

Molecular Diagnosis

Classification, Model Selection and Performance Evaluation

F. Markowitz, R. Spang

Computational Diagnostics Group, MPI for Molecular Genetics, Berlin, Germany

Summary

Objectives: We discuss supervised classification techniques applied to medical diagnosis based on gene expression profiles. Our focus lies on strategies of adaptive model selection to avoid overfitting in high-dimensional spaces.

Methods: We introduce likelihood-based methods, classification trees, support vector machines and regularized binary regression. For regularization by dimension reduction, we describe feature selection methods: feature filtering, feature shrinkage and wrapper approaches. In small sample-size situations efficient methods of data re-use are needed to assess the predictive power of a model. We discuss two issues in using cross-validation: the difference between in-loop and out-of-loop feature selection, and estimating model parameters in nested-loop cross-validation.

Results: Gene selection does not reduce the dimensionality of the model. Tuning parameters enable adaptive model selection. The feature selection bias is a common pitfall in performance evaluation. Model selection and performance evaluation can be combined by nested-loop cross-validation.

Conclusions: Classification of microarrays is prone to overfitting. A rigorous and unbiased assessment of the predictive power of the model is a must.

Keywords

Microarrays, statistical classification, generalization error, model assessment, gene selection

Methods Inf Med 2005; 44: 438–43

Introduction

Microarrays provide a bird's eye view of molecular mechanisms in the cell. The promise to find molecular disease signatures from this perspective makes supervised analysis of microarray data a very active field of research. Recent studies have demonstrated high potential of microarrays for the diagnosis of tumor entities [1], risk group determination [2], and the prediction of response to treatment [3]. The dimensionality of the data is always much larger than the number of data points: several thousand genes are measured in – at best – a few hundred experiments. This high dimensionality challenges statistical methods and calls for statisticians to develop methods adapted to the situation of having more variables than samples. In this introductory article we focus on basic tasks and problems of classification (sometimes also called discrimination or class prediction) in microarray analysis and illustrate them by methods that have proven well in many applications.

Goals and Methods

Classification Belongs to the Realm of Supervised Learning

An expression profile measured by a microarray is treated as a realization of a random vector $X \in \mathcal{R}^p$, where p is the number of genes, and the phenotypic class label as a binary random variable $Y \in \{-1, +1\}$. We will only discuss binary classification, but methods are easily extended to Y taking more than two values. In typical applications Y encodes diagnostic entities (e.g. different types of cancer), disease outcome

(e.g. benign and malignant tumor) or patient response to a certain treatment. The existence of the response variable Y makes classification different from unsupervised approaches like clustering. There is a wide range of general textbooks on classification; prominent examples are books by Hastie et al. [4], Duda et al. [5], Ripley [6] and Bishop [7]

The Objective Is Good Performance on Future Data

The statistical task is to find a model f relating to X to Y . Since the application is diagnosis, all models need to be predictive. Formally, this means that we want to find a model f , which minimizes the expected prediction error

$$R[f] = E(L(Y, f(X))) = \int L(Y, f(X)) dP(X, Y). \quad (1)$$

The integral is called the *risk functional* [8]. It averages the loss function $L(Y, f(X))$ over the joint distribution of X and Y . A typical choice of the loss function for classification is the 0/1-loss $L(Y, f(X)) = 1/2 \cdot |Y - f(X)|$, which is 0 if the prediction is true and 1 otherwise and thus counts the number of misclassifications. The joint distribution $P(X, Y)$ constitutes information over the whole “population” of patients or tissue samples. In general the integral in Eq. 1 cannot be evaluated, since we do not know the joint distribution. We only have access to training data sampled from the population. The data constitute a set $T = \{(x_1, y_1), \dots, (x_N, y_N)\}$ of N realizations of the pair (X, Y) . The learning set T is used to fit a model f . In the following we shortly describe maximum likelihood classification rules, classification trees, support vector machines and regularized binary regression.

Maximum Likelihood Discrimination Rules

A very well studied family of classification methods is based on comparison of multivariate Gaussian likelihoods for the two classes [4, 9]. These methods model the conditional density $P(x|y=c)$ of the data given membership to class $c \in \{-1, +1\}$ as a multivariate normal distribution $N(\mu_c, \Sigma_c)$ with class mean μ_c and covariance matrix Σ_c . The two means and covariance matrices are estimated from the training data; a new point is classified to the class for which it is most likely. Restrictions on the form of the covariance matrix control model complexity: Quadratic discriminant analysis (QDA) allows different covariance matrices in both classes; linear discriminant analysis (LDA) assumes that they are the same, and diagonal linear discriminant analysis (DLDA) additionally restricts the covariance matrix to diagonal form.

