Statistical Applications in Genetics and Molecular Biology

Volume 3, *Issue* 1 2004 *Article* 37

A Compendium to Ensure Computational Reproducibility in High-Dimensional Classification Tasks

Markus Ruschhaupt[*] Wolfgang Huber[†]

Annemarie Poustka[‡] Ulrich Mansmann[**]

[*] Division of Molecular Genome Analysis, German Cancer Research Centre, m.ruschhaupt@dkfz-heidelberg.de

[†] German Cancer Research Center, Heidelberg, Germany, huber@ebi.ac.uk

[‡] Division of Molecular Genome Analysis, German Cancer Research Centre, a.poustka@dkfz-heidelberg.de

[**] Department for Medical Biometrics/Informatics, University of Heidelberg, mansmann@imbi.uni-heidelberg.de

A Compendium to Ensure Computational Reproducibility in High-Dimensional Classification Tasks*

Markus Ruschhaupt, Wolfgang Huber, Annemarie Poustka, and Ulrich Mansmann

Abstract

We demonstrate a concept and implementation of a compendium for the classification of high-dimensional data from microarray gene expression profiles. A compendium is an interactive document that bundles primary data, statistical processing methods, figures, and derived data together with the textual documentation and conclusions. Interactivity allows the reader to modify and extend these components. We address the following questions: how much does the discriminatory power of a classifier depend on the choice of the algorithm that was used to identify it; what alternative classifiers could be used just as well; how robust is the result. The answers to these questions are essential prerequisites for validation and biological interpretation of the classifiers. We show how to use this approach by looking at these questions for a specific breast cancer microarray data set that first has been studied by Huang et al. (2003).

KEYWORDS: compendium, machine learning, classification, microarray, cancer

*We thank both reviewers and the editor for their constructive criticism which was stimulating in shaping our compendium. Furthermore, we thank Robert Gentleman, Günther Sawitzki, and Anja von Heydebreck for critical comments and fruitful discussions. Thanks to Jennifer Pitman for providing the software and background information on the use of the Bayesian Binary Prediction Tree Models. This work was supported by NGFN-01GR0456 and NGFN-01GR0101.

1 Introduction

Reproducibility of calculations is a longstanding issue within the statistical community (Leisch and Rossini, 2003; Gentleman, 2004). Due to the complexity of the algorithms, the size of the data sets, and the limitations of the medium of printed paper it is usually not possible to report all the minutiae of the data processing and statistical computations. Large software projects almost surely contain programming errors but in the scientific community there is little critical approval of results based on complex calculations. It should be possible to check the software behind a complex data analysis.

Microarrays are a recent biotechnology that offer the hope of improved cancer classification, providing clinicians with the information to choose the most appropriate form of treatment (van't Veer et al., 2002; van de Vijver et al., 2002; Huang et al., 2003; Brenton and Caldas, 2003). A number of publications have presented clinically promising results by combining this new kind of biological data with specifically designed algorithmic approaches. To transfer these proofs of principle into clinical practice two criteria have to be fulfilled: reproducibility of the presented results and prospective verification of new data. Reproducing published results in this domain is harder than it may seem. Tibshirani and Efron (2002) report: ”We reanalyzed the breast cancer data from van't Veer et al. (2002). . . . Even with some help of the authors, we were unable to exactly reproduce this analysis.”

Molecular signatures cannot be directly read off from high-dimensional data. What we get may depend highly on the algorithm that we use to extract the signature. When a new molecular signature of presumed prognostic validity is presented, the scientist has to validate how much of the reported discrimation can be attributed to a real biological difference: the scientist needs to disentangle biology and algorithm. In this report, we implement a reproducible framework for this task to enter into the following questions:

1. Which classification result could be achieved using standard algorithms and is there a difference in classification quality between a standard algorithm and the proposed one?

2. If there is a substantial difference, what is the reason?

We show how to use this approach by looking at these questions for a specific breast cancer microarray data set that was first studied by Huang et al. (2003).

2 Methodological Background

2.1 Reproducible Calculations

To achieve reproducible calculations and to offer an extensible computational framework we use the tool of a *compendium* (Sawitzki, 1999; Leisch, 2002; Gentleman and Temple Lang, 2004; Sawitzki, 2002; Gentleman, 2004). A compendium is a document that bundles primary data, processing methods (computational code), derived data, and statistical output with the textual description and conclusions. It is interactive and extensible in the sense that it allows the reader to modify the processing options, plug in new data, or insert further algorithms and visualizations. Our compendium is based on the *vignette* and *packaging* technology available from the R and Bioconductor projects (Ihaka and Gentleman, 1996; Gentleman and Carey, 2002).

An R package is simply a directory (or a corresponding *tar* archive) structured in a defined format. It includes computer code, documentation, data, and revivable documents. The source of a revivable document is a so-called noweb (Knuth, 1992) file, a text file that in our case consists of a combination of LaTeX and R code chunks. Typically, it has the extension .Rnw. This file does not contain numerical or graphical results, instead, it contains the program code needed to obtain these. Using the R function Sweave the program code is evaluated and a new LaTeX file is created, which can be transformed into a PDF file that contains the text, the program code, and the resulting values, tables, and figures. Such a document is revivable in the sense that they can be automatically updated whenever data or analysis change by just altering the program code in the .Rnw file (Wolf).

Our compendium is the R package *compHuang* that contains the experimental data, the .Rnw file, and the resulting PDF file. This is the file you are reading right now. The .Rnw file contains further explanatory comments of the R code beyond what is displayed in the PDF file. By convention, lines with R code start with a '>'. A '+' indicates continuation across a line break.

2.2 Processing Methods

The computational part of the compendium covers the following tasks:

1. Given a family of classifiers and a fixed set of parameters, construct a discrimination rule from the data.

2. Find an optimal set of parameters for the family of classifiers.

3. Estimate the misclasification rate (MCR).

The computations rely on a number of `R` packages. The function `MCRestimate` in the `R` package of the same name implements tasks 2 and 3 by by two nested cross–validation loops. Cross–validation is an appropiate instrument to estimate the MCR if it is applied in a correct way (Ransohoff, 2004). One common error is the exclusion of preprocessing steps from the cross–validation. With preprocessing we refer to variable selection and aggregation. The exclusion can cause overfit and selection bias (Ambroise and McLachlan, 2002). To prevent this kind of error `MCRestimate` first combines classification algorithm and preprocessing stategies to create a new classification algorithm we term *complete classification procedure* (Ambroise and McLachlan, 2002). Then the cross-validations are performed with the complete classification procedure (Fig. 1).

The outer cross–validation loop is used to estimate the misclassification rate and the inner cross–validation loop is used to select the optimal parameters for a given complete classification procedure. It is crucial that the test set for the estimation of the MCR not be included in the cross–validation loop for the tuning of the parameters.

