Supervised Analysis When the Number of Candidate Features (p) Greatly Exceeds The Number of Cases (n)

Richard Simon
National Cancer Institute
9000 Rockville Pike, MSC 7434
Bethesda, MD 20892
rsimon@nih.gov
301.496-0975

Abstract
New genomic and proteomic technologies provide measurements of thousands of features for each case. This provides a context for enhanced discovery and false discovery. Most statistical and machine learning procedures were not developed for the p>>n setting and the literature of DNA microarray studies contains many examples of mis-use of analytic and computational methods such a cross-validation. This paper highlights some of key aspects of p>>n problems for identifying informative features and developing accurate classifiers.

Keywords

Prediction, classification, cross-validation

1. Introduction
New technologies for the analysis of biological samples provide information on a genomic scale. For example, DNA microarrays can provide an estimate of the level of abundance of messenger RNA transcripts for all genes of the organism. Technologies for estimating the abundance of thousands of proteins and the copy number of all genes are becoming available. Biologists are utilizing these new technologies for a wide variety of objectives. Contrary to popular belief, effective utilization of these technologies depends on having clear objectives, effective experimental designs and appropriate analysis plans. With genomic or proteomic technologies, the objectives are not gene or protein specific mechanistic hypotheses, as is often the case in other biological investigations. Nevertheless, clear objectives are important and effective investigations rarely represent unstructured searching for interesting patterns in a data archive.

Three frequently occurring kinds of objectives in DNA microarray investigations have been called *class comparison*, *class prediction* and *class discovery* [18]. Class comparison involves identifying differentially expressed in cells from different types of tissue, different kinds of patients, or in cells exposed to different experimental conditions. The characteristic feature of class comparison is that the classes to be compared are defined independently of the expression data.

Class prediction is similar to class comparison in that the classes are defined independently of the expression data. The emphasis in class prediction problems, however, is in developing a multi-gene classifier that can be applied to expression profiles of samples whose class is unknown to predict the class of the new samples. For example, a class comparison problem may involve identifying the genes that are differentially expressed between patients who respond to a specified treatment and those who don't respond. Developing a classification function that can be used to predict whether a new patient will respond to that therapy based on the gene expression profile of his or her tumor, is class prediction. Class prediction is particularly useful in medical problems of therapy selection or diagnostic classification or prognostic prediction.

Class discovery is quite different from class comparison or class prediction. In class discovery there is no classification defined independently of the expression profiles. The objective is to discover subsets (clusters) of the cases revealed by gene expression profiles and to identify the genes that distinguish the clusters. For example, Bittner et al. [4] examined expression profiles of patients with advanced malignant melanoma. The focus of the study was on attempting to identify a new taxonomy of advanced melanoma based on gene expression. No useful clinical classification existed. Class discovery also includes studies whose objective is to discover classes of co-regulated genes.

Although class comparison and class discovery problems are not unique to the genomics setting, most methods used for such problems were not developed for the context where the number of candidate features (p) is orders of magnitude greater than the number of cases (n). This is the case for genomic and proteomic studies. The number of features can number in the tens of thousands but the number of cases rarely exceeds a few hundred and often is less than one hundred. In this article I shall review some analytic methods that have been found useful for p>>n class prediction problems, and will touch on some problems that frequently occur when the p>>n issues are not adequately addressed.

Four main components to developing a class predictor are: (i) Feature selection; (ii) Selecting a prediction model; (iii) Fitting the prediction model to training data; and (iv) Estimating the prediction error that can be expected in future use of the model with independent data.

2. Feature selection
Feature selection is often key to developing an accurate class predictor. Often in microarray studies, as long as feature selection

is performed reasonably, accurate predition is achieved with even the simplest of predictive models. It is well known from the theory of linear regression that including too many "noise variables" in the predictor reduces the accuracy of prediction. A noise variable is a variable that is not related to the thing being predicted. Feature selection is particularly important in microarray studies because the number of noise variables may be orders of magnitude greater than the number of relevant variables. The influence of the genes that actually distinguish the classes may be lost among the noise of the more numerous noise genes unless we select the informative genes to be utilized by the class predictor.

