The data were obtained by applying the Affymetrix gene chip technology and are available at http://mgm.duke.edu/genome/dna_micro/work/. We thresholded the raw data with a floor of 100 and a ceiling of 16,000 and then applied a base 10 logarithmic transformation. Finally, each experiment was standardized to zero mean and unit variance across the genes. Two different response variables are available: The first one describes the status of the estrogen receptor (ER). 25 samples are ER+, whereas the remaining 24 samples are ER-. The second response variable describes the lymph nodal (LN) status, which is an indicator for the metastatic spread of the tumor, a very important risk factor for disease outcome. Also here, 25 samples are positive (LN+) and 24 samples are negative (LN-).

Lymphoma

This dataset is publicly available at <http://llmpp.nih.gov/lymphoma/data/figure1> and contains gene expression levels of the $J = 3$ most prevalent adult lymphoid malignancies: 42 samples of diffuse large B-cell lymphoma, 9 observations of follicular lymphoma and 11 cases of chronic lymphocytic leukemia. The

total sample size is $n = 62$, and the expression of $p = 4,026$ well-measured genes, preferentially expressed in lymphoid cells or with known immunological or oncological importance is documented. More information on these data can be found in [37]. We imputed missing values and standardized the data as described in [19].

NCI

This dataset comprises gene expression levels of $p = 5,244$ genes for $n = 61$ human tumor cell lines from cDNA microarrays, which can be divided in $J = 8$ classes: 7 breast, 5 central nervous system, 7 colon, 6 leukemia, 8 melanoma, 9 non small cell lung carcinoma, 6 ovarian and 9 renal tumors. A more detailed description of the data can be found on the website <http://genome-www.stanford.edu/nci60> and in [39]. We work with preprocessed data as described in [19].

Empirical Study

We performed leave-one-out cross validation to explore the classification potential of our method. This means that we set aside the i th observation and carry out feature selection, stopping parameter estimation and classifier fitting by considering only the remaining $(n - 1)$ data points. We then predict \hat{Y}_i , the class label of the i th observation and repeat this process for all observations in the training sample. Each observation is held out and predicted exactly once. We determine the test set error using symmetrically equal misclassification costs

$$Error = \frac{1}{n} \sum_{i=1}^n 1_{[Y_i \neq \hat{Y}_i]}.$$

Tables 19–22 report test set errors with different gene subset size from feature selection for several classifiers. LogitBoost is reported with the optimal stopping time, yielding the minimal cross-validated error across the boosting iterations. This stopping time is not known in real life problems and results in an over-optimistic misclassification rate. Thus, also the error after a

fixed number of 100 iterations as well as the error using our stopping parameter estimate from equation (22) are given. A close competitor to LogitBoost is the discrete AdaBoost algorithm of [15]. We report its error rate after 100 iterations and observe that its accuracy is inferior to LogitBoost in 19 cases, equal in 11 cases and superior in 12 cases. LogitBoost thus seems to have an edge over AdaBoost, but this is far from being significant. To illustrate the benefit of boosting, we also ran the (optimally tuned) CART algorithm [41] to produce single classification trees. Boosting uses them as weak learners and leads to massive improvements in all except the estrogen and nodal datasets. As a benchmark method we applied the 1-nearest-neighbor classifier [40] with simultaneous classification in multiclass problems, using all the genes from the one-against-all approach in conjunction with boosting. This simple rule is known to perform reasonably well on gene expression data in connection with precedent feature selection. For the smaller gene subsets, it is better than boosting for the leukemia and lymphoma data, at about the same level for the colon and NCI data and worse than boosting for the estrogen and nodal data. With larger gene subsets, if many noise variables are present, its accuracy often deteriorates severely.

It is known that repeated random splitting of the data into training and larger test sets may yield more accurate estimates of the test set error than leave one out cross validation, but the former has the disadvantage of being difficult to reproduce. In our setting, the error rates from random splitting (data not shown) were often at a somewhat higher level, but the relationship between the classifiers remained unchanged.

The choice of the stopping parameter for boosting is not very critical in all six datasets. Our classifier did not overfit much and Figure 15 shows that the error-rates are at, or close to the minimal error-rate for many boosting iterations. We conjecture that stopping after a large, but arbitrary number of 100 iterations is a reasonable strategy in the context of gene expression data. Our data-driven approach for estimating the stopping parameters by cross validation on the training data does not improve and most

<i>Leukemia</i>	10	25	50	75
LogitBoost, optimal	4.17%	2.78%	4.17%	2.78%
LogitBoost, estimated	6.94%	5.56%	5.56%	4.17%
LogitBoost, 100 iter.	5.56%	2.78%	4.17%	2.78%
AdaBoost, 100 iter.	4.17%	4.17%	4.17%	4.17%
1-Nearest-Neighbor	4.17%	1.39%	4.17%	5.56%
Classification Tree	22.22%	22.22%	22.22%	22.22%
<i>Colon</i>	10	25	50	75
LogitBoost, optimal	14.52%	16.13%	16.13%	16.13%
LogitBoost, estimated	22.58%	19.35%	22.58%	20.97%
LogitBoost, 100 iter.	14.52%	22.58%	22.58%	19.35%
AdaBoost, 100 iter.	16.13%	24.19%	24.19%	17.74%
1-Nearest-Neighbor	17.74%	14.52%	14.52%	20.97%
Classification Tree	19.35%	22.58%	29.03%	32.26%
<i>Estrogen</i>	10	25	50	75
LogitBoost, optimal	4.08%	4.08%	2.04%	2.04%
LogitBoost, estimated	6.12%	6.12%	6.12%	6.12%
LogitBoost, 100 iter.	8.16%	6.12%	6.12%	4.08%
AdaBoost, 100 iter.	8.16%	8.16%	2.04%	2.04%
1-Nearest-Neighbor	4.08%	8.16%	18.37%	12.24%
Classification Tree	4.08%	4.08%	4.08%	4.08%

Table 19: Test set error rates based on leave one out cross validation for leukemia, colon, and estrogen data with gene subsets from feature selection ranging between 10 to 75 genes for several classifiers. LogitBoost error rates are reported with optimal stopping (minimum cross-validated error across iterations), after a fixed number of 100 iterations as well as with the estimated stopping parameter.

<i>Nodal</i>	10	25	50	75
LogitBoost, optimal	16.33%	18.37%	22.45%	22.45%
LogitBoost, estimated	22.45%	30.61%	30.61%	34.69%
LogitBoost, 100 iter.	18.37%	20.41%	26.53%	42.86%
AdaBoost, 100 iter.	18.37%	16.33%	28.57%	40.82%
1-Nearest-Neighbor	18.37%	30.61%	30.61%	42.86%
Classification Tree	22.45%	20.41%	20.41%	20.41%
<i>Lymphoma</i>	10	25	50	75
LogitBoost, optimal	1.61%	3.23%	1.61%	1.61%
LogitBoost, estimated	3.23%	3.23%	3.23%	1.61%
LogitBoost, 100 iter.	1.61%	3.23%	1.61%	1.61%
AdaBoost, 100 iter.	4.84%	3.23%	1.61%	1.61%
Nearest Neighbor	1.61%	0.00%	0.00%	0.00%
Classification Tree	22.58%	22.58%	22.58%	22.58%
<i>NCI</i>	10	25	50	75
LogitBoost, optimal	32.79%	31.15%	27.87%	22.95%
LogitBoost, estimated	36.07%	44.26%	36.07%	39.34%
LogitBoost, 100 iter.	37.70%	44.26%	34.43%	29.51%
AdaBoost, 100 iter.	50.82%	37.70%	34.43%	29.51%
Nearest Neighbor	36.07%	29.51%	27.87%	24.59%
Classification Tree	70.49%	68.85%	65.57%	65.57%

Table 20: Test set error rates based on leave one out cross validation for nodal, lymphoma and NCI data with gene subsets from feature selection ranging between 10 to 75 for several classifiers. LogitBoost error rates are reported with optimal stopping (minimum cross-validated error across iterations), after a fixed number of 100 iterations as well as with the estimated stopping parameter.

<i>Leukemia</i>	100	200	3571
LogitBoost, optimal	2.78%	2.78%	2.78%
LogitBoost, estimated	4.17%	5.56%	5.56%
LogitBoost, 100 iter.	2.78%	2.78%	2.78%
AdaBoost, 100 iter.	4.17%	2.78%	4.17%
1-Nearest-Neighbor	4.17%	2.78%	1.39%
Classification Tree	22.22%	22.22%	23.61%
<i>Colon</i>	100	200	2000
LogitBoost, optimal	16.13%	14.52%	12.90%
LogitBoost, estimated	22.58%	19.35%	19.35%
LogitBoost, 100 iter.	17.74%	16.13%	16.13%
AdaBoost, 100 iter.	20.97%	17.74%	17.74%
1-Nearest-Neighbor	19.35%	17.74%	25.81%
Classification Tree	27.42%	14.52%	16.13%
<i>Estrogen</i>	100	200	7129
LogitBoost, optimal	2.04%	4.08%	2.04%
LogitBoost, estimated	6.12%	6.12%	6.12%
LogitBoost, 100 iter.	4.08%	8.16%	6.12%
AdaBoost, 100 iter.	6.12%	4.08%	4.08%
1-Nearest-Neighbor	14.29%	14.29%	16.33%
Classification Tree	4.08%	4.08%	4.08%

Table 21: Test set error rates based on leave one out cross validation for leukemia, colon and estrogen data with gene subsets from feature selection ranging between 100 to all genes for several classifiers. LogitBoost error rates are reported with optimal stopping (minimum cross-validated error across iterations), after a fixed number of 100 iterations as well as with the estimated stopping parameter.