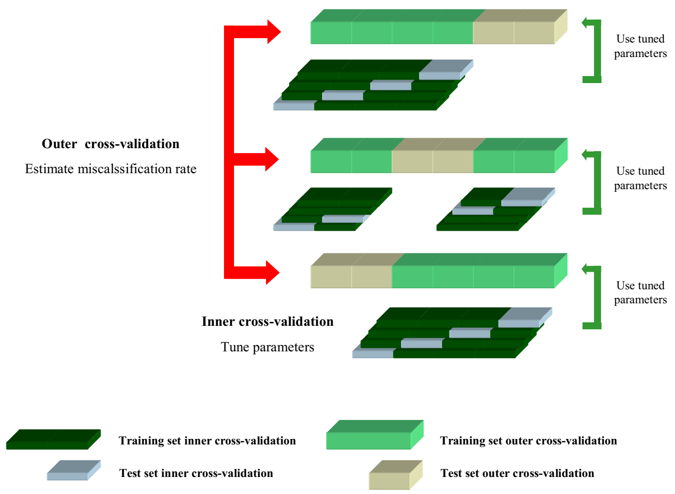

Use tuned parameters
Use tuned parameters
Use tuned parameters
Outer cross-validation
Estimate miscalssification rate
Inner cross-validation
Tune parameters
Training set inner cross-validation
Test set inner cross-validation
Training set outer cross-validation
Test set outer cross-validation

Figure 1: The cross-validation strategy implemeted in MCRestimate

Many different classification algorithms can be used for task 1. All that is needed is a wrapper that provides them with a uniform interface for `MCRes-`

timate. Here, we consider four algorithms from the canon of classification and machine learning: nearest shrunken centroids (PAM), support vector machine (SVM), penalized logistic regression (PLR), and random forest (RF). They can be combined with different strategies for data preprocessing such as variable selection and transformation. To see if there are substantial differences between selected standard methods and the algorithm used by Huang et al. (2003), we also wrote a wrapper for the Bayesian Binary Prediction Tree Model (BBT). To asure comparable results, the cross-validations for the different classification methods were performed with the same folds.

Random Forest: Breiman (2001) proposed a method that he called the random forest. Different decision trees are grown by introducing a random element into their construction, e. g. in the selection of a subset of predictor variables that is considered at each node. Each decision tree makes a prediction of one class for easch observation, and all predictions are combined to get the final classification result.

Class prediction by nearest shrunken centroids: Centroids are location estimates of the distribution of predictors within each class. The predicted class of a new sample is the one whose centroid is closest to it. Tibshirani et al. (2003) introduced the nearest shrunken centroid method, which uses denoised versions of the centroids as prototypes for each class.

Penalized logistic regression: The logistic regression model arises from the desire to model the posterior probabilities of two classes via a logistic transformation of linear functions in the predictors. Eilers et al. (2001) showed that ridge penalty can make logistic regression an effective tool for the classification of microarray data.

Support Vector Machines: In high-dimensional spaces an infinite number of hypersurfaces can be found that separate the training data. The support vector machine selects the one with the largest margin (Vapnik, 1999).

Bayesian Binary Prediction Tree Models: This is a Bayesian variant of decision trees that is described in Pittman et al. (2004). To quantify the split quality of a node, a Bayes factor measure for association is used. The tree is grown in a forward-selection process similar to traditional classification tree induction. In considering potential split candidates at any node, there may be a number with high Bayes factors, so that multiple possible trees at this node are suggested. A marginal likelihood for each tree is calculated which provides a weight for assessing trees and defines posterior probabilities that can be used to make predictions for new cases based on weighted average from the individual trees. Our wrapper to this procedure uses the Windows executable file BinTree.exe that can be downloaded from the website[1] of Pittman et al.

The next chapter is the core of the compendium and could stand on its own. It is divided into sub-chapters and presented in the standard format of a scientific report: Overview, Methods, Results, Comments and Conclusions. It is intended as an instrument that can be used by a statistical novice to combine the building blocks offered by the many available R packages for the exploration of classification problems as well as by an expert who may extend the offered framework with his/her own algorithms and expertise.

3 The compendium

3.1 Overview

Breast cancer is clinically heterogeneous, with varying natural history and response to treatment. Despite much effort to identify clinical measures of risk, methods to accurately predict an individual's clinical course are lacking. While lymph-node status at diagnosis is the most important measure for future recurrence and overall survival, it is a surrogate that is imperfect at best. Improved predictions of disease course, including lymph-node metastasis or recurrence, will profoundly affect clinical decisions.

Huang et al. (2003) used 89 tumor samples from patients with mostly infiltrating ductal carcinoma (78, 88%) for comparative measurements of gene expression. These samples were taken from a heterogeneous population, and were selected on the basis of clinical parameters and outcomes, to generate cases suitable for two focused studies: prediction of nodal metastatic states and relapse for breast cancer patients. The data set on nodal status prediction consists of 19 probes from patients with negative nodes and 18 probes of patients with more than 10 positive lymph nodes. The remaining 52 samples were used for analysing risk of 3-year recurrence. It appears that the reduction of the recurrence information to a binary endpoint followed the example of van't Veer et al. (2002). The 89 samples were obtained through biopsy of primary tumor at the Koo Foundation Sun Yat-Sen Cancer Center, Taipei, collected and banked between 1991 and 2001.

[1] http://www.isds.duke.edu/~quanli/BinTree_FW.htm

In our reanalysis of the data we concentrate on the recurrence aspect. Recurrence of tumor is of high medical interest. The main part of the data presented by Huang et al. are related to 3-year recurrence. All patients in this group had 1-3 pos. lymph nodes and received the same treatment by resection of the tumor and adjuvant chemotherapy. There are 18 samples of patients suffering a recurrence within three years after surgery, and 34 samples without. Huang et al. concluded from their analysis that they could predict tumor recurrence with misclassification rates of 2/34 for no recurrence within three years and 3/18 for patients with 3-year recurrence.

The authors presented a novel algorithm for classification. A description is available in Pittman et al. (2004) and the algorithm is publicly available as a Windows program. The aim of this compendium is to answer the following questions:

1. Which classification result could be achieved using standard algorithms and is there a difference in classification quality between a standard algorithm and the proposed one?
2. If there is a substantial difference, what is the reason?

The data we based our study on consists of 52 files ("CEL files") each with about 400,000 intensity measurements on probes representing about 12,625 transcripts. The data and the file `Huang1Info.xls`, which includes the available sample information, were downloaded from the website[2] of Duke University.