Classification Trees

To construct binary tree structured classifiers the measurement space \mathcal{X}^p repeatedly has to be split into subsets. Each split corresponds to a node in the tree. The main statistical issues in tree construction are where to split, when to stop and how to assign a label to a terminal node. An often-used approach called CART (for *Classification and Regression Trees*) is described by Breiman et al. [10]. Trees are easy to interpret but tend to be unstable and lacking in accuracy. Their performance can be greatly enhanced by introducing a random element in the construction and aggregating many resulting trees. Methods implementing this are bagged trees [11], boosted trees [12] and random forests [13]. In a comparative study by Dudoit et al. [9] the simple DLDA performs remarkably well compared to more sophisticated methods like aggregated trees.

Support Vector Machines

Support vector machines (SVMs [8, 14, 15]) were used successfully in many appli-

cations (e.g. [16, 17]). SVMs consist of two building blocks: First, they construct a *maximal margin hyperplane*, which trades off the number of misclassifications with the distance between hyperplane and nearest data-points in the training set. The maximal margin hyperplane can be constructed by means of inner products $x_i^T x_j$ between pairs of training examples x_i and x_j . This observation is the key to the second building block of SVMs: the inner product $x_i^T x_j$ is substituted by a non-linear *kernel function*. Typical choices are polynomial kernels $k(x_i, x_j) = (x_i^T x_j)^d$ and radial basis function kernels

$$k(x_i, x_j) = \exp\left(-\frac{1}{2\sigma^2} \|x_i - x_j\|^2\right),$$

where d and σ are parameters chosen by the user. The use of kernel functions implicitly maps the data into a high-dimensional space, in which the maximal margin hyperplane is then constructed [11].

Regularized Binary Regression

With more variables than samples, linear binary regression models need to be augmented by regularizing terms to avoid saturation. Prominent examples are penalized maximum likelihood regression [18] and the LASSO [19]. In a Bayesian framework, regularization can be naturally implemented by using informative priors on the model parameters [20]. In the context of microarray analysis, this approach has been followed by Roth [21], Krishnapuram et al. [22], West et al. [23] and Spang et al. [24]. Spang et al. [24] in addition describe how full posterior distributions of classification probabilities can be used to uncover conflicts in the data with respect to the classification of some of the patients, highlighting them as critical cases for which additional investigations are appropriate.

Diagnosis Consists of Three Parts: Classification, Model Selection, and Performance Evaluation

How do we find an accurate model f from the training examples T ? The answer to this

question involves the three steps training, model selection and testing or validation [4, 25]: Several models are fitted to the data. We assess the predictive power of each model to choose the optimal one, and then we estimate its generalization error on future data. Ideally, the three steps are reflected in a threefold splitting of the dataset T : a big subset of the data is used for training and two smaller subsets for model selection and testing. If there is insufficient data to split it into three parts, the training and selection set are united, or – in the worst case – even the independent test set is abandoned. In the last two situations efficient sample reuse by cross-validation or the bootstrap is needed to estimate generalization ability [4]. We describe cross-validation in greater detail in the section on *Adaptive Model Selection and Assessment by Cross-validation* but first shift attention to the peculiarities of classifying in a $p \ll N$ situation.

Classification in High Dimensions

The Major Problem Is Overfitting

Given the dataset T we can approximate the integral in Eq. 1 by the *training error*, also called *empirical risk* [8],

$$R_{emp}[f] = \frac{1}{N} \sum_{i=1}^N L(y_i, f(x_i)). \quad (2)$$

The mean loss on the training set is used to approximate the expected loss on the whole population. Often the decision rule optimal on the training data is only suboptimal for the test data. This problem is called *overfitting* [4]. Because of sample variance, the distribution of data in the training set differs from the one in the test set. This leads to an increased number of misclassifications on the test data when using the optimal decision rule established on the training data. When results of multivariate genomic analyses do not generalize to novel data [26], overfitting is one of the main reasons.