<i>Nodal</i>	100	200	7129
LogitBoost, optimal	22.45%	18.37%	20.41%
LogitBoost, estimated	28.57%	26.53%	24.49%
LogitBoost, 100 iter.	42.86%	18.37%	22.45%
AdaBoost, 100 iter.	36.73%	22.45%	28.57%
1-Nearest-Neighbor	36.73%	36.73%	48.98%
Classification Tree	20.41%	20.41%	20.41%
<i>Lymphoma</i>	100	200	4026
LogitBoost, optimal	1.61%	3.23%	8.06%
LogitBoost, estimated	3.23%	3.23%	-%
LogitBoost, 100 iter.	1.61%	3.23%	8.06%
AdaBoost, 100 iter.	1.61%	1.61%	3.23%
Nearest Neighbor	0.00%	1.61%	1.61%
Classification Tree	22.58%	22.58%	25.81%
<i>NCI</i>	100	200	5244
LogitBoost, optimal	26.23%	24.59%	31.15%
LogitBoost, estimated	44.26%	47.54%	-%
LogitBoost, 100 iter.	26.23%	24.59%	36.07%
AdaBoost, 100 iter.	32.79%	29.51%	36.07%
Nearest Neighbor	22.95%	22.95%	27.87%
Classification Tree	60.66%	62.30%	62.30%

Table 22: Test set error rates based on leave one out cross validation for nodal, lymphoma and NCI data with gene subsets from feature selection ranging between 100 to all genes for several classifiers. LogitBoost error rates are reported with optimal stopping (minimum cross-validated error across iterations), after a fixed number of 100 iterations as well as with the estimated stopping parameter. The cross validation with estimated stopping parameters for the lymphoma and NCI data with all genes was not feasible.

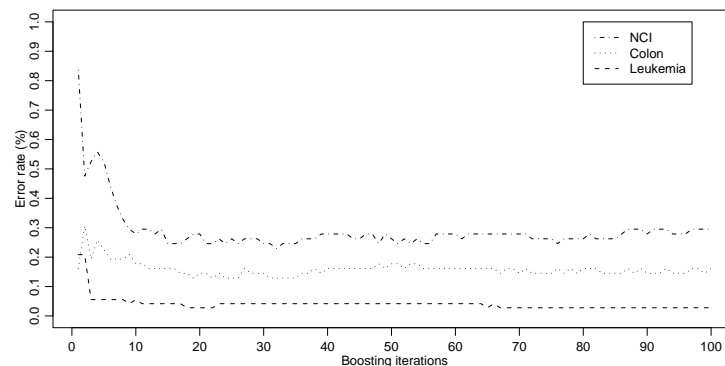


Figure 15: Test set error curves for leukemia, colon and NCI data. The number of genes was chosen such that the performance was optimal: 75 for leukemia and NCI data, and 2,000 for the colon data. The error curves for estrogen, nodal and lymphoma data look similar and are not displayed for reasons of clarity.

often yields slightly worse results, probably due to additional random variation.

ROC curves

In our evaluation, we determined the test set error using symmetrically equal misclassification costs. In a clinical setting, one often prefers to punish misclassifications asymmetrically, since false negative errors, i.e. classifying a tumorous tissue as normal can be fatal, whereas false positive errors, i.e. predicting a normal tissue as a tumor may be less serious since in this case additional tests will be carried out.

ROC curves illustrate how accurate classifiers are under asymmetric losses, by plotting the tradeoff between false positives and false negatives. Each point on the two-dimensional ROC curve corresponds to a particular probability $\beta \in [0, 1]$ that was used as a threshold for positive (tumorous) classification. The (x, y)

coordinates of each point are then the fractions of negative and positive samples that are classified as positive with this particular threshold β . In the ideal case, the ROC curve goes through $(0, 1)$, the upper left corner of the plot.

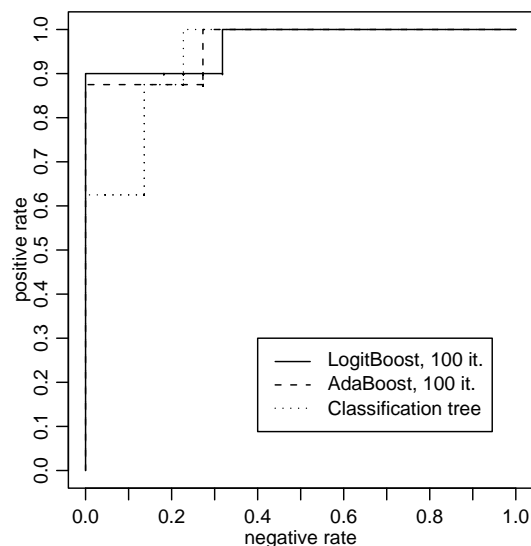


Figure 16: ROC curves for LogitBoost, AdaBoost and classification trees applied on colon data without feature preselection. On the x-axis is the fraction of negative examples classified as positives (tumorous), the y-axis show the fraction of positive examples classified as positives. Each point on the curves represents the fractions achieved with a particular probability $\beta \in [0, 1]$ as threshold for positive classification. The probabilities for class membership were estimated by leave one out cross validation.

Figure 16 shows the ROC curves for LogitBoost after 100 iterations, AdaBoost after 100 iterations and classification trees applied to the colon data with $\tilde{p} = 2,000$ predictor variables. The class membership probabilities for each sample were determined

by leave one out cross validation. We can see that both boosting classifiers yield a similar curve which comes closer to the ideal ROC curve than the one from classification trees. Note that this is a case where the test set errors under equal misclassification losses are very similar. However for this example, Boosting has an advantage for small negative rates.

Validation of the Results

The leukemia dataset has been considered by many authors. On the original test set comprising 34 observations, LogitBoost assigns the correct label to 33 of the 34 patients. This can be directly compared to the study in [32], where 29 observations were classified correctly by their weighted voting scheme. [43], working with support vector machines, report results ranging between 30 to 32 correct classifications. [18] applied AdaBoost and carried out cross validation. After 10,000 boosting iterations, they obtained 2.78% misclassified and 1.39% unclassified instances without feature preselection, and 1.39% either mis- or unclassified instances with several gene subsets.

The colon dataset has been cross validated by [18] with various classifiers, with and without precedent feature selection. AdaBoost performed comparably bad and the best result they report are 17.74% of misclassified, plus another 9.68% of unclassified instances. Our best result here are in the range between 12.90% and 14.52% wrongly classified observations. We gain evidence that LogitBoost can be superior over AdaBoost in some cases. The best support vector machine of [43] misclassified only 6 tissue samples in the full cross validation cycle, being equivalent to an error-rate of 9.68%, whereas our error-rate of 12.9% corresponds to 8 misclassifications.

The NCI dataset has been extensively analyzed by [19]. They tried several classification methods including AdaBoost on a precedently reduced feature space. Also in their study, AdaBoosting was not among the best classifiers with a median error of about 48% in 150 random divisions in training and test set. Our method with reduction to binary problems and LogitBoost shows

a considerable improvement to an error of only 22.9%, but a part of this reduction could be caused by the two different setups, i.e. random divisions versus cross validation for estimating the test set error.

For the estrogen and nodal datasets, we obtain better predictions than [33] with their Bayesian approach, even without omitting the most difficult cases as they do. A validation of the results for the lymphoma dataset in comparison to other studies is not possible. Since our classifier does well with respect to the benchmarks, we expect that it yields competitive results here too.

3.2 Simulation

Due to the scarcity of samples in real datasets, relevant differences between classification methods may be difficult to detect. We consider here simulated gene expression data: by generating large test sets, the performance of our modified LogitBoost classifier can be much more accurately compared against the benchmark classifiers and assessing significant differences becomes possible. We start by producing gene expression profiles from a multivariate normal distribution, $X \sim \mathcal{N}_p(0, \Sigma)$, where the covariance structure Σ is from the colon dataset. This reflects the real situation with microarray data, yielding gene expression profiles with $p = 2,000$ genes. We continue by assigning one out of two response classes to the simulated expression profiles according to $Y | X = x \sim \text{Bernoulli}(p(x))$, where the conditional probabilities are from the model

$$\log\left(\frac{p(x)}{1-p(x)}\right) = \sum_{j=1}^{10} \beta_j \bar{x}^{(C_j)} \sum_{j=1}^{10} \left(1 + \gamma_j \bar{x}^{(C_j)}\right) \sum_{j=1}^{10} \left(1 + \delta_j \bar{x}^{(C_j)}\right)$$

The $\bar{x}^{(C_j)} = \sum_{g \in C_j} x^{(g)} / |C_j|$ are mean values across random gene clusters $C_j \subseteq \{1, \dots, p\}$ of uniformly random size between 1 and 10 genes, the expected number of relevant genes is thus $10 \cdot 5.5 = 55$. The model coefficients β_j, γ_j and δ_j were randomly drawn from normal distributions with zero mean and standard deviation $\sigma = 2, 1$ and $1/2$, respectively. This leads to a complex

non-additive decision boundary, where LogitBoost with stumps, which fits an additive model, is not in favor of the benchmark classifiers¹.

The training sample size was chosen to be $n = 200$ and we considered the performance of the classifiers on single but large test sets comprising 1,000 new observations. The process was independently repeated 20 times, which enables to explore whether LogitBoost yields significantly better test set error-rates than the benchmark classifiers by performing paired Wilcoxon signed rank tests for the hypothesis of equal misclassifications against the two-sided alternative. The test always points towards better accuracy of LogitBoost, results are given in Table .

	1-Nearest-Neighbor
LogitBoost, optimal	12.37%, $p = 1.7 \cdot 10^{-4}$
LogitBoost, 150 iter.	7.54%, $p = 1.4 \cdot 10^{-3}$
	Classification Tree
LogitBoost, optimal	10.21%, $p = 1.1 \cdot 10^{-3}$
LogitBoost, 150 iter.	5.27%, $p = 1.7 \cdot 10^{-2}$

Table 23: *Percentual improvement and p-values of LogitBoost (stopped optimally and after a fixed number of 150 iterations) against the generic 1-nearest-neighbor method and classification trees in 20 independent realizations from our simulation model. The p-values are from paired two-sided Wilcoxon signed rank tests for equal test set error and are always in favor of LogitBoost.*

Not only when the LogitBoost algorithm was optimally stopped, but also after a fixed number of 150 iterations (which was found to be a reasonable ad-hoc choice for this problem) it significantly outperformed the benchmark methods. This confirms our findings from real data that our classifier is more accurate than the benchmarks.

¹LogitBoost with larger trees would allow to pick up non-additive decision boundaries.