3.2 Methods

3.2.1 Loading and normalizing the data

We start by loading the necessary libraries.

```
> library(randomForest)
> library(pamr)
> library(e1071)
> library(affy)
> library(xtable)
> library(RColorBrewer)
> library(MCRestimate)
```

[2] http://data.cgt.duke.edu/lancet.php

randomForest (Breiman et al.), *pamr* (Hastie et al.), *e1071* (Dimitriadou et al.), and *MCRestimate* provide implementations of the classification methods. The package *affy* (Gautier et al., 2004) offers tools to handle Affymetrix data, *RColorBrewer* (Neuwirth) provides a range of color maps, and *xtable* (Dahl) has tools for transforming `R` tables into LaTeX code. *MCRestimate* contains the function `MCRestimate`, wrappers for classification algorithms, several preprocessing functions, and the penalized logistic regression algorithm.

We further load the compendium *compHuang*, that consists of an `.Rnw` file used to build this text, and the raw data from Huang et al.

```
> library(compHuang)
```

The patient data is contained in the file `phenodata.RE.csv`, which we manually converted from the Excel table `Huang1Info.xls`. The function `ReadAffy` combines this with the microarray data from the 52 CEL files into an R object.

```
> datadir <- function(x) file.path(.path.package("compHuang"),
+     "extdata", x)
> pd.RE <- read.phenoData(datadir("phenodata.RE.csv"),
+     sep = ";", header = TRUE, row.names = 1)
> find.file <- function(x) file.path(datadir(""),
+     grep(x, dir(datadir("")), value = TRUE))
> filenames.RE <- sapply(pData(pd.RE)[,
+     "Sample"], find.file)
> ab.RE <- ReadAffy(filenames = filenames.RE,
+     phenoData = pd.RE, verbose = TRUE)
```

To obtain normalized expression measures from the microarray data, Huang et al. used Affymetrix' Microarray Suite (MAS) Version 5 software. Additionally, they transformed the data to the logarithmic scale. Here, we use the function `mas5` from the package *affy*, which implements the MAS 5 algorithm. `eset` is an object of class `exprSet`, which comprises the normalized expression values as well as the tumor sample data. All the following analyses are based on this object. By setting the starting point of our analysis at the level of log-transformed MAS5 outputs, we have included all steps that might introduce selection or training bias into our *complete classification procedure.*

```
> eset <- mas5(ab.RE)
> exprs(eset) <- log2(exprs(eset))
```

```
> eset
```

```
Expression Set (exprSet) with
        12625 genes
        52 samples
                 phenoData object with 3 variables and 52 cases
         varLabels
                Sample: Sample ID
                Number.in.figure: No in figures of Huang et al.
                Recurrence: Recurrence yes(=1)/no(=0)
```

3.2.2 Estimation of misclassification rate

We will now apply the processing methods described in section 2.2. The outer cross–validation loop is used to estimate the misclassifation rate. Tibshirani and Efron (2002) recommended using 5- to 10–fold cross-validation in this context. To achieve low variability, we use 10–fold cross-validation, and repeat it 20 times with randomly sampled block constellations. This is controlled by the parameters `c.o` and `c.rep`.

The inner cross-validation loop selects the optimal parameters for a given complete classification procedure. This is implemented in the function `tune` from the package *e1071* (Dimitriadou et al.). We use 5–fold cross-validation to limit the computation time. This is controlled by the parameter `c.i`. The parameter `class.label` is the name of the column of the phenodata which contains the group information for the *complete classification procedure*.

```
> c.o <- 10
> c.i <- 5
> c.rep <- 20
> class.label <- "Recurrence"
```

3.2.3 Preprocessing

Preprocessing steps are often used with the goal of removing noise from the data. Huang et al. first removed all genes with low variability across the samples and a maximum signal intensity below 2^9. The authors do not comment on their stategy, which is encoded in the function `g.red.max.var`.

Second, Huang et al. formed new variables from linear combinations (*metagenes*) by k-means clustering with $k = 496$, and representing each cluster by its first principal component. The function `metagenes.kmeans.svd` is our implementation of this strategy. In contrast to Huang et al., we consider k a parameter of the algorithm, and explore four possible choices (see the *poss.k* object below).

```
> red.fct <- "g.red.max.var"
> mg.fct <- "metagene.kmeans.svd"
> poss.k <- list(number.clusters = c(100,
+     496, 1000, 2500))
```

The *metagene* is not an actual gene, but rather a feature that encompasses much of the discriminatory information in a cluster of genes. We also consider classification procedures that omit the use of *metagenes*. Since the preprocessing is part of the complete classification procedure, it needs to be included in the estimation of the misclassification rate through cross-validation (Ambroise and McLachlan, 2002).

3.3 Results

3.3.1 Classification with metagenes

Random Forest `RF.wrap` is a wrapper for the training and prediction functions from the package *randomForest* (Breiman et al.).

```
> classificationAlgorithm = "RF.wrap"
> r.forest.meta <- MCRestimate(eset,
+     class.label, poss.parameters = poss.k,
+     classification.fun = classificationAlgorithm,
+     cluster.fun = mg.fct, variableSel.fun = red.fct,
+     cross.outer = c.o, cross.inner = c.i,
+     cross.repeat = c.rep)
```

`MCRestimate` return a vote matrix and a confusion table. A vote matrix contains for each sample the number of times it was assigned to each class in the outer cross-validation loop. The confusion table counts the coincidences between true class and consensus class, that for a sample is the class with the highest vote. The information is also presented in a figure, the *vote plot* (Fig. 2).

```
> r.forest.meta

Result of MCRestimate with 20 repetitions of
10-fold cross-validation

Selection function     : g.red.max.var
Cluster function       : metagene.kmeans.svd
```

```
Classification function: RF.wrap

The confusion table:
   0 1 class error
0 31 3       0.088
1 10 8       0.556

> plot(r.forest.meta, main = "Random Forest",
+     rownames.from.object = TRUE, shading = 15,
+     sample.order = TRUE)
```

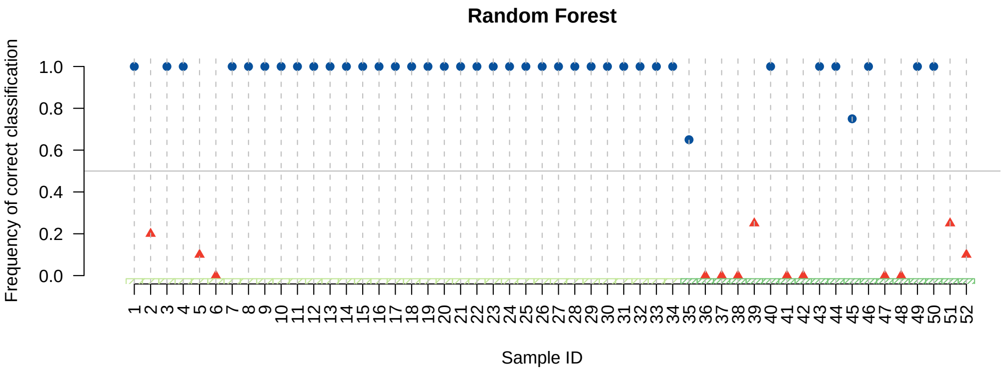


Figure 2: Visualization of the vote matrix. Samples that were misclassified most of the time are plotted in red triagles, the others in blue dots. The horizontal bars (light green with light shading and dark green with dense shading) represent the no recurrence and the recurrence group, respectively.