The most commonly used approach to feature selection is to identify the genes that are differentially expressed among the classes when considered individually. For example, if there are two classes, one can compute a t-test or a Mann-Whitney test for each gene. The log-ratios or log-signals are generally used as the basis of the statistical significance tests. The genes that are differentially expressed at a specified significance level are selected for inclusion in the class predictor. The stringency of the significance level controls the number of genes that are included in the model. If one wants a class predictor based on a small number of genes, the threshold significance level is made very small.

Several authors have developed methods to identify optimal sets of genes which together provide good discrimination of the classes [5], [13], [6], [12]. These algorithms are generally very computationally intensive. Unfortunately, it is not clear whether the increased computational effort of these methods is warranted. In some cases, the claims made do not appear to be based on properly cross-validated calculations; all of the data being used to select the genes and cross-validation used only for fitting the parameters of the model. Thorough studies comparing the performance of such methods to the simpler univariate methods are needed.

Some investigators have used linear combinations of gene expression values as predictors [24] [11]. *Principal components* are the orthogonal linear combinations of the genes showing the greatest variability among the cases. The principal components are sometimes referred to as singular values [1]. Using principal components as predictive features provides a vast reduction in the dimension of the expression data, but has two serious limitations. One is that the principal components are not necessarily good predictors. The second problem is that measuring the principal components requires measuring expression of all the genes. The method of gene shaving attempts to provide linear combinations with properties similar to the principal components that does not require measuring all of the genes [10].
.

3. Prediction Model

Many algorithms have been used effectively with DNA microarray data for class prediction. Dudoit et al. [7] compared several algorithms using publicly available data sets. The algorithms compared included nearest neighbor classification and several variants of linear discriminant analysis and classification trees. A linear discriminant is a function

$$l(\underline{x}) = \sum_{i \in F} w_i x_i \qquad (1)$$

where x_i denotes the log-ratio or log-signal for the i'th gene, w_i is the weight given to that gene, and the summation is over the set F of features (genes) selected for inclusion in the class predictor. For a two-class problem, there is a threshold value d, and a sample with expression profile defined by a vector $\underline{x}$ of values is predicted to be in class 1 or class 2 depending on whether $l(\underline{x})$ as computed from equation (1) is less than or greater than d respectively.

Several kinds of class predictors used in the literature have the form shown in (2). They differ with regard to how the weights are determined. The oldest form of linear discriminant is Fisher's linear discriminant [16]. The weights are selected so that the mean value of $l(\underline{x})$ in class 1 is maximally different from the mean value of $l(\underline{x})$ in class 2. The squared difference in means divided by the pooled estimate of the within-class variance of $l(\underline{x})$ was the specific measure used by Fisher. To compute these weights, one must estimate the correlation between all pairs of genes that were selected in the feature selection step. The study by Dudoit et al. indicated that Fisher linear discriminant analysis did not perform well unless the number of selected genes was small relative to the number of samples; otherwise there are too many correlations to estimate and the method tends to be un-stable and over-fit the data.

Diagonal linear discriminant analysis is a special case of Fisher linear discriminant analysis in which the correlation among genes is ignored [7]. By ignoring such correlations, one avoids having to estimate many parameters, and obtains a method which performs better when the number of samples is small. Golub's weighted voting method [9] and the compound covariate predictor of Radmacher et al.[14] are similar to diagonal linear discriminant analysis and tend to perform very well when the number of samples is small. They compute the weights based on the univariate prediction strength of individual genes and ignore correlations among the genes.

Support vector machines are popular in the machine learning literature. Linear kernel support vector machines use a predictor of the form of equation (2). The weights are determined by optimizing an error rate criterion, however, instead of a least-squares criterion as in linear discriminant analysis [15]. Although there are more complex forms of support vector machines, they appear to be inferior to linear kernel SVM's for class prediction with large numbers of genes [3].

Khan et al.[11] reported accurate class prediction among small, round blue cell tumors of childhood using an artificial neural network. The inputs to the ANN were the first ten principal components of the genes; that is, the l0 orthogonal linear combinations of the genes that accounted for most of the variability in gene expression among samples. Their neural network used a linear transfer function with no hidden layer and hence it was a linear *perceptron* classifier of the form of equation (2). Most true artificial neural networks have a hidden layer of nodes, use a non-linear transfer functions and individual features

as inputs. Such a "real" neural network may not perform as well as the principal component perceptron model of Khan et al. because of the number of parameters to be estimated would be too large for the available number of samples.