4. Conclusions

We propose modifications and extensions of boosting classifiers for microarray gene expression data from several tissue or cancer types. We applied precedent feature selection and used the more robust LogitBoost combined with an alternative approach for binary problems. The results on six real and a simulated datasets indicate that these modifications are successful and make boosting a competitive player for predicting expression data. Our feature preselection generally improved the predictive power of a classifier. Moreover, we observed slightly better performance of LogitBoost over AdaBoost, and our whole procedure (feature selection plus LogitBoost) compares favorably with previously published results using AdaBoost. Finally, we propose to reduce multiclass problems to multiple binary problems which are solved separately. This was found to have a great potential for more accurate results on gene expression data, where the choice of predictor variables is crucial.

Our LogitBoost classifier is very suitable for application in a clinical setting. In comparison to other methods, it yields good results, is easy to implement and does not require sophisticated tuning and model or kernel selection as with neural networks or support vector machines. Unlike several other classifiers, it directly provides class membership probabilities. They are essential to quantify the uncertainty of a class label assignment and allow decisions under unequal misclassification costs which are often encountered in practice.

BagBoosting for Tumor Classification

Marcel Dettling
ETH Zürich

Abstract

Motivation: Microarray experiments are expected to contribute significantly to progress in cancer treatment by enabling a precise and early diagnosis. They create a need for class prediction tools that can deal with a large number of highly correlated input variables, perform feature selection, and provide class probability estimates that serve as a quantification of the predictive uncertainty. A very promising solution is to combine the two ensemble schemes bagging and boosting to a novel algorithm called BagBoosting.

Results: When bagging is used as a module in boosting, the resulting classifier consistently improves the predictive performance and the probability estimates of both bagging and boosting on real and simulated gene expression data. This quasi-guaranteed improvement can be obtained by simply making a bigger computing effort. The advantageous predictive potential is also confirmed by comparing BagBoosting to several established class prediction tools for microarray data.

Availability: Software for the modified boosting algorithms, for benchmark studies and for the simulation of microarray data are available as an *R* package under GNU public license at <http://stat.ethz.ch/~dettling/bagboost.html>

1. Introduction

A precise diagnosis of cancerous malignancies is difficult but often crucial for successful treatment. Given the large-scale, high-throughput gene expression technology and accurate statistical methods, bio-molecular information could become as, or even more important for cancer diagnosis than traditional clinical factors. The challenge is that microarray experiments generate large datasets with expression values for thousands of genes, but usually not more than a few dozens of arrays. The situation with so many more predictor variables than experiments rises new statistical challenges and has led to a wealth of research. The task of diagnosing cancer on the basis of microarray data has been termed *class prediction* in the literature, and encompasses methods ranging from modified versions of traditional discriminant analysis, over penalized regression approaches, classical nonparametric methods such as the nearest neighbor rule to modern tools of machine learning. See for example Dudoit and Fridlyand [62] for an overview and references.

Boosting is a flexible class prediction tool from machine learning with remarkable success in a wide variety of applications, especially in those with high-dimensional predictors. It aims at producing an accurate committee classifier from a sequential ensemble of *base learners*, which are fitted to adaptively reweighted versions of the data. It is attractive to use boosting for class prediction with microarray data due to its natural ability to perform multivariate feature selection, and because it provides class probability estimates which serve as a natural quantification of the predictive uncertainty. This has first been tried by Ben-Dor et al. [18] and Dudoit et al. [19] with moderate success: empirically, boosting could at best keep up with much simpler methods such as the nearest neighbor rule. Later we suggested a boosting algorithm that was tailored for microarray data, showing a more satisfactory predictive performance, see Dettling and Bühlmann [17].

Here, we present a novel type of boosting algorithm and show its promising potential to improve class prediction with microarray

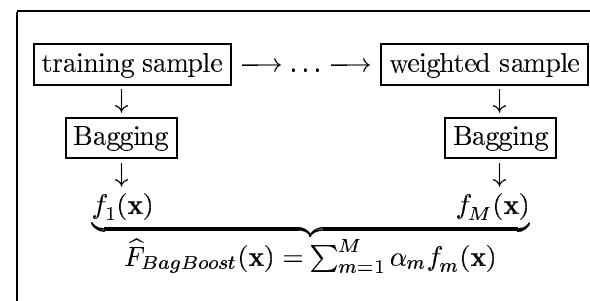


Figure 17: *The fundamental idea of BagBoosting*

data. Our algorithm is called *BagBoosting*, because it uses bagging as a module in our tailored boosting algorithm for microarray data. The idea is illustrated in Figure 17: in each boosting iteration, we do not just rely on a single base learner, but instead aggregate the output from several base predictors generated from bootstrap samples, each drawn with replacement from the reweighted training data. The rationale for combining the two ensemble schemes bagging and boosting is as follows. While the boosting committee has clearly lower bias but slightly increased variance than the base learner, bagging of (unstable) base learners leads to an ensemble with lower variance but approximately non-altered bias. Hence BagBoosting might combine the advantages of both methods and results in a prediction tool achieving both lower bias and variance, i.e. lower mean squared error. Even though we cannot present a strict mathematical justification that applies for the microarray setting, simulation studies clearly reflect these heuristically derived advantages. As a consequence, we also expect BagBoosting to yield superior class prediction results on real microarray data. An elaborate empirical study confirms the improvement compared to both bagging and boosting. Also with respect to established classifiers including discriminant analysis, nearest neighbor methods and modern tools such as support vector machines and random forests, BagBoosting is competitive.

Finally, we show how the BagBoosting fit can be rewritten in terms of an additive model. This serves to study the influence of single genes on real world classification problems, and to analyze the ability to recover the true model structure in simulated data.

2. Methods

2.1 Class Prediction

The main goal in class prediction with gene expression data is a precise and early diagnosis of cancerous malignancies that allows to tailor the patients' treatment for maximal efficacy and minimal toxicity. Given microarray experiments and information about the disease outcome from n former patients, the task of class prediction amounts to learning the relation between the transcript levels and the outcome. Then, presented with the gene expression profiles of new, independent patients, we can establish a diagnosis of their disease development and outcome. In mathematical notion, we assume that we are given a learning sample \mathcal{L} of n training data pairs

$$\mathcal{L} = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\},$$

which are independent and identically distributed (*iid*) realizations of a random vector (\mathbf{x}, y) . The feature vector $\mathbf{x} \in \mathbb{R}^p$ is the gene expression profile, which can be either from cDNA microarrays or from Affymetrix oligonucleotide chips, but we assume that it is accurately preprocessed and normalized, such that it can be taken at face-value. In the simplest form, the response variable $y \in \{0, 1\}$ codes for a dichotomous response², as for example tumor subtype or risk category. For class prediction, we choose the approach of estimating conditional class probabilities

$$\hat{p}(\mathbf{x}) = \hat{\mathbb{P}}_{\mathcal{L}}[y = 1 | \mathbf{x}]$$

from the learning sample \mathcal{L} , based on the gene expression profile \mathbf{x} . They are a natural quantification for the uncertainty of class

²Extensions to a polytomous response are discussed later in section

predictions and can in turn be used to predict class labels. In the case of equal misclassification costs, a new patient with gene expression profile \mathbf{x}_ν is assigned to one of the response classes via

$$\hat{y}(\mathbf{x}_\nu) = \begin{cases} 0, & \text{if } \hat{p}(\mathbf{x}_\nu) < 1/2 \\ 1, & \text{if } \hat{p}(\mathbf{x}_\nu) \geq 1/2. \end{cases} \quad (23)$$

Estimating conditional probabilities and subsequent class prediction is a thoroughly discussed problem in statistics, but microarray data with thousands of predictor variables p and just dozens of samples n are a new challenge which requires adaption of known and development of novel methodology. A promising tool performing multivariate variable selection, providing probability estimates and having good predictive potential in such high-dimensional problems are modified boosting algorithms.

2.2 BagBoosting

Boosting, first proposed by Freund and Schapire [15], is a powerful ensemble method that consistently estimates conditional class probabilities $\hat{p}(\mathbf{x})$ from a sequence of *base classifiers* which are fitted to iteratively reweighted versions of the training data. The initial notion of boosting was that in each iteration, the cases that were misclassified in the previous round get their weights increased, whereas the weights are decreased for cases that were correctly classified. Rather than as a sequential data reweighting scheme, boosting can more fruitfully be seen as a forward stagewise strategy for function estimation. It works by iterative optimization of an empirical risk function

$$R(\mathcal{L}, \hat{p}(\mathbf{x}), L) = \frac{1}{n} \sum_{i=1}^n L(y_i, \hat{p}(\mathbf{x}_i))$$

from the learning set \mathcal{L} via constrained functional gradient descent, where $\hat{F}_M(\cdot)$ denotes the current function estimate and $L(\cdot, \cdot)$ a statistically motivated loss function. If we employ the binomial log-likelihood

$$L(y, \hat{p}(\mathbf{x})) = y \cdot \log(\hat{p}(\mathbf{x})) + (1 - y) \cdot (1 - \hat{p}(\mathbf{x})),$$

a continuous surrogate for the 0/1-misclassification loss and a very established criterion for binary classification, it is easy to show that the resulting algorithm, called LogitBoost [16], yields an approximation to half of the log-odds ratio. That is,

$$\widehat{F}_M(\mathbf{x}) = \sum_{m=1}^M \alpha_m f_m(\mathbf{x}) \approx \frac{1}{2} \log \left(\frac{p(\mathbf{x})}{1-p(\mathbf{x})} \right).$$

Hence, LogitBoost is a linear expansion in a set of base learners f_m on the logit scale, obtained by stagewise optimization of the binomial log-likelihood. Estimated conditional class probabilities are obtained by the simple transformation

$$\widehat{p}(\mathbf{x}) = \frac{1}{1 + \exp(-2\widehat{F}_M(\mathbf{x}))},$$

and can be used for class prediction as in (23). Obviously, the choice of the base learner f_m is crucial for the final boosting estimate \widehat{F}_M . Originally, mainly decision trees have been used as a base procedure. In the context of microarray data, we observed that *stumps*, these are the simplest decision trees with only one split and two terminal nodes, yield the best empirical performance [17].

Now, we present a novel type of algorithm which uses Breimans bagging [63] as a module in our tailored boosting procedure for microarray data. This amounts to a modification of boostings base learner: instead of relying on a single decision tree/stump, we aggregate multiple versions of them, obtained from bootstrap samples which are drawn with replacement from the reweighted training data of the m th boosting iteration, see Figure 17. This intuitive idea has been thought of by different researchers, it was briefly sketched in [64], but there are neither any publications containing a formal description of the algorithm, nor are there any systematical analyses of its performance via empirical studies on real or simulated data.