The gap between training and test error depends on the complexity of the model. We illustrate this in Figure 1 by a comparison of quadratic discriminant analysis (QDA), lin-

ear discriminant analysis (LDA) and diagonal linear discriminant analysis (DLDA) on simulated data with an increasing number of variables. The effect of overfitting can be seen clearly by a simple simulation: we sample data from two Normal distributions, with parameters sampled from a Normal-inverse-Wishart distribution [27]. This is no attempt to model the complexity and dimensionality of real microarray data, but serves our purpose to illustrate overfitting. Models involving more variables are more complex and the estimated model parameters have higher sample variances. Overfitting is reflected by the gap between test error and training error, which becomes wider for larger numbers of genes and is in general biggest in QDA, smaller in LDA and smallest in DLDA. The simpler the model, the better training error and test error agree. However, little overfitting does not guarantee good predictive performance. Models can also be too simple to capture the relationship between X and Y . In the example, the intermediate model LDA on average achieves the lowest prediction error on the test set.

Model Complexity Increases with Dimensionality

The complexity of classifiers increases with the dimension of the space the data live in.

In p dimensions a simple linear classifier can shatter $p + 1$ points in general position, that is, the classifier can perfectly separate the points, no matter how we assign a binary label to them [4, 15]. This observation is of special importance for medical diagnosis using microarrays. If the number of patients is less than the number of genes (which will be the general case in microarray analysis) we will always be able to perfectly separate the dataset. Even if we randomly permute the labels and again learn a linear classifier, a training error of zero can always be reached. This is a manifestation of an overfitting disaster. Perfect separation does not necessarily indicate a biological relationship of expression values and clinical phenotypes. It can be an artifact of high-dimensional data. Zero prediction error on the training data does not guarantee high generalization ability when applying the model to a new specimen.

Model Complexity Needs to Be Controlled

To reach high generalization ability we have to trade off training error against model complexity. A simple but stable model with small training error will often generalize better than a highly complex classifier with zero training error. The current gold stan-

dard in microarray analysis are methods of *adaptive* model selection where the influence of a penalty for model complexity can be regulated by an additional tuning parameter, which is chosen data-dependently. An example is restricting models to use only a small number of genes, as discussed in the next paragraphs. In support vector machines, regularization is directly implemented by controlling the width of the margin of separation. The tuning parameter is the error weight C , which trades off training error against margin width. Since microarray data will generally be linear separable, increasing complexity by using non-linear kernels is usually not needed. Thus, nonlinear kernels should only be used with care.

The Most Widely Used Approach to Complexity Reduction Is Gene Selection

A straightforward way to enforce simple models is to require that they only depend on a small subset of variables. For high-dimensional data, searching for a subset of size k , maximizing prediction accuracy is not feasible. Overviews of feature selection methods can be found in review articles by Blum and Langley [28], Kohavi and John