In the study of Dudoit et al. [7], the simplest methods, diagonal linear discriminant analysis and nearest neighbor classification, performed as well or better than the more complex methods. Nearest neighbor classification is based on a feature set F of genes selected to be informative for discriminating the classes and a distance function $d(\underline{x},\underline{y})$ which measures the distance between the expression profiles $\underline{x}$ and $\underline{y}$ of two samples. The distance function utilizes only the genes in the selected set of features F. To classify a sample with expression profile $\underline{y}$, compute $d(\underline{x},\underline{y})$ for each sample $\underline{x}$ in the training set. The predicted class of $\underline{y}$ is the class of the sample in the training set which is closest to $\underline{y}$ with regard to the distance function. A variant of nearest neighbor classification is k-nearest neighbor classification. For example with 3-nearest neighbor classification, you find the three samples in the training set which are closest to the sample $\underline{y}$. The class which is most represented among these three samples is the predicted class for $\underline{y}$.

Dudoit et al. also studied some more complex methods such as classification trees and aggregated classification trees. These methods did not appear to perform better than diagonal linear discriminant analysis or nearest neighbor classification. Ben-Dor et al. [3] also compared several methods on several public datasets and found that nearest neighbor classification generally performed as well or better than more complex methods.

4. Fitting Predictive Model to Training Data

Most kinds of predictors have parameters that must be assigned values before the predictor is fully specified. These parameters are in many ways equivalent to the regression coefficients of linear and non-linear regression models.

After selecting the kind of class predictor to be used, the predictor is fitted to a set of data. The number of parameters that must be specified is often proportional to the number of genes selected for inclusion in the model. For some kinds of predictors there is a cut-point that must be specified for translating a quantitative predictive index into a predicted class label (eg 0 or 1) for binary class prediction problems. Completely specifying the predictor means specifying all of these aspects of the predictor, the type of predictor, the genes included and the values of all parameters.

5. Estimating Prediction Accuracy

It is important to estimate the accuracy of class prediction for future samples for which the class is unknown? Knowing that there are highly statistically significant genes that are differentially expressed between the classes is not enough. We want to know how accurately we can predict which class a new sample is in. For a future sample, we will apply a fully specified predictor developed using the data available today. If we are to emulate the future predictive setting in developing our estimate of predictive accuracy, we must set aside some of our samples and make them completely inaccessible until we have a fully specified predictor that has been developed from scratch without utilizing those set aside samples.

To properly estimate the accuracy of a predictor for future samples, the current set of samples must be partitioned into a training set and a separate test set. The test set emulates the set of future samples for which class labels are to be predicted. Consequently the test samples cannot be used in any way for the development of the prediction model. This means that the test samples cannot be used for estimating the parameters of the model and they cannot be used for selecting the genes to be used in the model. This later point is often overlooked.

The most straightforward method of estimating the accuracy of future prediction is the *split-sample* method of partitioning the set of samples into a training set and a test set as described in the previous paragraph. Rosenwald et al. [17] used this approach successfully in their international study of prognostic prediction for large cell lymphoma. They used two thirds of their samples as a training set. Multiple kinds of predictors were studied on the training set. When the collaborators of that study agreed on a *single fully specified predictive model*, they accessed the test set for the first time. On the test set there was no adjustment of the model or fitting of parameters. They merely used the samples in the test set to evaluate the predictions of the model that was completely specified using only the training data.

Cross-validation is an alternative to the split sample method of estimating prediction accuracy. There are several forms of cross-validation. One commonly used variant, *leave-one-out cross-validation (LOOCV)*, starts like split-sample cross validation in forming a training set of samples and a test set. With LOOCV, however, the test set consists of only a single sample; the rest of the samples are placed in the training set. The sample in the test set is placed aside and not utilized at all in the development of the class prediction model. Using only the training set, the informative genes are selected and the parameters of the model are fit to the data. Let us call M_1 the model developed with sample 1 in the test set. When this model is fully developed, it is used to predict the class of sample 1. This prediction is made using the expression profile of sample 1, but obviously without using knowledge of the true class of sample 1. Symbolically, if $\underline{x}_1$ denotes the complete expression profile of sample 1, then we apply model M_1 to $\underline{x}_1$ to obtain a predicted class $\hat{c}_1$. This predicted class is compared to the true class label c_1 of sample 1. If they disagree, then the prediction is in error. Then a new training set – test set partition is created. This time sample 2 is placed in the test set and all of the other samples, including sample 1, are placed in the training set. A new model is constructed from scratch using the samples in the new training set. Call this model M_2. Model M_2 will generally not contain the same genes as model M_1. Although the same algorithm for gene selection and parameter estimation is used, since model M_2 is constructed from scratch on the new training set, it will in general not contain exactly the same gene set as M_1. After creating M_2, it is applied to the expression profile $\underline{x}_2$ of the sample in the new test set to obtain a predicted class $\hat{c}_2$. If this predicted class does not agree with the true class label c_2 of the second sample, then the prediction is in error.