Algorithmic details

Our BagBoosting algorithm for class prediction with microarray

data is the LogitBoost algorithm using bagged stumps as base learner. It is defined as follows.

Step 1: Initialization

Set the initial boosting estimate to $\widehat{F}_0(\mathbf{x}_i) \equiv 0$ and the initial probabilities to $\widehat{p}(\mathbf{x}_i) \equiv 1/2$.

Step 2: Boosting iterations

For $m \in \{1, 2, \dots, M = 100\}$ repeat:

- A. Compute the weights w_i and the working response z_i for $i = 1, \dots, n$

$$\begin{aligned} w_i &\leftarrow \widehat{p}(\mathbf{x}_i) \cdot (1 - \widehat{p}(\mathbf{x}_i)), \\ z_i &\leftarrow \frac{y_i - \widehat{p}(\mathbf{x}_i)}{w_i}. \end{aligned}$$

- B. Bagging to obtain the base learner

For $b \in \{1, 2, \dots, B = 50\}$ repeat:

- (i) By randomly drawing n triples with replacement from the original data triples, construct the bootstrap sample $\mathcal{B} = \{(\mathbf{x}_1^*, w_1^*, z_1^*), \dots, (\mathbf{x}_n^*, w_n^*, z_n^*)\}$.
(ii) Fit a regression stump $g^{(b)}$ by weighted least squares on the bootstrap sample \mathcal{B}

$$g^{(b)}(\cdot) = \arg \min_{g(\cdot)} \sum_{i=1}^n w_i^* (z_i^* - g(\mathbf{x}_i^*))^2$$

Average these B stumps to obtain boostings base learner

$$f_m(\mathbf{x}_i) \leftarrow \frac{1}{B} \sum_{b=1}^B g^{(b)}(\mathbf{x}_i), \quad \text{for all } i = 1, \dots, n$$

- C. Updating boosting estimate and probabilities

$$\begin{aligned} \widehat{F}_m(\mathbf{x}_i) &\leftarrow \widehat{F}_{m-1}(\mathbf{x}_i) + \frac{1}{2} f_m(\mathbf{x}_i), \\ \widehat{p}(\mathbf{x}_i) &\leftarrow \frac{1}{1 + \exp(-2\widehat{F}_M(\mathbf{x}_i))}. \end{aligned}$$

The definition of weights w_i and working response z_i in the Logit(Bag)Boost algorithm is such that the (bagged) base learner is forced to focus on observations close to the decision boundary, i.e. data points where the boosting classifier is in doubt about class membership. The final number of boosting iterations M regulates the complexity of the prediction model, early stopping is a form of shrinkage. In the context of microarray data, we recommend a default value of $M = 100$, which is a reasonable compromise between computing time, predictive accuracy and prevention of overfitting. This choice was shown to be empirically superior to approaches where M was estimated on the training data via cross validation [17]. In contrast, the number of bagging iterations B is not a tuning parameter: in theory, the bagged estimator corresponds to the bootstrap expectation and would require infinitely many iterations. In practice, we employ Monte Carlo methods for an approximation; our choice of $B = 50$ has been recognized as sufficiently large [63].

Some Heuristics About BagBoosting

If the underlying base algorithm is a decision stump, this is a univariate indicator function with one split point in a single variable x_j and constant values in the two terminal nodes, BagBoosting yields a model which is additive in the predictor variables. This, since in every boosting iteration an average of B univariate functions is linearly added to the current fit. We can always rearrange the summation and represent the BagBoosting estimate as an additive combination

$$\hat{F}_M(\mathbf{x}) = \sum_{j=1}^p \theta_j h(x_j), \quad (24)$$

where $h(\cdot)$ are aggregated indicator functions typically showing a much smoother behavior than stumps, see the section on model recovery on page 142 and Figure 20. The coefficient θ_j is determined by when, how often and with which accuracy the j th variable was used during the bagging and boosting iterations; it reflects gene j 's importance in the final BagBoosting committee.

One of BagBoostings strengths is that it performs multivariate variable selection and adds complexity in a stagewise fashion: some of the genes are never selected at all and have $\theta_j \equiv 0$. The boosting iterations, i.e. the incorporation of new terms in (24) is always conditional on the current fit, such that we select genes that provide supplementary information to the previous ones, rather than picking genes that all re-explain the same phenomenon as with univariate gene selection. The relatively simple, additive model from (24) is focusing on the main effects, but free of interaction terms. This does not mean that we deny them in the “true” model, but from an empirical viewpoint (data not shown), probably due to the usually small sample size n , it pays off to rely on this simpler auxiliary model. As microarray studies get larger, more complicated models may become appropriate. The big advantage of BagBoosting is that we can obtain them, without alterations on the generic algorithm itself, by just using larger decision trees as the base algorithm.

It is well known that stumps are an unstable weak learner, producing highly biased and variable estimates. Here, we give some heuristical arguments why BagBoosting, an additive expansion in the set of stumps, improves these poor properties. Bühlmann and Yu show that squared error loss boosting with smoothing splines as base learner leads to an ensemble that has strongly reduced bias, but only slightly higher variance than the base learner [65]. Although the mathematical results cannot be directly transferred, the pronounced bias reduction and the weak increase of the variability presumably hold as well when applying LogitBoost with stumps to microarray data. Thus, there is room to improve boosting with a low variance base learner, which however still needs to have weak learning capacity only, i.e. a considerable amount of bias. Bagging, a smoothing operation that reduces the variance of unstable prediction tools by averaging out hard decisions as from indicator functions, but without having much of an effect on their bias, is predestinated to be used as base learner in boosting. The rationale is that BagBoosting profits from the synergy of baggings variance and boostings

bias reduction and achieves lower mean squared error, such that we can expect a better predictive performance. These heuristics are supported by our empirical work on real and simulated data, shown in the results section.

Comparison to Other Modifications of Boosting

Here, we emphasize that BagBoosting differs from other boosting bagging hybrids that have been proposed in the literature. In his stochastic gradient boost [66], Friedman draws a single, random subset of the data points for each boosting iteration and fits a single decision tree learner. The subsampling even increases the variability of the base learner, but Friedman argues that it reduces the correlation among the learners from different stages, which results in a variance reduction of the final boosting committee. He demonstrates superior empirical results on a few simulated regression and (to a much lesser extent) classification problems. Another procedure sharing similarities with BagBoosting is Breimans “Iterated Bagging to Debias Regressions” [67]. As the heading suggests, it is a procedure primarily developed for the regression, but not the classification context. It is closely related to squared error loss boosting and works by stagewise fitting of unbiasedly estimated residuals from the out-of-bag samples in a bagged base learner. Besides our common goal of simultaneously reducing bias and variance by combining stagewise modeling with bagged learners, our procedures are fundamentally different, since we are boosting small bagged decision trees in a classification problem, without making use of the out-of-bag samples. Finally, Friedman and Popescu [68] view ensemble learning from the perspective of high-dimensional numerical quadrature. They reveal a connection between sequential ensemble classification and quasi Monte Carlo integration, and they empirically show that hybrid approaches yield computational advantages. None of their hybrids exactly corresponds to our BagBoosting algorithm, but their methodology could serve as a route for explaining BagBoosting’s success.

2.3 Multiclass Problems

Since there are usually no genes that accurately discriminate more than two classes at once, we recommend to run multiple binary comparisons in J -class microarray problems where $y \in \{0, 1, \dots, J-1\}$. The simplest solution is the *one-against-all* approach, which works by defining the response in the j th problem as $y^{(j)} = 1$ if $y = j$, and $y^{(j)} = 0$ else. Then, we are running BagBoosting J times on the modified data $\mathcal{L}^{(j)} = \{(\mathbf{x}_1, y_1^{(j)}), \dots, (\mathbf{x}_n, y_n^{(j)})\}$. The estimated conditional class probabilities are normalized and can in turn be used for maximum likelihood classification via

$$\hat{p}^{(j)}(\mathbf{x}) = \frac{\hat{\mathbb{P}}_{\mathcal{L}^{(j)}}[y^{(j)} = 1|\mathbf{x}]}{\sum_{k=0}^{J-1} \hat{\mathbb{P}}_{\mathcal{L}^{(k)}}[y^{(k)} = 1|\mathbf{x}]}$$

$$\hat{y}(\mathbf{x}) = \arg \max_{j \in \{0, \dots, J-1\}} \hat{p}^{(j)}(\mathbf{x})$$

In [17] we have shown that this is empirically superior to boosting algorithms where multiclass problems are handled more simultaneously. Depending on the structure of the response classes, more complex schemes than *one-against-all* may be more accurate for splitting polytomous into multiple binary problems.

2.4 Feature Preselection

BagBoosting incorporates multivariate feature selection and hence does not crucially depend on preliminary gene filtering by univariate methods as many other class prediction tools. When running our analyses for several different numbers (10, 25, 50, 75, 100, 200) of genes filtered by the Wilcoxon test statistic, we observed that the error rates as well as the ranking among the classifiers did hardly change. Moreover, the predictive potential of BagBoosting was only slightly worse *without* gene preselection, whereas many benchmark classifiers deteriorated severely. For a fair comparison and due to space constraints,

we just display the outcome with 200 genes in the results section. The complete information is available from the webpage <http://stat.ethz.ch/~dettling/bagboost.html>.

Numerous alternative methods for gene filtering do exist. These include for example the popular t -test statistic, the TNoM-score [18] or a selection based on the Gini-index which is used as the splitting criterion for the stumps. However, we prefer the Wilcoxon statistic due to its theoretical property of being close to optimal over a wide range of data generating distributions, and due to our empirical evidence that Boosting performed worse with features preselected by the t -test and the TNoM-score, see [17].