Class prediction by nearest shrunken centroids `PAM.wrap` is a wrapper for the training and prediction functions in the package *pamr* (Hastie et al.). The method requires a threshold that determines the number of genes selected in the classifier, that is, the amount of shrinkage. We let the function `pamr.train` suggest 10 possible thresholds. We reverse the order of the thresholds because in the case of multiple minima, `tune` chooses the first one, and we prefer to use a classifier with a smaller number of genes. The resulting vote plot is shown together with the plots for the other methods on page 15, page 16, page 17, and page 18 .

```
> classificationAlgorithm <- "PAM.wrap"
> pam <- pamr.train(list(x = exprs(eset),
+     y = pData(eset)[, class.label]),
+     n.threshold = 10)
> thresholds <- rev(pam$threshold)
> poss.pars <- c(list(threshold = thresholds),
+     poss.k)
```

Penalized logistic regression PLR.wrap is a wrapper for the training and prediction functions, which we implemented in the package *MCRestimate*. It requires a penalty parameter κ. We use the following list of possible values. For other applications, it may be necessary to consider larger penalty values.

```
> classificationAlgorithm = "PLR.wrap"
> kappa.range <- 10^seq(0, 5, 0.5)
> poss.pars <- c(list(kappa = kappa.range),
+     poss.k)
```

Support Vector Machines SVM.wrap is a wrapper for the support vector machine training and prediction functions that are implemented in the package *e1071* (Dimitriadou et al.). Here, we use a radial Gaussian kernel, i. e.

$$K(u, v) = \exp\left(-\frac{\gamma\|u - v\|^2}{n}\right) \qquad u, v \in \mathbb{R}^n.$$

The method requires two parameters, γ and the cost of constraints violation. We choose them from the following ranges.

```
> classificationAlgorithm <- "SVM.wrap"
> gamma.range <- 2^(-2:2)
> cost.range <- 2^(-2:2)
> poss.pars <- c(list(gamma = gamma.range,
+     cost = cost.range), poss.k)
```

Bayesian Binary Prediction Tree Models BBT.wrap is a wrapper for the Bayesian Binary Prediction Tree Models. Pittman et al. provide an executable file for MS-Windows operating systems on their website[3]. Since we performed our analysis on a Unix machine, we used the Windows emulator *wine*[4] to execute the program.

[3] http://www.isds.duke.edu/~quanli/BinTree_FW.htm

[4] http://www.winehq.com

```
> classificationAlgorithm <- "BBT.wrap"
```

3.3.2 Classification without metagenes

We redo the same analysis as above without the clustering of genes into metagenes.

```
> mg.fct <- "identity"
> poss.k <- list()
```

3.3.3 Summary

Two summary tables and one heatmap are produced. Tables 1 and 2 summarize the confusion tables. Figure 3 gives a synopsis of the per-patient performance of the different *complete classification procedures*.

	RF-M	PAM-M	PLR-M	SVM-M	BBT-M	Size
No Recurrence	3	2	3	2	1	34
Recurrence	10	9	12	12	9	18
All	13	11	15	14	10	52

Table 1: Overall number of misclassifications for methods with metagenes

	RF	PAM	PLR	SVM	BBT	Size
No Recurrence	2	3	2	2	2	34
Recurrence	12	8	12	12	10	18
All	14	11	14	14	12	52

Table 2: Overall number of misclassifications for methods without metagenes

3.4 Comments and Conclusions

We can now come back to the questions we posed in the introduction. Tables 1 and 2 answer question 1: we obtained misclassification rates of around 25% with all eight methods. The use of metagenes did not seem to make a big difference either way. Most of the misclassified samples come from the group of patients with recurrence, which is the smaller group (Fig. 3). Possibly, this

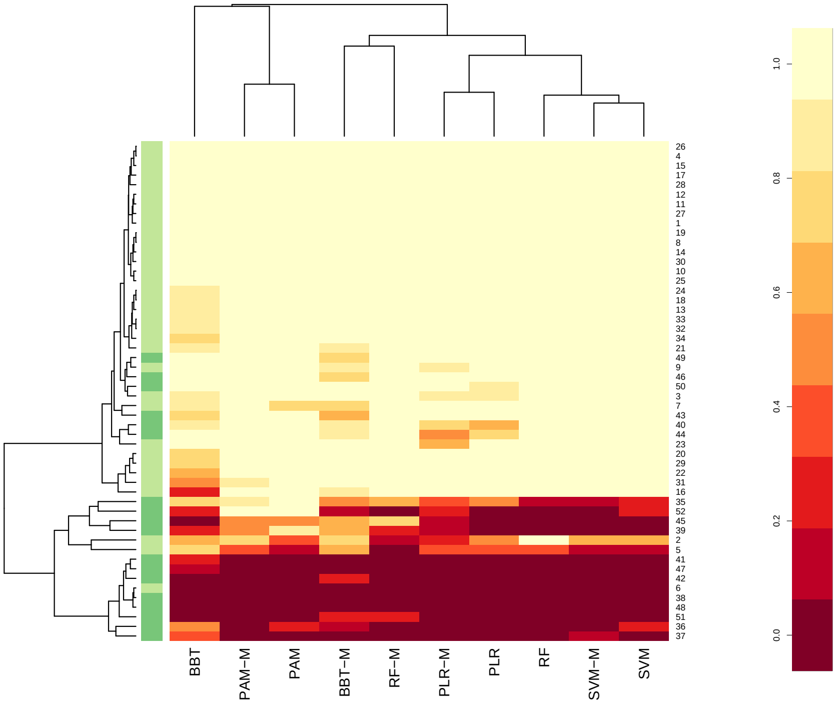


Figure 3: The performance of the different *complete classification procedures* for each sample. Brighter colors indicate samples that have been classified correctly more frequently. The color bar on the left side represents the no recurrence (light green) and the recurrence (dark green) group, respectively.

could be explained by a preference of the classification algorithms to favour the larger group.