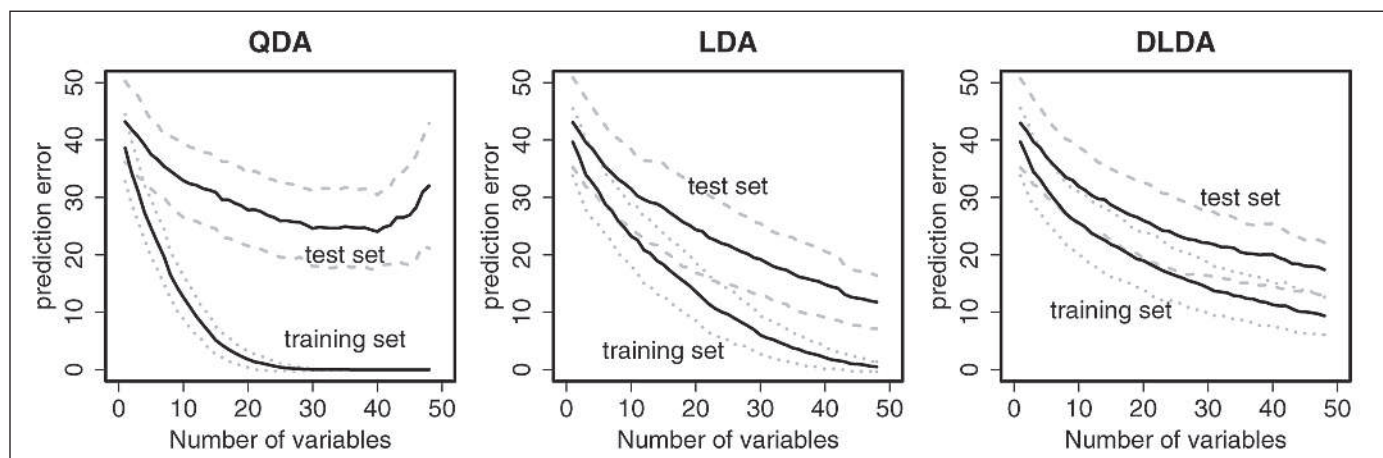


Fig. 1 The gap between training error and test error increases with the number of variables and with the complexity of the classification model. We see this effect for quadratic discriminant analysis (QDA), linear discriminant analysis (LDA) and diagonal linear discriminant analysis (DLDA). In all three plots, the upper curve is the error on the test set, the lower curve the error on the training set. The errors shown are averages over 100 simulations; the

gray dotted lines correspond to mean plus/minus one standard deviation. In each simulation, we sampled independent datasets of size 100 for training and testing, which were equally split into two classes. In each class, samples are drawn from a Normal distribution, with parameters sampled from a Normal-inverse-Wishart distribution. On average, the best result on the test set is achieved by LDA.

[29], and more recently by Guyon and Elisseeff [30]. Common gene-wise selection criteria are feature filtering and shrinking. In *feature filtering*, genes are ranked according to discriminative power using, e.g., the t -statistic or Wilcoxon-statistic. The k top ranking genes are then used for classification; the others are discarded. Filtering uses a hard threshold: Gene $k + 1$ is thrown away even if it bears as much information as gene k . *Feature shrinkage* is a smoother, continuous, soft thresholding method. An application to nearest centroid classification is demonstrated by Tibshirani et al. [31, 32]. The method is a variant of DLDA, in which the class centroids are shrunken in the direction of the overall centroid. For each gene g , the value δ_{gc} measures the distance of the centroid for class c to the overall centroid in units of standard deviation. Each δ_{gc} is then reduced by an amount Δ in absolute value and is set to zero if its absolute value is less than zero. With increasing Δ all genes lose discriminative power and more and more will fade away. Genes with high variance vanish faster than genes with low variance.

Wrapping Is a Powerful Alternative for Selecting Genes

Both feature filtering and feature shrinkage act on single genes. They do not take interactions between genes into account. Clustering genes first and performing feature selection on gene clusters leads to improved performance [33]. Feature filtering is usually done by a criterion different from the one used for classification. This problem is mended in wrapper approaches for feature selection, where the feature selection is “wrapped around” the classification method [29]: the gene selection mechanism uses the classification method to evaluate feature subsets. Guyon et al. [34] use a backward selection procedure called *recursive feature elimination* (RFE) to choose genes with biggest weight in the normal vector of the separating hyperplane constructed by a support vector machine. *Embedded* methods do feature selection while training the classifier; see Krishnapuram et al. [22] or Hochreiter and Obermayer

[35] for recent advances for kernel methods like SVM.

Gene Selection Does not Reduce the Dimensionality of the Model

Classification on a reduced gene set is a two-step procedure: Start with thousands of genes, select an informative subset and use it for classification. One could think that after selecting a small number of genes, all problems due to the high dimensionality of the data have vanished. But LDA on 10 genes selected out of 10,000 is still a model in 10,000 dimensions: The selection process cannot be separated from the model. With thousands of genes and a low signal to noise ratio, we will find many genes with high t -value by chance, the discriminative power of the top genes may be spurious. The generalization error of a discrimination rule based on these genes will be high. Figure 2 demonstrates this point. The number of variables in the datasets ranges from 20 to 10,000, only five of them discriminate be-

tween the two classes, the rest is noise. We select the ten variables with highest t -value on the training set and use them for the prediction of the test set. Although the decision function depends on ten variables in all cases, Figure 2 shows: If these ten variables are chosen from an increasing pool of variables the training error diminishes and the gap between training and test error rises. The generalization properties of the models are different, even though they all use the same small number of variables.