2.5 Other Classifiers

We compared BagBoosting to several competitors, using the implementations that are accessible from the statistical software bundle R [69]. The classifiers include: 1) Boosting: 100 iterations of LogitBoost with stumps, exactly as described in [17]. 2) Random Forests [70]: a technique based on an ensemble of bagged trees that use random feature selection at each node. We relied on the R function `randomForest()` [71] and tuned the number of candidate variables for each split, as well as the minimum size of terminal nodes by searching the grid $\{1, 2, \dots, 8\} \times \{1, 2, \dots, 10\}$ on the training data, as suggested by Meyer et al. [72]. 3) Support Vector Machines [73]: a modern machine learning technique that fits hyperplanes with maximum margins to appropriately transformed data. We used the R implementation `svm()` [74] that is based on the LIBSVM C++ library of Chang and Lin [75], and performed C -classification with RBF kernels. The parameter γ and the cost C were tuned by a grid search on $\{2^{-10}, 2^{-9}, \dots, 2^5\} \times \{2^{-5}, 2^{-4}, \dots, 2^{10}\}$ by 10-fold cross validation on each training dataset, similar to Meyer et al. in [72]. 4) Nearest Shrunken Centroids (also known as PAM, [76]): this classifier is similar to diagonal linear discriminant analysis, but uses a soft-thresholding scheme to obtain sparse prediction models. We employed the R implementation of the original authors, which we also used for the determination of the shrinkage parameter

by 10-fold cross validation as suggested in the original publication [76]. As benchmark methods, we employ 5) Diagonal linear discriminant analysis (DLDA) and 6) the 1-nearest neighbor rule (NNR), relying on Euclidean distances.

3. Results

3.1 Real Data

We report the class prediction performance of BagBoosting on six publicly available datasets. These are:

<i>Dataset</i>	<i>Publication</i>	<i>n</i>	<i>p</i>	<i>J</i>
Leukemia	Golub (1999)	72	3,571	2
Colon	Alon (1999)	62	2,000	2
Prostate	Singh (2002)	102	6,033	2
Lymphoma	Alizadeh (2000)	62	4,026	3
SRBCT	Khan (2001)	63	2,308	4
Brain	Pomeroy (2002)	42	5,597	5

After preprocessing, all gene expression profiles were base 10 log-transformed and, in order to prevent single arrays from dominating the analysis, standardized to zero mean and unit variance. In the absence of genuine test sets for four of the six datasets, we performed our benchmark study by repeated random splitting into learning and test sets exactly as in [19]. The data were partitioned into a balanced learning set \mathcal{L} comprising two thirds of the arrays, used for feature preselection, tuning and fitting the classifiers. Then, the class labels of the remaining third of the experiments were predicted and the misclassification error was computed as the fraction of predicted class labels that differed from the true one. To reduce the variability, the splitting into learning and test sets was repeated 50 times and the error estimates were averaged. It is important to note that these results are honest in the sense that all gene filtering, classifier tuning and fitting operations were re-done on each of the 50 learning sets to

allow for reliable conclusions and to avoid over-optimistic results with downward bias.

	Leuke	Colon	Prost
BagBoost	4.08%	16.10%	7.53%
Boosting	5.67%	19.14%	8.71%
RanFor	1.92%	14.86%	9.00%
SVM	1.83%	15.05%	7.88%
PAM	3.75%	11.90%	16.53%
DLDA	2.92%	12.86%	14.18%
kNN	3.83%	16.38%	10.59%
	Lymph	SRBCT	Brain
BagBoost	1.62%	1.24%	23.86%
Boosting	6.29%	6.19%	27.57%
RanFor	1.24%	3.71%	33.71%
SVM	1.62%	2.00%	28.29%
PAM	5.33%	2.10%	25.29%
DLDA	2.19%	2.19%	28.57%
kNN	1.52%	1.43%	29.71%

Table 24: Misclassification rates for 7 classifiers on 6 microarray datasets, based on 50 random partitions into learning sets ($\frac{2}{3}$ of the data) and test sets ($\frac{1}{3}$ of the data).

In Table 24, we report the misclassification rates of the classifiers over the six datasets. Figure 18 contains a visual illustration of these results, following the suggestions about the presentation of benchmark studies by Hothorn et al. [77]. The left panels show boxplots, where the median is highlighted in red. The right panels show density curves, where the vertical red line corresponds to the mean error rate. The performance of the classifiers varies across the different datasets, but in summary, BagBoosting, support vector machines and random forests seem to have an edge. The nearest shrunken centroid classifier (PAM), as well as the simple benchmarks NNR and DLDA do surprisingly well and can almost keep up except on the prostate data, notably

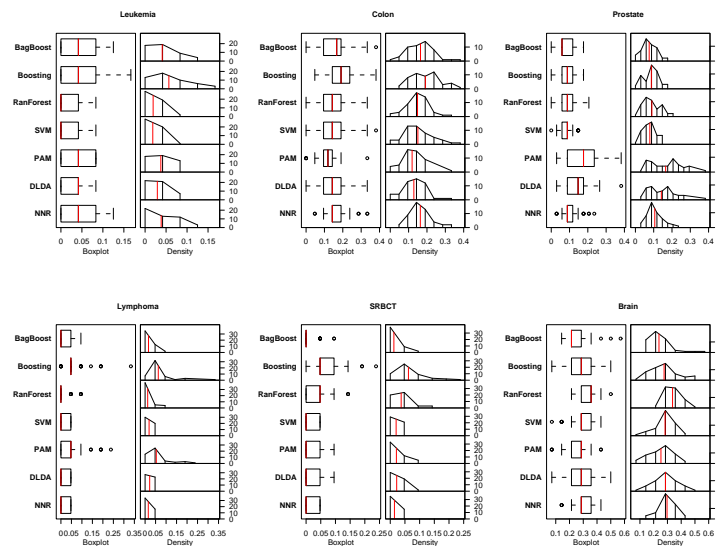


Figure 18: Boxplots and density curves of the misclassification rates for 7 classifiers on 6 microarray datasets, based on 50 random splits into learning and test sets. The vertical red lines highlight the median (boxplots) and the mean value (density curves).

the largest dataset in the analysis. This may point out that the success of such methodologically simple tools is limited to gene expression datasets with small sample size. BagBoosting achieves the lowest error rates on the prostate, SRBCT and brain data. It consistently improves upon plain boosting on all six datasets, which already legitimates the additional computing effort by using bagged stumps as a learner in boosting.

3.2 Simulation Studies

Our motivation to run simulation studies is three-fold. First, due to the scarcity of samples in real datasets, it is hard to detect relevant differences and to assess clinically relevant improvements for the classifiers. We will thus run a benchmark study on independent realizations of large gene expression datasets. Second, due to the lack of knowledge about the data generating process and the underlying probability distribution on real datasets, we cannot draw any inference about BagBoostings heuristically claimed improvement of bias and variance, i.e. mean squared error. Third, since we know the true response model on the simulated data, we can check how accurately BagBoosting recovers it from data, which is important to get an idea how well the BagBoosting model can be trusted in real datasets to draw biological conclusions. All these tasks critically hinge on a realistic simulation model for gene expression data. It is as follows.

Step 1: Estimating correlation and means

Use a real gene expression dataset of choice for estimating the $(p \times p)$ -covariance matrix Σ , as well as the p -dimensional mean vectors $\mu^{(k)} = (\mu_1^{(k)}, \dots, \mu_p^{(k)})$ from the samples of class $k \in \{0, 1\}$.

Step 2: Generating new gene expression profiles

For an arbitrary sample size n of choice repeat independently:

- (i) Generate a random vector by the p -dimensional multivariate standard normal distribution,

$$\mathbf{z} \sim \mathcal{N}_p(0, 1_{p \times p})$$

- (ii) Transform \mathbf{z} into a gene expression profile via

$$\mathbf{x} = B\mathbf{z} + \hat{\mu}^{(k)},$$

where B is a square root of the covariance matrix Σ , determined by singular value decomposition.

Step 3: Response model

Determine conditional probabilities and class labels by one of 3 response models of different complexity

- (a) Additive model with 10 genes

Use a set of 10 genes, assume w.l.o.g. that these are the first 10 genes. Then,

$$F(\mathbf{x}) = \sum_{j=1}^{10} x_j.$$

- (b) Weighted additive combination of 25 genes

Use a set of 25 genes, assume w.l.o.g. that these are the first 25 genes. Fix a set of coefficients β_j , for example randomly drawn from a uniform distribution on $[1, 3.5]$. Then,

$$F(\mathbf{x}) = \sum_{j=1}^{25} \beta_j x_j.$$

- (c) Complex interaction model with 25 genes

Use a set of 25 genes, assume w.l.o.g. that these are the first 25 genes. Fix three sets of coefficients β_j , γ_j and δ_j , for example, each is randomly drawn from a uniform distribution on $[0, 2]$, $[0, \frac{2}{10}]$ and $[0, \frac{1}{10}]$, respectively. Then,

$$F(\mathbf{x}) = \sum_{j=1}^{25} \beta_j x_j \cdot (1 + \sum_{j=1}^{25} \gamma_j x_j) \cdot (1 + \sum_{j=1}^{25} \delta_j x_j).$$

We regard $F(\mathbf{x})$ as the log-odds ratio, which can be converted into a conditional class probability $p(\mathbf{x})$. Finally, the class label $y(\mathbf{x})$ is randomly generated from a Bernoulli experiment, i.e.

$$p(\mathbf{x}) = \frac{1}{1 + \exp(-F(\mathbf{x}))},$$

$$y(\mathbf{x}) \sim \text{Bernoulli}(p(\mathbf{x})).$$

Note that the choice of genes and coefficients happens by a random mechanism, but is fixed for all samples in a simulation run.

Using this recipe, we can generate an arbitrary number of *iid* gene expression profiles that follow the covariance and differential expression properties of a microarray dataset of choice. For our empirical work, we considered the structure of leukemia, colon and prostate data. Due to space constraints and since the conclusions drawn from different structures were very similar, we here display the results for leukemia data only and refer to our supplementary webpage <http://stat.ethz.ch/~dettling/bagboost.html> for the complete information. The response models yield decision boundaries of various complexity and result in a Bayes error (theoretical misclassification risk) between 5-10%. As this is even more than the estimated generalization error on many real gene expression datasets, we conjecture that our response models are sufficiently complicated and realistic.