There is a difference between the classification accuracy we observed for the Bayesian Binary Prediction Tree Models and that presented by Huang et al. (2003). They reported misclassification rates of 2/34 and 3/18 for patients with and without recurrence, respectively. We observed misclassification rates similar to the ones achieved with the four other methods (compare tables 1 and 2). Regarding question 2, there are a number of possible explanations why there is a difference between our and their estimation of the misclassification rate.

First, their reported estimate of the misclassification rate could be over–optimistic. We tested two possible reasons: the use of leave–one–out cross–validation, and the exclusion of the preprocessing from the cross–validation.

Both can cause overfit and selection bias (Ambroise and McLachlan, 2002). Using such procedures for the present data, however, we only obtained marginal reduction of the estimated MCR.

Second, the estimates of the misclassification rates that we report could be over–pessimistic. To explore this possibility, we applied the method presented here to several other classification problems. For example, the reader may easily check the effectiveness of the tool by applying it to T. Golub's ALL/AML data (Golub et al., 1999). This can be done by using the R commands

```
> library(golubEsets)
> data(golubMerge)
> eset <- golubMerge
> class.label <- "ALL.AML"
> red.fct <- "g.red.highest.var"
```

and then following our calculations as above. The package *golubEsets* is available from the Bioconductor website. With this, for each classification method we found an almost perfect classification rate, comparable to that reported by Golub et al.

Third, the BBT algorithm is quite complex and allows many different choices of parameters. It is possible that the software version with the parameters that we used is different from the one that was used for the Huang et al. paper.

The main feature of a classification algorithm is its ability to derive generalizable structures from a limited set of training data, and to make them useful for the prediction of the class membership of new observations. Since the data is noisy and high–dimensional, it is not of much use to overfit. Rather, some kind of regularization is necessary. But regularization is the answer to an ill–defined problem, and there are many possible answers.

If, for a given classification problem, standard methods produce a satisfactory result, it would not be economical to devise a new, idiosyncratic approach. If they do not, as appears to be the case for Huang et al.'s breast cancer recurrence data, this poses a tough choice: either one has to concede that the current data is insufficient, or, one sets out to develop a new method. But then one has to be aware that this may introduce a different kind of overfitting: that of trying out many different methods and their variations, until one happens to "work".

The question of whether breast cancer recurrence can be accurately predicted from microarray profiles of the primary tumor remains open: current reports that are based on small sample numbers may be over-optimistic; on the other hand there does seem to be a signal with the potential to discriminate. Studies with larger sample numbers, and perhaps with improved accuracies of the measured molecular tumor properties, will be necessary.

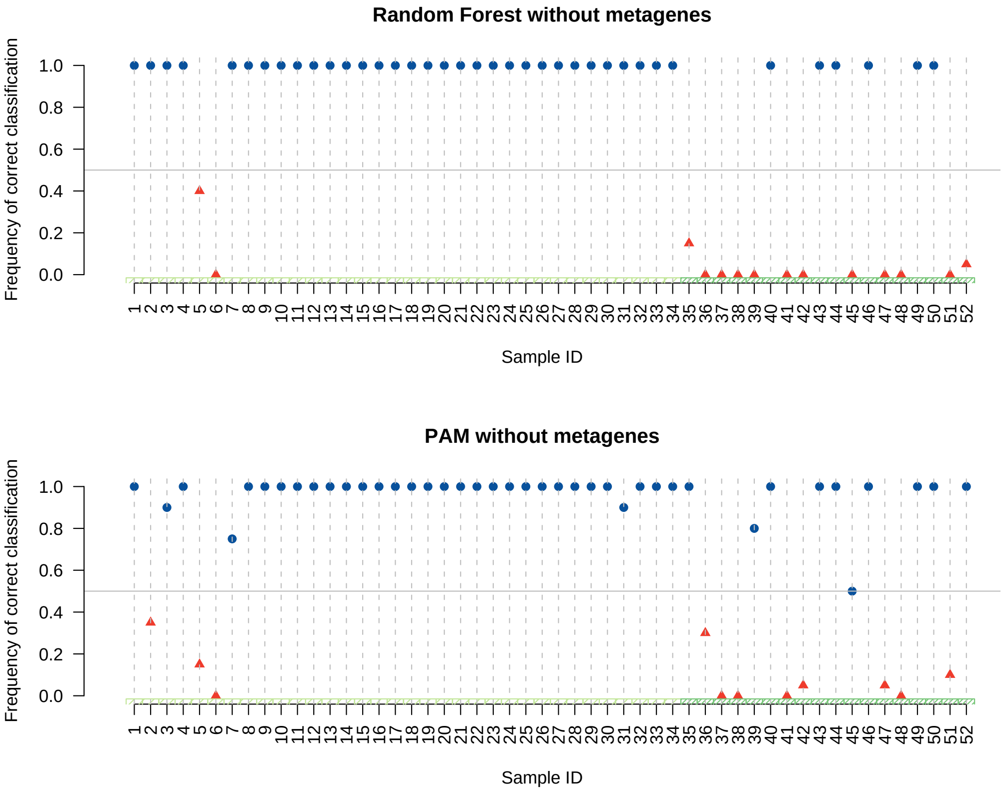
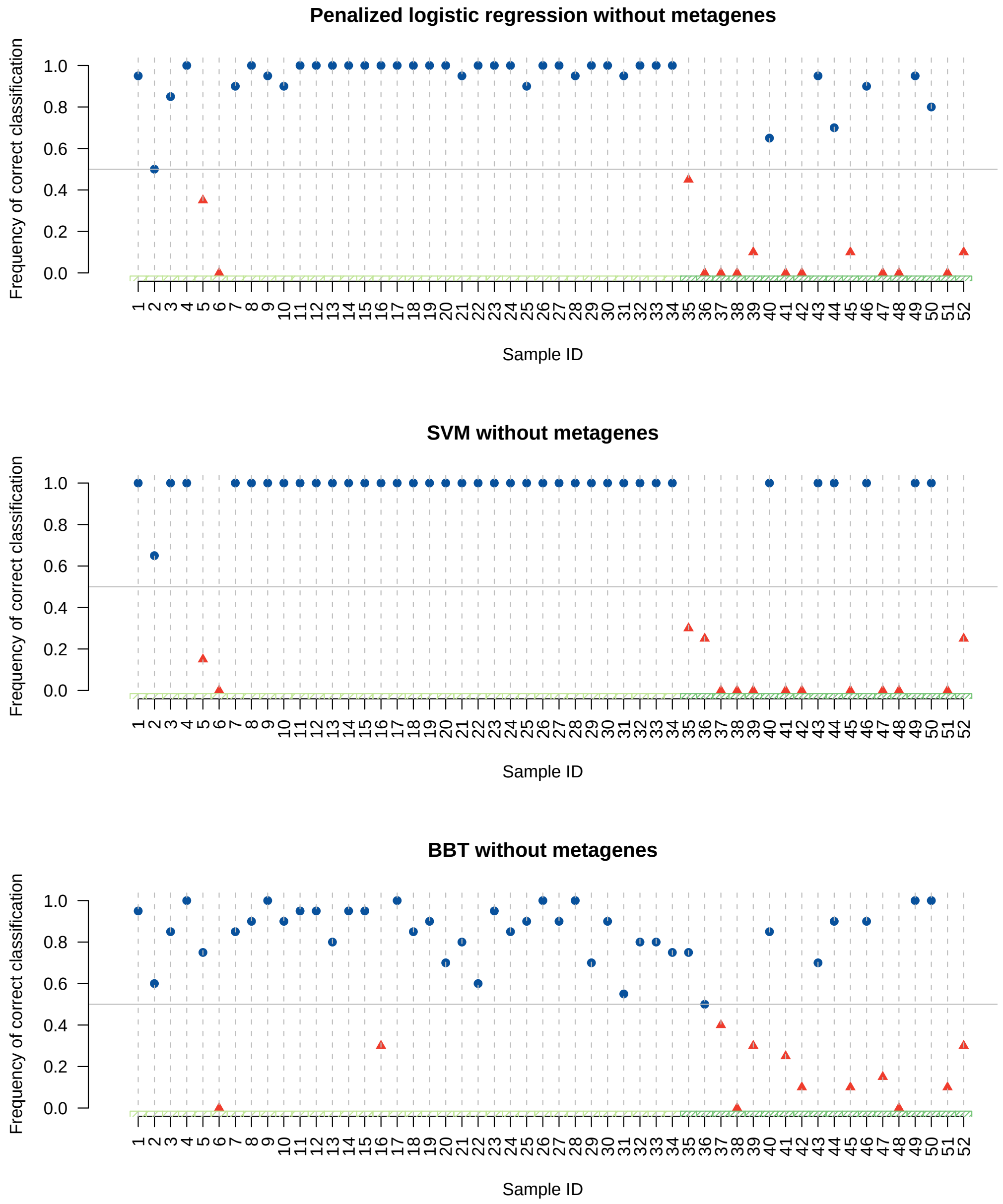