Adaptive Model Selection and Assessment by Cross-validation

Tuning Parameters Enable Adaptive Model Selection

The methods presented in the last section are adaptive in the sense that they depend on a parameter to tune how strong simplicity of

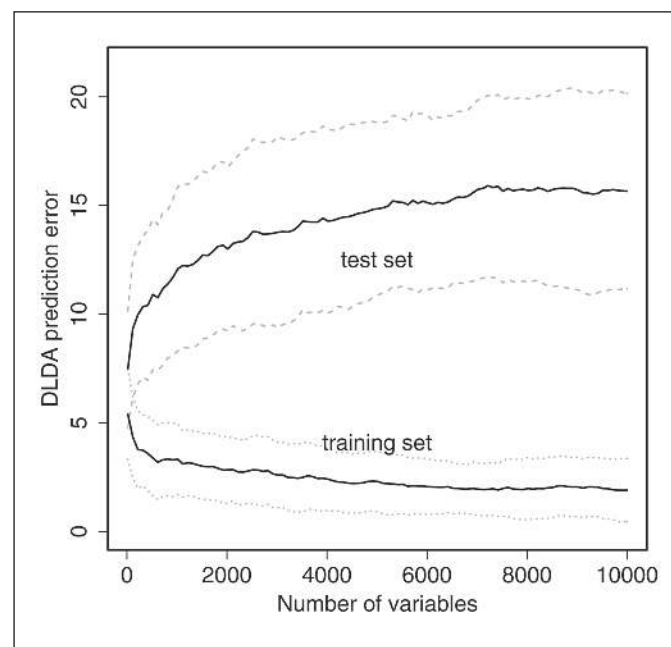


Fig. 2 Training and test error for DLDA classification models based on ten variables with highest t -score selected from an increasing number of variables. The simulated data consists of 100 samples for training and testing, equally divided into two classes. Five variables were sampled from for the first class and for the second class; the other variables are sampled from in all cases and constitute noise. The results are averages over 100 repetitions of data generation, training and testing. The gray dotted lines indicate mean plus/minus one standard deviation. The error on the training set decreases, while the error on the test set increases with the number of noise variables.

the model is enforced. In soft thresholding it is the amount of shrinkage, Δ , in gene selection it is the number of selected genes k , and in SVM it is the regularization parameter controlling the trade-off between margin-width and number of training errors. For SVM a recent result by Hastie et al. [36] allows to fit the entire path of solutions for every value of the regularization parameter with only a little more computational cost than fitting a single SVM model. But usually, these tuning parameters are chosen adaptively from the training data. A simple and widely used way is cross-validation [4]: In each of n steps, a subset of N/n samples is held back to assess the predictive power of a model trained on the remaining $N - N/n$ samples. The tuning parameter with the best cross-validation performance is then chosen. A reasonable choice in the number of steps is $n = 10$ [37].

The Feature Selection Bias Is a Common Pitfall in Performance Evaluation

Evaluating prediction error is not trivial. There exist several possible pitfalls leading to overoptimistic estimations like using too small or unbalanced validation sets [38]. A prominent problem is to introduce selection bias when combining cross-validation with feature selection [23, 37, 38]. There are two possible ways to do the combination: Either apply feature selection to the complete dataset and then perform cross-validation on the reduced data, or perform feature selection in every single step of cross-validation anew. We will call the first alternative *out-of-loop* feature selection and the second *in-loop* feature selection, other names used are *external cross validation* versus *internal cross validation*. Which of them gives the better estimate of generalization performance? The answer is: in-loop feature selection. The out-of-loop procedure is overoptimistic and will be biased downwards. The genes selected for discriminative power on the whole dataset bear information on the samples used for testing in each CV step. The test set is no longer independent of the training set. In-loop feature selection avoids this problem.

The genes are only selected on the training data of each step and the test set remains independent. The problem with in-loop feature selection also occurs in similar form when the generalization error is estimated via the bootstrap. To avoid a downwards-biased estimator, feature selection has to be done on each bootstrapped training-set anew. For model assessment, feature selection and classification cannot be separated.

Model Selection and Performance Evaluation Can Be Combined by Nested-loop Cross-validation

A common way of model selection is to choose the parameter with the smallest cross-validation error. This is an efficient way of sample re-use and in practice usually achieves good results. But often the cross-validation error of the chosen model is also used for performance evaluation. The result will be overoptimistic: because the cross-validation error was subject to minimization, it is biased downwards. A better way to combine model selection and performance evaluation is by *nested-loop cross-validation*. In an outer loop, data is repeatedly split into subsets for training and testing. On each training set we choose the model parameter with minimal cross-validation error. The best model is then tested on the independent test set. The results for all test sets are averaged to gain an estimate of generalization error. Overall, the method consists in two nested cross-validation loops: An outer one for testing, an inner one for choosing parameters. Not only feature selection, but also selecting model parameters has to be done inside the cross-validation loop to avoid underestimation of the generalization error.