Classification Results

For a discussion of the predictive potential, and for testing the null hypothesis of equal performance, we run a benchmark study on simulated gene expression data with several classifiers. In each simulation experiment, we generated a learning set of 200 arrays and a large test set with 1000 observations, both with balanced class distributions. Exactly as on the real data, the preselection of 200 genes, as well as tuning and fitting of the classifiers was performed on the learning data, before the misclassification risk was estimated by the fraction of predicted test set class labels differing from the true one. This simulation experiment was independently repeated 100 times. The error rates are illustrated in Figure 19, where again the vertical red lines correspond to the median and mean value in the boxplots and density curves, respectively. The *iid* property of the error estimates also allows to formally test the null hypothesis that classifiers perform equally. We focus on pairwise comparisons of BagBoosting against its competitors. Table 25 reports the error rates for each classifier, averaged over the 100 simulation runs, as well as the number of runs where it was less accurate than BagBoosting. In a third column, we report the *p*-value of the two-sided signed rank test for the null hypothesis. In our simulation study with correlation

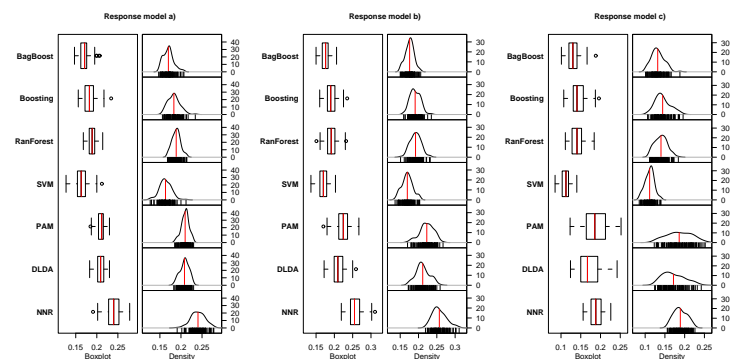


Figure 19: Misclassification rates for outsample classification on simulated gene expression data with various classifiers and three different response models. The left panels show boxplots where the median is highlighted in red, the right panels show density curves where the red vertical line corresponds to the mean error rate.

and differential expression structure from the leukemia dataset, the relation among the classifiers does not vary much over the three response models. Support vector machines is clearly the best classifier, showing a highly significant advantage in the error rates. BagBoosting is second best and yields significantly better predictions than the remaining competitors: the hardest challenger is random forests, but even when using response model c) with interactions of third order, BagBoosting has an edge. This is quite surprising, since random forests are built from mid-sized trees that encompass such interaction terms, whereas BagBoosting with stumps only fits a main effect model. Although boosting shows a much better performance on simulated compared to real data, it remains clearly behind BagBoosting. The lag is strongly significant and increases with the complexity of the response model. The difference between these top four methods and the benchmarks PAM/DLDA/NNR is much bigger than on

	Response a)			Response b)		
	Error	+/-	<i>p</i> -val	Error	+/-	<i>p</i> -val
BagBoost	17.02%	-	-	17.57%	-	-
Boosting	18.32%	89	0.0000	19.09%	96	0.0000
RanFor	18.87%	93	0.0000	19.19%	87	0.0000
SVM	16.33%	36	0.0000	16.94%	31	0.0000
PAM	20.99%	100	0.0000	22.25%	99	0.0000
DLDA	20.83%	99	0.0000	21.15%	98	0.0000
kNN	24.04%	100	0.0000	25.67%	100	0.0000
Response c)						
	Error	+/-	<i>p</i> -val			
BagBoost	13.19%	-	-			
Boosting	14.42%	97	0.0000			
RanFor	14.00%	68	0.0002			
SVM	11.13%	7	0.0000			
PAM	18.61%	96	0.0000			
DLDA	17.18%	91	0.0000			
kNN	18.90%	100	0.0000			

Table 25: Misclassification rates for outsample classification on 100 iid simulation experiments with various classifiers and three different response models, as well as the number of simulations where each of the classifiers was worse than BagBoosting, and the *p*-value for the two-sided signed rank test that the performance is equal.

real data, yielding further evidence that the benchmarks' success is limited to small datasets.

Improvement of Bias and Variance

The knowledge about the probability distribution $\mathbb{P}[y = 1|\mathbf{x}]$ in simulation experiments allows to check whether BagBoosting has the heuristically claimed property to lower bias *and* variance in comparison with single, bagged and boosted stumps. There is no bias-variance decomposition of the 0/1-misclassification error, we

are thus running such an analysis on the basis of the predicted conditional class probabilities $\hat{p}(\mathbf{x})$. Since we are not focusing on a particular input value \mathbf{x} , but want to learn about the precision of an estimated probability structure over the whole input space \mathcal{X} , the measure of interest is the integrated mean squared error

$$IMSE(\hat{p}) = \int_{\mathcal{X}} \text{Var}(\hat{p}(\mathbf{x})) + \mathcal{E}[\hat{p}(\mathbf{x}) - p(\mathbf{x})]^2 d\mathcal{F}(\mathbf{x}),$$

where \mathcal{F} is the probability distribution on the input space. The *IMSE* is decomposed into variance and bias and can be estimated via approximating the outer integral by randomly drawing a sufficiently large number of input values \mathbf{x}_i from \mathcal{F} . Variance and expectation are approximated by averaging over a sufficiently large number of simulations: an estimation of integrated variance and integrated squared bias of a probability function are given by

$$\begin{aligned} \text{Var}(\hat{p}) &= \frac{1}{T} \sum_{i=1}^T \left[\frac{1}{S} \sum_{k=1}^S (\hat{p}_k(\mathbf{x}_i) - \bar{p}(\mathbf{x}_i))^2 \right] \\ \text{Bias}(\hat{p})^2 &= \frac{1}{T} \sum_{i=1}^T \left[\frac{1}{S} \sum_{k=1}^S \hat{p}_k(\mathbf{x}_i) - p_k(\mathbf{x}_i) \right]^2, \end{aligned}$$

where $\hat{p}_k(\mathbf{x}_i)$ and $p_k(\mathbf{x}_i)$ denote the estimated and true probabilities in the k th simulation run, and where $\bar{p}(\mathbf{x}_i) = \sum_{k=1}^S \hat{p}_k(\mathbf{x}_i)$. For checking the bias-variance properties of BagBoosting versus single stumps, bagged stumps and boosted stumps, we performed $S = 100$ simulation runs and generated learning sets of 200 observations each. Gene preselection and fitting the four predictors was redone on each learning set, before out-of-sample probability estimates $\hat{p}(\mathbf{x}_i)$ for $T = 1,000$ fixed test points were computed. The results are summarized in Table 26.

All our heuristical claims are confirmed by the simulation results which are consistent over the three response models. In terms of the *IMSE*, BagBoosting is best, ranking before the about equally good bagging and boosting, whereas single stumps

	Response a)			Response b)		
	<i>MSE</i>	<i>Var</i>	<i>Bias</i> ²	<i>MSE</i>	<i>Var</i>	<i>Bias</i> ²
Stumps	0.102	0.041	0.061	0.116	0.040	0.075
Bagging	0.069	0.010	0.059	0.083	0.010	0.073
Boosting	0.076	0.048	0.028	0.086	0.056	0.030
BagBoost	0.045	0.020	0.026	0.050	0.023	0.027

	Response c)		
	<i>MSE</i>	<i>Var</i>	<i>Bias</i> ²
Stumps	0.128	0.037	0.091
Bagging	0.096	0.008	0.088
Boosting	0.090	0.047	0.043
BagBoost	0.056	0.019	0.037

Table 26: Estimates of integrated mean squared error, variance and squared bias for conditional class probabilities, obtained from four different prediction methods on simulated gene expression data with three different response models.

are clearly worse. This means that BagBoosting not only yields the best 0/1-classification, but also the most precise estimates of the conditional class probabilities. As expected from theory, bagged stumps have considerably lower variance but similar bias as single stumps. On the other hand, boosted stumps have slightly higher variance than single stumps, but they compensate by a much bigger gain in bias. Finally, as heuristically derived, BagBoosting has both lower bias *and* variance than stumps. The use of a bagged stump as base learner in boosting clearly pays off: while the variance is about 60% less than for plain boosting, the bias does not increase and is even about 10% lower. Hence we conjecture that BagBoosting really exploits the synergy of bagging and boosting; the mechanisms for improving bias and variance work in the microarray setting. These results confirm our previous evidence that BagBoosting improves upon bagging or boosting, and that it is worth the additional computing effort.

Model Recovery

The purpose of this section is to check how well BagBoosting recovers the true response model in such a difficult setting as microarray data with thousands of highly correlated predictor variables. The inference is based on a single (but representative) learning set \mathcal{L} from our simulation model with response type a) and correlation/differential expression structure of the leukemia data. As a slight modification, we standardized all genes to unit variance to facilitate the inference. Then, we select the 200 most discriminative genes according to the Wilcoxon statistic, run BagBoosting and rewrite the fit in its componentwise additive representation from equation (24). For each variable x_j , the smoothed step functions, obtained by combining numerous stumps, are centered to zero mean and scaled to unit variance across the 200 fitted values; this yields the univariate functions $h(x_j)$. The scaling factor is then defined as the coefficient $\hat{\theta}_j$. It reflects the importance of the j th variable in the final BagBoosting estimate.

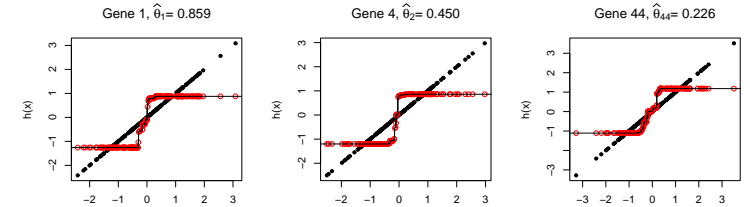


Figure 20: BagBoosting model fit and true predictors: the black dots represent the linear univariate functions for 3 genes in the simulation model. Superimposed are the smoothed univariate step functions and their fitted values (grey circles), obtained by BagBoosting with stumps.

Figure 20 shows the true function values x_j (black dots on the diagonal), along with the smoothed step functions and their fitted values (grey circles) from the final BagBoosting estimate. Displayed are the three most important predictor variables, i.e.

the genes with the biggest coefficients $\hat{\theta}_j$. Note that the estimated functions $h(x_j)$ are accurately centered, but their shape is smooth and linear in just a restricted interval around zero. This is not too surprising, since Logit(Bag)Boost mainly focuses on accurate estimation in a region around the decision boundary, and data points that are classified with higher confidence are regarded as less interesting. This argument is supported by the fact that a log-odds-ratio of $F(\mathbf{x}) = 1$, where the step functions level out, corresponds to an estimated probability of $p(\mathbf{x}) = 0.88$.