Figure 4: Vote plot for random forest and PAM without metagenes.



Penalized logistic regression without metagenes

SVM without metagenes

BBT without metagenes

Figure 5: Vote plot for penalized logistic regression, SVM, and binary bayesian tree without metagenes.

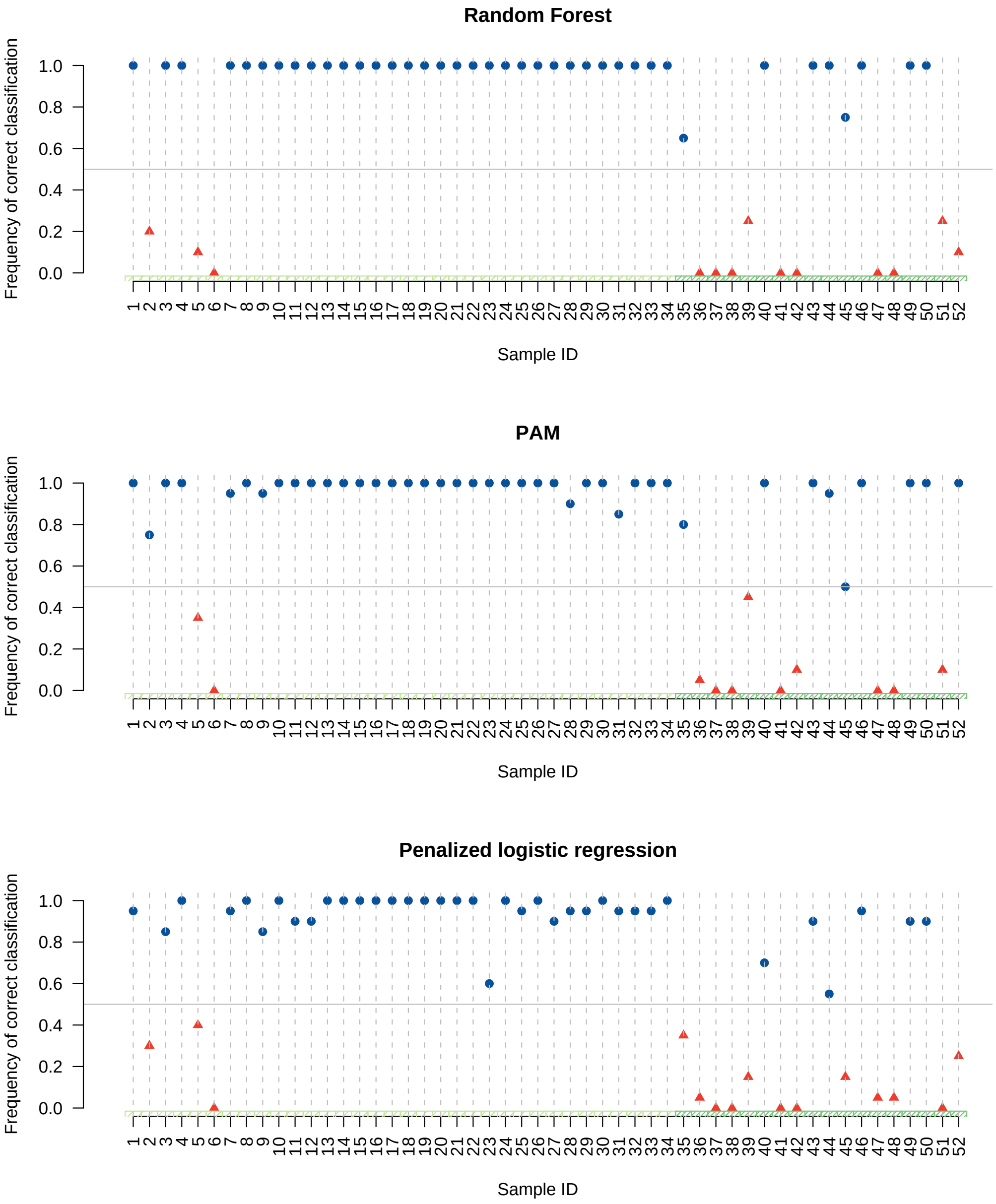


Figure 6: Vote plot for random forest, PAM, and penalized logistic regression with metagenes.