Conclusions

We described different methods of classification and strategies of gene selection as means to avoid overfitting in microarray data. For tuning the parameters in adaptive model selection we described a nested

cross-validation scheme and in-loop feature selection.

Software

Most of the techniques we described can be found in packages contributed to the statistical computing environment R [39] and are available at <http://cran.R-project.org>. LDA, QDA and many other classification methods are implemented in package *V/R*; DLDA in package *sma*. Support vector machines are part of package *e1071*; the entire regularization path for the SVM can be computed by *svmpath*. The nearest centroid method is implemented in package *pamr*. Package *randomForests* contains software for random forests. In our simulation, we sampled from the Wishart distribution using package *MCMCpack*. Recursive feature elimination is implemented in the R-package *rfe* available at <http://www.hds.utc.fr/~ambroise/software/RFE>.

Good Statistical Practice in Microarray Studies

Classification in microarray studies is still a controversial field. Our recommendations are:

1. Do not be fooled by small training errors. Classification of microarrays is prone to overfitting and your model might be biologically meaningless and could break down on future data.
2. The key to predictive models is regularization. Gene filtering is the most widely used approach. But wrappers are good alternatives.
3. Be aware that classification on the top 10 genes out of thousands does not ban the curse of dimensionality.
4. A tuning parameter for model selection is advantageous. It can be calibrated adaptively using (nested) cross validation.
5. A rigorous and unbiased assessment of the predictive power of the model is a must. Consider feature selection as part of the classification method. Don't cheat yourself by selecting informative genes globally outside the cross-validation loop.

An important *caveat* at the end: Statistical classification techniques refine medial diagnosis. But genes, which allow high classification accuracy, are not necessarily the ones causing the disease. The main effects we observe on microarrays will often be secondary or tertiary ones. What we see is the avalanche, not the little pebble causing it. The induction from discriminative power of genes to biological importance is misleading in the vast majority of cases.

Acknowledgments

The authors acknowledge support by BMBF grants 031U109C and 03U117.