Important Genes				True Genes			
Gene	$R(\hat{\theta}_j)$	$\hat{\theta}_j$	Corr	Gene	$R(\hat{\theta}_j)$	$\hat{\theta}_j$	Corr
1	1	0.859	1.000	1	1	0.859	1.000
2	2	0.450	1.000	2	2	0.450	1.000
44	3	0.226	0.494	3	19	0.129	0.810
582	4	0.220	0.646	4	61	0.047	0.752
1026	5	0.217	0.623	5	66	0.041	0.864
1072	6	0.212	0.482	6	74	0.033	0.846
520	7	0.204	0.710	7	88	0.026	0.832
930	8	0.197	0.582	8	137	0.012	0.779
1894	9	0.188	0.661	9	149	0.009	0.655
261	10	0.183	0.859	10	155	0.008	0.694

Table 27: Comparison of the 10 true and the 10 most important BagBoosting genes: given are their estimated model coefficients $\hat{\theta}_j$, the ranking $R(\hat{\theta}_j)$ of the coefficients according to their magnitude, and the maximal correlation of each gene to one of the 10 genes from the other group.

Table 27 reports the estimated model coefficients $\hat{\theta}_j$ for the 10 true genes from the simulation model, as well as for the 10 most important features in the BagBoosting fit. The importance ranking $R(\hat{\theta}_j)$ is determined by the magnitude of $\hat{\theta}_j$. We observe a small overlap of only 2 genes between the true and the most important variables. Moreover, many of the true genes have low importance $\hat{\theta}_j$, far from the theoretical value $\theta_j = 1$. On the

other hand, the 10 most important genes have limited influence, too. BagBoosting spreads the responsibility on the shoulders of a larger gene set: 6 of the 200 available genes were never selected at all, 141 have a neglectable influence with $\hat{\theta}_j < 0.05$, whereas the remaining 53 genes form the core for class prediction. The discrepancy between the true and the fitted model may seem disappointing, but is explained as follows. While only 10 genes determine the hidden target function $F(\mathbf{x})$ and the exact course of the decision boundary, many more are closely associated with the class labels y on the learning set that BagBoosting is presented with, due to the high correlation and the (often strong) differential expression for the genes. This is confirmed with our empirical analysis in Table 27, where we show the maximal correlation of each gene from the true and the important feature set to a member of the other group. These correlations are above 0.5 throughout and indicate that the true genes are substituted by very similarly regulated genes. The discrepancy between the true and the fitted model is a flaw that not only BagBoosting suffers from. When analyzing the fitted models of the competing classifiers (where possible, results not shown), we observe a similar disagreement, which is inherent to the highly collinear gene expression data. This emphasizes that the prediction models, despite their merits in cancer diagnosis, must be treated with enormous care for drawing biological conclusions.

4. Conclusions

The goal in class prediction with microarray data is a precise classification of cancerous malignancies at an early stage, allowing for directed and more successful therapies. Important for this task are classification algorithms that can deal with the high dimensionality of gene expression data, and that exploit as much of the available information as possible. We propose a hybrid approach of ensemble methods, where bagged stumps are employed as the base learner in boosting. On both real and simulated gene expression data, we show that BagBoosting consistently lowers the

misclassification error of plain boosting and bagging, and that it is competitive in comparison to modern prediction tools such as random forests and support vector machines. On simulated data, we furthermore provide sound empirical evidence that BagBoosting results in more precise conditional class probability estimates by reducing both the bias and variance of the base algorithm.

It has been recognized by several researchers that the introduction of randomness into ensemble schemes can improve their predictive potential. Due to the Monte Carlo approximation of the bootstrap expectation, BagBoosting is a non-deterministic rule, too. However, we do not think that this is the principal explanation for its success, which is rather caused by a variance reduction achieved by averaging over a set of stumps, a very crude and unstable learner for microarray data. In simulation studies, this variance reduction is shown to propagate within the boosting algorithm, which however still lowers the bias as desired. BagBoosting thus really combines the advantages of the two ensemble methods it is built from.

Critical voices are often concerned that the time-consuming effort of fitting complex prediction tools such as (Bag)Boosting, random forests, etc. is not legitimated by the small empirical improvement over much simpler methods on microarray data. Although at present, the advantage of BagBoosting versus the nearest neighbor rule or diagonal linear discriminant analysis on gene expression data with limited number of samples is not very big in terms of the 0/1-misclassification error. But the difference grows to significant size on larger microarray problems and becomes very pronounced on simulated datasets with more than 1,000 expression profiles. While already a small improvement today can save an additional patients life, sophisticated class prediction tools such as BagBoosting are expected to display their full benefit only in future studies with larger sample size.

Outlook

Genomic investigations have brought valuable insight in many areas of bio-medical research. One example is cancer diagnosis and prognosis based on microarray gene expressions, where a wealth of work has been published. Most publications have shown some success in regarding data from one particular high-throughput biotechnology as a replacement of the traditional tumor markers. The future challenge is to integrate the predictive information from microarrays, proteomic assays and yet-to-be-developed biotechniques as a supplement to clinical tumor markers in a biologically reasonable and interpretable fashion. It is important to note that the dimensionality-problem is still present: acutally, the amount of information gets even larger by considering multiple sources of data. This rises the important question of drawing inference about prediction models. The question is which variables contribute significantly to tumor diagnosis, and which others are not worthwhile to be considered. An extension with practical relevance could be to incorporate the cost, time consumption and accessibility of the features into a formalized decision.

Although some progress has been made, the function and role of many genes remains unknown. The often employed guilt-by-association principle in conjunction with microarray data has provided some insight, but due to the mostly small sample size and the high variability of the output, its potential is limited. The future challenge is to show gene relations and functions by combining data across studies, across measurement technologies and across biological systems, with algorithms that work under biological constraints. Only this will allow to fully exploit the formidable potential and the massive financial investment that are made in genomic studies.

Bibliography

- [1] Li C. and Wong W. (2001), *Model-Based Analysis of Oligonucleotide Arrays: Expression Index Computation and Outlier Detection*. PNAS, **98**, 31–36.
- [2] Irizarry R.A., Bolstad B.M., Collin F., Cope L.M., Hobbs B. and Speed T.P. (2003), *Summaries of Affymetrix GeneChip Probe Level Data*. Nucleic Acids Research, **31**, e15.
- [3] Naef F., Lim D.A., Patil N. and Magnasco M.O. (2003), *From Features to Expression: High-Density Oligonucleotide Array Analysis Revisited*. In *Proceedings of DIMACS Workshop on Analysis of Gene Expression Data*.
- [4] Wu Z., Irizarry R., Gentleman R., Murillo M. and Spencer F. (2003), *A Model Based Background Adjustment for Oligonucleotide Expression Arrays*. Technical report, Johns Hopkins University.
- [5] Eisen M.B., Spellman P.T., Brown P.O. and Botstein D. (1998), *Cluster Analysis and Display of Genome-Wide Expression Patterns*. PNAS, **95**, 14863–14868.
- [6] Hartigan J.A. and Wong M.A. (1979), *A k-Means Clustering Algorithm*. Applied Statistics, **28**, 100–108.
- [7] Kohonen T. (1982), *Analysis of a Simple Self-Organizing Process*. Biological Cybernetics, **43**, 59–69.
- [8] Yeung K.Y. and Ruzzo W. (2001), *Principal Component Analysis for Clustering Gene Expression Data*. Bioinformatics, **17**, 763–774.
- [9] Benjamini Y. and Hochberg Y. (1995), *Controlling the False Discovery Rate: A Practical Approach*. JRSSB, **57**, 289–300.
- [10] Dettling M. and Bühlmann P. (2002), *Supervised Clustering of Genes*. Genome Biology, **3**, research 0069.1–0069.15.
- [11] Dettling M. and Bühlmann P. (2004), *Finding Predictive Gene Groups from Microarray Data*. Journal of Multivariate Analysis, **90**, 106–131.
- [12] Diaz-Uriarte R. (2003), *A Simple Method for Finding Molecular Signatures from Gene Expression Data*. Technical Report Nr. 004, Spanish National Cancer Center (CNIO).
- [13] Jörnsten R. and Yu B. (2003), *Simultaneous Gene Clustering and Subset Selection for Sample Classification via MDL*. Bioinformatics, **19**, 1100–1109.
- [14] Bair E. and Tibshirani R. (2004), *Semi-Supervised Methods to Predict Patient Survival from Gene Expression Data*. PLoS Biology, **2**, e108.
- [15] Freund Y. and Schapire R. (1996), *Experiments with a New Boosting Algorithm*. In *Machine Learning: Proceedings to the 13th International Conference*, pages 148–156, Morgan Kaufman, San Francisco.
- [16] Friedman J., Hastie T. and Tibshirani R. (2000), *Additive Logistic Regression: A Statistical View of Boosting*. Annals of Statistics, **28**, 337–407, (with discussion).
- [17] Dettling M. and Bühlmann P. (2003), *Boosting for Tumor Classification with Microarray Data*. Bioinformatics, **19**, 1061–1069.

- [18] Ben-Dor A., Bruhn L., Friedman N., Nachman I., Schummer M. and Yakhini Z. (2000), *Tissue Classification with Gene Expression Profiles*. Journal of Computational Biology, **7**, 559–583.
- [19] Dudoit S., Fridlyand J. and Speed T. (2002), *Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data*. Journal of the American Statistical Association, **97**, 77–87.
- [20] Dettling M. (2004), *BagBoosting for Tumor Classification with Gene Expression Data*. Technical Report Nr. 122, Seminar für Statistik, ETH Zürich.
- [21] Weinstein J., Myers T., O'Connor P. et al. (1997), *An Information-Intensive Approach to the Molecular Pharmacology of Cancer*. Science, **275**, 343–349.
- [22] Tamayo P., Slonim D., Mesirov J., Zhu Q., Dmitrovsky E., Lander E.S. and Golub T.R. (1999), *Interpreting Gene Expression with Self-Organizing Maps: Methods and Application to Hematopoietic Differentiation*. PNAS, **96**, 2907–2912.
- [23] Ben-Dor A., Shamir R. and Yakhini Z. (1999), *Clustering Gene Expression Patterns*. Journal of Computational Biology, **6**, 281–297.
- [24] Hastie T., Tibshirani R., Botstein D. and Brown P. (2001), *Supervised Harvesting of Expression Trees*. Genome Biology, **2**, research 0003.1–0003.12.
- [25] Nguyen D. and Rocke D. (2002), *Tumor Classification by Partial Least Squares Using Microarray Gene Expression Data*. Bioinformatics, **18**, 39–50.
- [26] Geladi P. and Kowalski B.R. (1986), *Partial Least Squares Regression: A Tutorial*. Analytica Chimica Acta, **185**, 1–17.