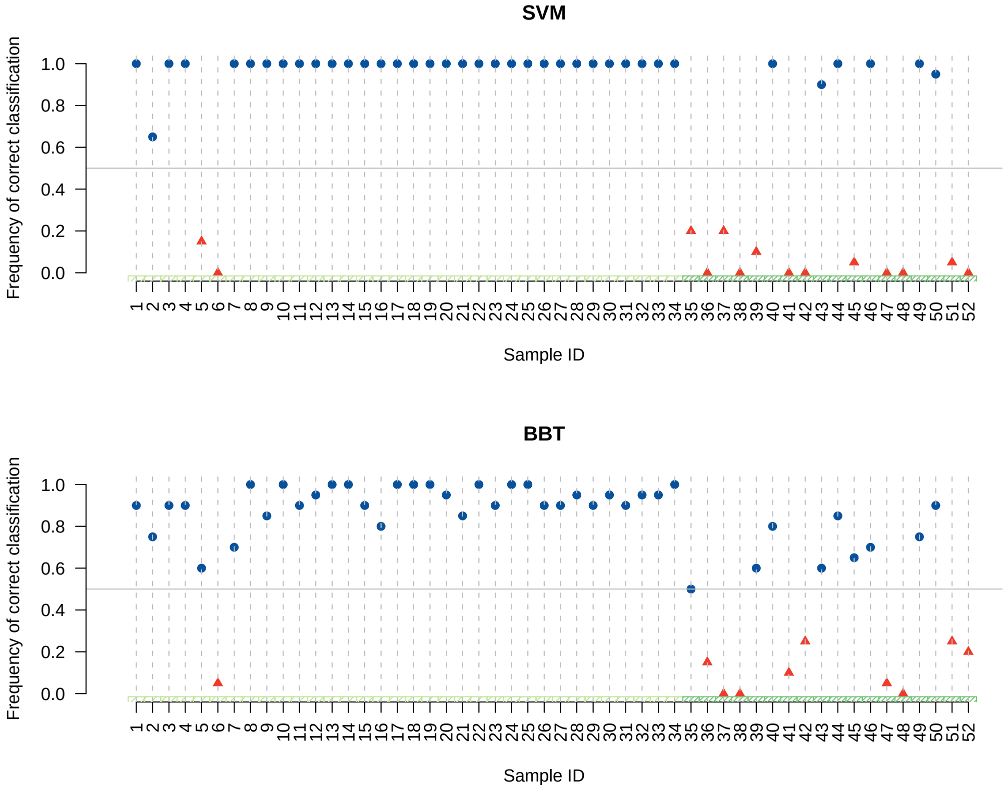


Figure 7: Vote plot for SVM and binary bayesian tree with metagenes.

4 Discussion

A large number of methods for classification have been devised by different communities, including statistics, machine learning, data mining, and pattern recognition. These different communities have approached the problem from rather different perspectives. We tried to take this background into account when choosing the methods offered in our compendium: Penalized logistic regression (PLR), support vector machines (SVM), shrunken centroids (PAM), and random forest (RF). The four methods showed good performance for a wide range of classification problems based on microarray data.

As mentioned in the introduction, the interpretation of molecular signatures entangles the data and the algorithm. We believe that simply depositing the primary data and a software tool in a web supplement or a public database is not enough. The crucial step is deriving the results from raw data. The algorithm for this task is so complex such that only the computer code is an appropriate description. The ability to provide and document programming language code is critical to understandable, explainable, and reproducible research.

As a software platform for the compendium we chose a combination of the statistical environment R/Bioconductor and the document processing tool LaTeX, following Leisch (2002). These systems are freely available, easy to install on every desktop computer and familiar to a large number of users. The Comprehensive R Archive Network (CRAN, cran.r-project.org) offers a large collection of software modules with classification and machine learning algorithms. Other text processing and computing engines could also have been used.

Another approach to explore classification strategies is to endeavour on a zealous computational project which tries to cover the full combinatorial space of all possible classification methods, their parameters, and preprocessing methods for an extensive series of datasets (Dudoit et al., 2002; Liu et al., 2002; Li et al., 2004). Without interpretation, the reader will most likely be left startled how using different data, or changing a particular setting would affect the results. The orientation given by such empirical approaches is necessarily coarse.

Therefore, it may be wise to replace the static by an interactive approach and to offer the machinery which allows the researcher to perform such a study on the algorithms and preprocessing strategies of interest together with relevant data. The compendium offers different levels of interactivity. It can be used to produce a textual output comparable with the static approach. On an intermediate level one interacts with the compendium by specifying parameters for the complete classification procedure and data sets. This is the level of sensitivity analyses or of comparing the performance of implemented algorithms on different data sets.

For example in our compendium one could change the kernel of a support vector machine by simply changing the parameter *poss.pars*

```
> poss.pars <- c(list(cost = cost.range,
+     kernel = "linear"), poss.k)
```

The advanced level of interaction consists in introducing new ideas like new classification algorithms, new preprocessing methods or new tools for the presentation of the results. Writing wrapper functions for new classification methods is simple. The following example shows a wrapper for diagonal discriminant analysis:

```
> DLDA.wrap <- function(x, y, pool = 1,
+     ...) {
+     require(sma)
+     predict.function <- function(testmatrix) {
+         res <- stat.diag.da(ls = x,
+             as.numeric(y), testmatrix,
+             pool = pool)$pred
+         return(levels(y)[res])
+     }
+     return(list(predict = predict.function,
+         info = list()))
+ }
```

The literature on the induction of prognostic profiles from microarray studies is a methodological wasteland. Ambroise and McLachlan (2002) describe the unthorough use of cross-validation in a number of high-profile published microarray studies. Tibshirani and Efron (2002) report the difficulty in reproducing a published analysis, even with some help of the authors. Huang et al. (2003) present results with the potential to revolutionize clinical practice in breast cancer treatment but even after the data and an implemetation of the classification algorithm has been made public is was not possible to reproduce their results. A series of papers published in Nature, NEJM, and The Lancet base their impressive results on classification methods which were developed ad-hoc for the problem at hand.

This situation has several implications: 1) It is nearly impossible to assess the value of the presented studies in terms of statistical quality and clinical impact. 2) Scientists looking for guidance to design similar studies are left puzzled by the plethora of methods. 3) It is left unclear how much potential there is for follow-up studies to incrementally improve on the results. Our compendium offers a first approach to overcome these problems.

For scientists confronted with an idiosyncratic analysis in a new publication it offers the means to thoroughly assess the data. The compendium allows one to check the dependence of the classification result on the choice of the cross-validation parameters, the preprocessing, and the family of classifiers. For example, they can explore the deficiencies of the frequently used leave-one-out cross-validation. Subtle biases introduced by the choice of preprocessing for dimension reduction can be explored. Standard methods can be applied to the data and compared to the reported results. Only if the results are different may the reported new method warrant particular attention. Applying the standard methods will also reveal the inherent difficulty of the classification task.