References

- Yeoh EJ, Ross ME, Shurtleff SA, Williams WK, Patel D, Mahfouz R, Behm FG, Raimondi SC, Relling MV, Patel A, Cheng C, Campana D, Wilkins D, Zhou X, Li J, Liu H, Pui CH, Evans WE, Naeve C, Wong L, Downing JR. Classification, subtype discovery, and prediction of outcome in prediagnostic acute lymphoblastic leukemia by gene expression profiling. *Cancer Cell* 2002; 1 (2): 133-43.
- Huang E, West M, Nevins J. Gene expression profiling for prediction of clinical characteristics of breast cancer. *Recent Prog Horm Res* 2003; 58: 55-73.
- Cheok MH, Yang W, Pui CH, Downing JR, Cheng C, Naeve CW, Relling MV, Evans WE. Treatment-specific changes in gene expression discriminate in vivo drug response in human leukemia cells. *Nat Genet* 2003; 34 (1): 85-90.
- Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning*. New York: Springer; 2001.
- Duda RO, Hart PE, Stork DG. *Pattern classification*. New York: Wiley; second edition, 2001.
- Ripley BD. *Pattern recognition and neural networks*. Cambridge: Cambridge University Press; 1996.
- Bishop CM. *Neural networks for pattern classification*. Oxford: Clarendon Press; 1995.
- Schölkopf B, Smola AJ. *Learning with Kernels*. Cambridge (MA): MIT Press; 2001.
- Dudoit S, Fridlyand J, Speed TP. Comparison of discrimination methods for the classification of tumors using gene expression data. *J Am Stat Assoc* 2002; 97: 77-87.
- Breiman L, Friedman J, Olshen R, Stone C. *Classification and Regression Trees*. Wadsworth, 1984.
- Breiman L. Bagging predictors. *Machine Learning* 1996; 24 (2): 123-40.
- Breiman L. Arcing classifiers. *The Annals of Statistics* 1998; 26 (3): 801-49.
- Breiman L. Random Forests. *Machine Learning* 2001; 45 (1): 5-32.
- Vapnik V. *The Nature of Statistical Learning Theory*. New York: Springer; 1995.
- Vapnik V. *Statistical Learning Theory*. New York: Wiley; 1998.
- Furey TS, Christianini N, Duffy N, Bednarski DW, Schummer M, Haussler D. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 2000; 16 (10): 906-14.
- Ramaswamy S, Tamayo P, Rifkin R, Mukherjee S, Yeang CH, Angelo M, Ladd C, Reich M, Latulippe E, Mesirov JP, Poggio T, Gerald W, Loda M, Lander ES, Golub TR. Multiclass cancer diagnosis using tumor gene expression signatures. *Proc Natl Acad Sci USA* 2001; 98 (26): 15149-54.
- 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* 2001; 2: 187-98.
- Tibshirani R. Regression shrinkage and selection via the lasso. *J Royal Statist Soc B* 1996; 58 (1): 267-88.
- Johnson VE, Albert JH. *Ordinal data modeling*. New York: Springer; 1999.
- Roth V. The Generalized LASSO. *IEEE Transactions on Neural Networks* 2004; 15 (1).
- Krishnapuram B, Carin L, Hartemink A. Gene Expression Analysis: Joint Feature Selection and Classifier Design. In: Schölkopf B, Tsuda K, Vert JP (eds). *Kernel methods in Computational Biology*. Cambridge MA: MIT Press; 2004.
- West M, Blanchette C, Dressman H, Huang E, Ishida S, Spang R, Zuzan H, Olson JA Jr, Marks JR, Nevins JR. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc Natl Acad Sci USA* 2001; 98 (20): 11462-7.
- Spang R, Blanchette C, Zuzan H, Marks JR, Nevins J, West M. Prediction and uncertainty in the analysis of gene expression profiles. *In Silico Biol* 2002; 2 (3): 369-81.
- Altman DG, Royston R. What do we mean by validating a prognostic model? *Stat Med* 2000; 19 (4): 453-73.
- Ransohoff DF. Rules of evidence for cancer molecular-marker discovery and validation. *Nat Rev Cancer* 2004; 4 (4): 309-14.
- Gelman A, Carlin JB, Stern HS, Rubin DB. *Bayesian Data Analysis*. Chapman & Hall/CRC, second edition, 2003.
- Blum A, Langley P. Selection of relevant features and examples in machine learning. *Artificial Intelligence* 1997; 97: 245-71.
- Kohavi R, John GH. Wrappers for feature subset selection. *Artificial Intelligence* 1997; 97 (1-2): 273-324.
- Guyon I, Elisseeff A. An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* 2003; 3 (Mar): 1157-82.
- Tibshirani R, Hastie T, Narasimhan B, Chu G. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc Natl Acad Sci USA* 2002; 99 (10): 6567-72.
- Tibshirani R, Hastie T, Narasimhan B, Chu G. Class Prediction by Nearest Shrunken Centroids, with Applications to DNA Microarrays. *Statistical Science* 2003, 18 (1): 104-17.
- Jäger J, Sengupta R, Ruzzo WL. Improved Gene Selection for Classification of Microarrays. In: *Proc Pacific Symposium on Biocomputing* 2003; 53-64.
- Guyon I, Weston J, Barnhill S, Vapnik V. Gene Selection for Cancer Classification using Support Vector Machines. *Mach Learn* 2002; 46: 389-422.
- Hochreiter S, Obermayer K. Gene Selection for Microarray Data. In: Schölkopf B, Tsuda K, Vert JP (eds). *Kernel methods in Computational Biology*. Cambridge MA: MIT Press; 2004.
- Hastie T, Rosset S, Tibshirani R, Zhu J. The entire regularization path for the support vector machine. *Journal of Machine Learning Research* 2004; 5: 1391-1415.
- Ambrose C, McLachlan GJ. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc Natl Acad Sci USA* 2002; 99 (10): 6562-6.
- Simon R, Radmacher MD, Dobbin K, McShane LM. Pitfalls in the Use of DNA Microarray Data for Diagnostic and Prognostic Classification. *Journal of the National Cancer Institute* 2003; 95 (1).
- R Development Core Team (2004). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. URL: <http://www.R-project.org>.

Correspondence to:

Florian Markowetz
Max Planck Institute for Molecular Genetics
Computational Diagnostics Group
Ihnestrasse 63-73
14195 Berlin, Germany
E-mail: florian.markowetz@molgen.mpg.de