- [27] Park P., Pagano M. and Bonetti M. (2001), *A Nonparametric Scoring Algorithm for Identifying Informative Genes from Microarray Data*. In *Pacific Symposium on Biocomputing*, volume 6, pages 52–63.
- [28] Wilcoxon F. (1945), *Individual Comparisons by Ranking Methods*. Biometrics, **1**, 80–83.
- [29] Hastie T., Tibshirani R., Eisen M.B. et al. (2000), *Gene Shaving as a Method of Identifying Distinct Sets of Genes with Similar Expression Patterns*. Genome Biology, **1**, research 0003.1–research 0003.21.
- [30] Hastie T. and Tibshirani R. (1998), *Classification by Pairwise Coupling*. Annals of Statistics, **26**, 451–471.
- [31] Allwein E., Schapire R. and Freund Y. (2000), *Reducing Multiclass to Binary: A Unifying Approach for Margin Classifiers*. Journal of Machine Learning Research, **1**, 113–141.
- [32] Golub T., Slonim D., Tamayo P. et al. (1999), *Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring*. Science, **286**, 531–538.
- [33] West M., Blanchette C., Dressman H. et al. (2001), *Predicting the Clinical Status of Human Breast Cancer by Using Gene Expression Profiles*. Proceedings of the National Academy of Science, **98**, 11462–11467.
- [34] Alon U., Barkai N., Notterdam D., Gish K., Ybarra S., Mack D. and Levine A. (1999), *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**, 6745–6750.
- [35] Singh D., Febbo P., Ross K. et al. (2002), *Gene Expression Correlates of Clinical Prostate Cancer Behavior*. Cancer Cell, **1**, 203–209.

- [36] Khan J., Wei J., Ringner M. *et al.* (2001), *Classification and Diagnostic Prediction of Cancer Using Gene Expression Profiling and Artificial Neural Networks*. *Nature Medicine*, **6**, 673–679.
- [37] Alizadeh A., Eisen M., Davis E. *et al.* (2000), *Distinct Types of Diffuse Large B-Cell-Lymphoma Identified by Gene Expression Profiling*. *Nature*, **403**, 503–511.
- [38] Pomeroy S., Tamayo P., Gaasenbeek M. *et al.* (2002), *Prediction of Central Nervous System Embryonal Tumor Outcome Based on Gene Expression*. *Nature*, **415**, 436–442.
- [39] Ross D.T., Scherf U., Eisen M.B. *et al.* (2000), *Systematic Variation in Gene Expression Patterns in Human Cancer Cell Lines*. *Nature Genetics*, **24**, 227–235.
- [40] Fix E. and Hodges J. (1951), *Discriminatory Analysis - Non-parametric Discrimination: Consistency Properties*. Technical Report No. 4, US Air Force School of Aviation Medicine, Texas.
- [41] Breiman L., Friedman J., Olshen R. and Stone C. (1984), *Classification and Regression Trees*. Wadsworth.
- [42] Ripley B. (1996), *Pattern Recognition and Neural Networks*. Cambridge University Press.
- [43] Furey T., Cristianini N., Duffy N., Bednarski D., Schummer M. and Haussler D. (2000), *Support Vector Machine Classification and Validation of Cancer Tissue Samples Using Microarray Expression Data*. *Bioinformatics*, **16**, 906–914.
- [44] Lee Y. and Lee C. (2003), *Classification of Multiple Cancer Types by Multicategory Support Vector Machines Using Gene Expression Data*. *Bioinformatics*, **19**, 1132–1139.

- [45] Efron B. and Tibshirani R. (1998), *The Problem of Regions*. *Annals of Statistics*, **26**, 1687–1718.
- [46] Le Cessie S. and Van Houwelingen J. (1990), *Ridge Estimators in Logistic Regression*. *Applied Statistics*, **41**, 191–201.
- [47] Eilers P., Boer J., Van Ommen G.J. and Van Houwelingen H. (2001), *Classification of Microarray Data with Penalized Logistic Regression*. In *Proceedings of SPIE: Progress in Biomedical Optics and Imaging*, volume 2, pages 187–198.
- [48] Zhu J. and Hastie T. (2002), *Classification of Gene Microarrays by Penalized Logistic Regression*. Technical report, Department of Statistics, University of Stanford.
- [49] Bickel P., Klaassen C., Ritov Y. and Wellner J. (1993), *Efficient and Adaptive Estimation for Semiparametric Models*. Johns Hopkins University Press.
- [50] Dudoit S. and Fridlyand J. (2002), *A Prediction-Based Resampling Method to Estimate the Number of Clusters in a Dataset*. *Genome Biology*, **3**, research 0036.1–0036.21.
- [51] Tibshirani R., Walther G. and Hastie T. (2000), *Estimating the Number of Clusters in a Dataset via the Gap Statistic*. Technical Report 208, Department of Statistics, University of Stanford.
- [52] Huang E., Chen S., Dressman H. *et al.* (2003), *Gene Expression Predictors of Breast Cancer Outcomes*. *The Lancet*, **361**, 1590–1596.
- [53] Hoerl A. and Kennard R. (1970), *Ridge Regression: Biased Estimation for Nonorthogonal Problems*. *Technometrics*, **12**, 55–67.

- [54] Van't Veer L., Dai H., Van de Vijver M. *et al.* (2002), *Gene Expression Profiling Predicts Clinical Outcome of Breast Cancer*. *Nature*, **415**, 530–535.
- [55] Slonim D., Tamayo P., Mesirov J., Golub T. and Lander E. (2000), *Class Prediction and Discovery Using Gene Expression Data*. In *Proceedings of the 4th International Conference on Computational Molecular Biology*, pages 263–272, Universal Academy Press, Tokyo, Japan.
- [56] Zhang H., Yu C., Singer B. and Xiong M. (2001), *Recursive Partitioning for Tumor Classification with Gene Expression Microarray Data*. *PNAS*, **98**, 6730–6735.
- [57] Hampel F., Ronchetti E., Rousseeuw P. and Stahel W. (1986), *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York.
- [58] Breiman L. (1999), *Prediction Games and Arcing Algorithms*. *Neural Computation*, **11**, 1493–1517.
- [59] Hastie T., Tibshirani R. and Friedman J. (2001), *The Elements of Statistical Learning*. Springer, New York.
- [60] Quinlan J.R. (1993), *C4.5: Programs for Machine Learning*. Morgan Kaufman, San Francisco.
- [61] Spang R., Blanchette C., Zuzan H., Mark J., Nevins J. and West M. (2001), *Prediction and Uncertainty in the Analysis of Gene Expression Profiles*. In *Proceedings of the German Conference on Bioinformatics*.
- [62] Dudoit S. and Fridlyand J. (2003), *Statistical Analysis of Gene Expression Data*, chapter Classification in Microarray Experiments, pages 93–158. Chapman and Hall, New York.
- [63] Breiman L. (1996), *Bagging Predictors*. *Machine Learning*, **24**, 123–140.

- [64] Bühlmann P. and Yu B. (2000), *Discussion Paper on Additive Logistic Regression: A Statistical View of Boosting*. *Annals of Statistics*, **28**, 377–386.
- [65] Bühlmann P. and Yu B. (2003), *Boosting with the L_2 -Loss: Regression and Classification*. *JASA*, **98**, 324–339.
- [66] Friedman J. (2002), *Stochastic Gradient Boosting*. *Computational Statistics and Data Analysis*, **38**, 367–378.
- [67] Breiman L. (2001), *Using Iterated Bagging to Debias Regressions*. *Machine Learning*, **45**, 261–277.
- [68] Friedman J. and Popescu B. (2003), *Importance Sampled Learning Ensembles*. Technical report, Department of Statistics, Stanford University.
- [69] R Development Core Team, Vienna, Austria (2004), *R: A Language and Environment for Statistical Computing*. ISBN 3-900051-00-3.
- [70] Breiman L. (2001), *Random Forests*. *Machine Learning*, **45**, 5–32.
- [71] Liaw A. and Wiener M. (2002), *Classification and Regression by Random Forest*. *R News*, **2**, 18–22.
- [72] Meyer D., Leisch F. and Hornik K. (2003), *The Support Vector Machine Under Test*. *Neurocomputing*, **55**, 169–186.
- [73] Burges C. (1998), *A Tutorial on Support Vector Machines for Pattern Recognition*. *Knowledge Discovery and Data Mining*, **2**, 121–167.
- [74] Meyer D. (2001), *Support Vector Machines*. *R News*, **1**, 23–26.

- [75] Chang C. and Lin C. (2001), *LIBSVM: A Library for Support Vector Machines*. Available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [76] Tibshirani R., Hastie T., Narasimhan B. and Chu G. (2002), *Diagnosis of Multiple Cancer Types by Shrunk Centroids of Gene Expression*. PNAS, **99**, 6567–6572.
- [77] Hothorn T., Leisch F., Zeileis A. and Hornik K. (2003), *The Design and Analysis of Benchmark Experiments*. Technical Report Nr. 82, Institut für Statistik, Wirtschaftsuniversität Wien.

Curriculum Vitae

I, Marcel Dettling from Oberiberg (SZ), was born on October 21, 1974. After visiting the primary school in Wetzikon (ZH) from 1981 to 1987, I spent six years at the Kantonsschule Zürcher Oberland in Wetzikon, from where I graduated in 1993 with a type C matura.

I began my studies in Mathematics at ETH Zürich in October 1995. After the first six semesters I suspended my education for half a year to gain practical experience in the head office of Zurich Financial Services. I completed my studies in August 2000 with a diploma thesis on “Volatility and Risk Estimation with High Frequency Data”.

Since November 2000 I have been working as a teaching assistant in the Department of Mathematics at ETH Zürich. At the same time, I have been working on this doctoral thesis, which was concluded in June 2004.