For a scientist planning a prognostic study on a molecular signature the compendium offers a complete framework for the design, analysis and reporting of the study. To find a good design similar data can be used to explore classification strategies and sample size issues. Importing new data into the compendium generates a reproducible analysis and can be used as a template for a scientific report. Besides publishing the textual output in the traditional way, the scientist can make all details of the analysis available to colleagues.

The third point is of relevance for the development of molecular medicine. The studies published so far were retrospective, and prospective confirmation of the developed classifiers is of eminent clinical importance. Decisions on granting such expensive prospective longterm studies have to be based on reliable planning figures. Publishing results in the traditional paper based way in a journal hides too much information. Compendia can provide the insights needed to plan future projects.

Appendix

Processing the compendium requires the software R version 1.9.1 together with the add-on packages mentioned in Section 3.1 and latex $\geq$ 3.14. These are available for free and as open source, and run on many operating systems, including Windows 32, Linux, Unix, and Mac OS X. For the processing of the CEL files in section 3.1, we recommend to have 512 MB of RAM or more. The rest of the calculations require less memory. The processing time is 7 days on a 2.4 GHz Pentium CPU to perform the analysis without metagenes. The time will increase by a factor of 6 by including the metagenes preprocessing step, and it is directly proportional to the parameter `c.o` an `c.repeat`. It would be straightforward to parallelize the computations.

References

Leisch F and Rossini AJ. Reproducible statistical research. *Chance* 16(2):41-45 (2003).

Gentleman R. Reproducible research: a bioinformatics case study. *Statistical Applications in Genetics and Molecular Biology* Vol. 3 (2004).

van't Veer L, Dai H, van de Vijver MJ, He YD, Hart AAM, Mao M, Peterse HL, van de Kooy K, Marton MJ, Witteveen AT, Schreiber GJ, Kerkhoven RM, Roberts C, Linsley PS, Bernards R, and Friend SH. Gene expression profiling predicts clinical outcome of breast cancer. *Nature* 415:530–536 (2002).

van de Vijver MJ, He YD, van't Veer LJ, et al. A gene-expression signature as a predictor of survival in breast cancer. *N Engl J Med* Vol.347:1999–2009 (2002)

Huang E, Cheng SH, Dressman H, Pittman J, Tsou MH, Horng CF, Bild A, Iversen ES, Liao M, Chen CM, West M, Nevins JR, Huang AT. Gene expression predictors of breast cancer outcomes. *The Lancet* 361:1590–1596 (2003).

Brenton JD and Caldas C. Predictive cancer genomics – what do we need? *The Lancet* 362:340–341 (2003).

Tibshirani RJ and Efron B. Pre-validation and inference in microarrays *Statistical Applications in Genetics and Molecular Biology* Vol.1, Article 1 (2002)

Sawitzki G. Software components and document integration for statistical computing. in *Proceedings ISI Helsinki 1999 (52nd session) Bulletin of the International Statistical Institute* Tome LVIII, Book 2, pp. 117-120.

Leisch F. Dynamic generation of statistical reports using literate data analysis. In: *Compstat 2002 - Proceedings in Computational Statistics* pages 575–580. Physika Verlag, Heidelberg, Germany, 2002. ISBN 3-7908-1517-9.

Gentleman R and Temple Lang D. Statistical analyses and reproducible research *unpublished manuscript*

Sawitzki G. Keeping statistics alive in documents. *Computational Statistics* 17:65–88 (2002).

Ihaka R and Gentleman R. R: a language for data analysis and graphics. *Journal of Computational and Graphical Statistics* 5:299–314 (1996)

Gentleman R and Carey V. Bioconductor. *R News* 2(1):11-16, (2002)

Knuth DE. Literate Programming Center for the Study of Language and Information (1992)

Wolf HP. REVWEB
http://www.wiwi.uni-bielefeld.de/˜wolf/software/revweb/revweb.html

Ambroise C and McLachlan GJ. Selection bias in gene extraction on the basis of microarray gene–expression data. *Proc Natl Acad Sci* 99(10): 6562–6 (2002)

Ransohoff DF. Rules of evidence for cancer molecular-marker discovery and validation. *Nat Rev Cancer* 4(4):309–14 (2004)

Breiman L. Random forests. *Machine Learning Journal* 45:5–32 (2001)

Tibshirani R, Hastie T, Narasimhan B, Chu G. Class prediction by nearest shrunken centroids, with application to DNA microarrays. *Statistical Science* 18:104–117 (2003)

Eilers PH, Boer JM, Van Ommen GJ, Van Houwelingen HC. Classification of microarray data with penalized logistic regression. *Proceedings of SPIE volume 4266: progress in biomedical optics and imaging* 2:187–198 (2001)

Vapnik V. The nature of statistical learning theory. Springer Verlag (1999)

Pittman J, Huang E, Nevins J, Wang Q, West M. Bayesian analysis of binary prediction tree models for retrospectively sampled outcomes. *Biostatistics* 5(4):587–601 (2004)

Breiman L, Cutler A, Liaw A, Wiener M randomForest: Breiman's random forest for classification and regression. Fortran original by Leo Breiman and Adele Cutler and R port by Andy Liaw and Matthew Wiener. *R package* 4.3-3 url = http://stat-www.berkeley.edu/users/breiman/RandomForests

Hastie T, Tibshirani R, Narasimhan B, Chu G. pamr: Pam: prediction analysis for microarrays. *R package* 1.21

Dimitriadou E, Hornik K, Leisch F, Meyer D,Weingessel A. e1071: Misc functions of the department of statistics (e1071), TU Wien. *R package* 1.5-1

Gautier L, Cope L, Bolstad BM, Irizarry RA. affy–analysis of Affymetrix GeneChip data at the probe level. *Bioinformatics* 20(3):307-15 (2004).

Neuwirth E. RColorBrewer: ColorBrewer palettes. *R package* 0.2-1

Dahl DB. with contributions from many others xtable: Export tables to LaTeX or HTML. *R package* 1.2-4

Pepe MS, Longton G, Anderson GL, Schummer M. Selecting differentially expressed genes from microarray experiments. *Biometrics* 59(1):133-42 (2003).

Golub TR, Slonim DK, Tamayo P, Huard C, Gaasenbeek M, Mesirov JP, Coller H, Loh ML, Downing JR, Caligiuri MA, Bloomfield CD, Lander ES. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring *Science* 286(5439): 531-7 (1999).

Dudoit S, Fridlyand J, Speed T. Comparison of discrimination methods for the classification of tumors using gene expression data. *JASA* 97(459):77-87 (2002).

Liu H, Li J, Wong L. A comparative study on feature selection and classification methods using gene expression profiles and proteomic patterns. *Genome Informatics* 13:51-60 (2002).

Li T, Zhang C, Ogihara M. A comparative study on feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics* 15 Apr (2004).