The process described in the previous paragraph is repeated n times where n is the number of biologically independent samples. Each time it is applied, a different sample is used to form the single-sample test set. During the n steps, n different models are created and each one is used to predict the class of the omitted sample. The prediction errors are totaled and that is the leave-one-out cross-validated estimate of the prediction error. With two classes, one can use a similar approach to obtain cross-validated estimates of the sensitivity, specificity, and ROC curve [21].

If we use all of the data to select genes and construct a model, there is no independent data left to validly estimate prediction error. A commonly used invalid estimate is called the *re-substitution* estimate [19]. You use all the samples to develop a model M. Then you predict the class of each sample i using it's expression profile $\underline{x}_i$; $\hat{c}_i = M(\underline{x}_i)$. The predicted class labels are compared to the true class labels and the errors are totaled.

Simon et al. [19] performed a simulation to examine the bias in estimated error rates for class prediction. In a simulated data set, twenty expression profiles of 6000 genes were randomly generated from the same distribution. Ten profiles were arbitrarily assigned to "Class 1" and the other ten to "Class 2", creating an artificial separation of the profiles into two classes. Since no true underlying difference exists between the two classes class prediction will perform no better than a random guess for future biologically independent samples. Hence, the estimated error rates for simulated data sets should be centered around 0.5 (i.e, ten misclassifications out of twenty).

Figure 1 shows the observed number of misclassifications resulting from each level of cross-validation for 2000 simulated data sets. It is well-known that the re-substitution estimate of error is biased for small data sets and the simulation confirms this, with 98.2 % of the simulated data sets resulting in zero misclassifications even though no true underlying difference exists between the two groups. Moreover, the maximum number of misclassified profiles using the re-substitution method was only one.

Two types of leave-one-out cross-validation were studied. In one approach the features to be used in the class predictor were selected using all of the data before starting the cross-validation process. This is partial cross-validation. With proper cross-validation, the gene selection is re-done for each leave-one-out training set.

Figure 1 shows that partial cross-validation is about as bad as no cross-validation. Cross-validating the prediction rule after selection of differentially expressed genes from the full data set does little to correct the bias of the re-substitution estimator: 90.2% of simulated data sets still result in zero misclassifications. It is not until gene selection is also subjected to cross-validation that we observe results in line with our expectation: the median number of misclassified profiles jumps to eleven, although the range is large (0 to 20).

The simulation results underscore the importance of cross-validating all steps of predictor construction in estimating the error rate. A study of breast cancer also illustrates the point: van 't Veer *et al.* [23] predicted clinical outcome of patients with axillary node-negative breast cancer (metastatic disease within 5 years versus disease-free at 5 years) from gene expression profiles, first using the re-substitution method and then using a fully cross-validated approach. The investigators controlled the number of misclassified recurrent cases (i.e., the sensitivity of the test) in both situations, so here we focus attention on the difference in estimated error rates for the disease-free cases. The improperly cross-validated method and the properly cross-validation result in estimated error rates of 27% (12 out of 44) and 41% (18 out of 44), respectively. The improperly cross-validated method results in a seriously biased under-estimate of the error rate. While van 't Veer *et al.* report both estimates of the error rate, the properly cross-validated estimate was reported only in the supplemental results section on the website and the invalid estimate received more attention. Another example of this occurred in a study where classification trees were built from gene expression data to classify specimens as normal colon or colon cancer [25]. The authors used a procedure that only cross-validated steps that occurred after selection of genes for inclusion in the predictor from the full data set. As our simulation shows, not subjecting gene selection to cross-validation can result in a large bias. Other examples are described by Ambroise and McLachlan [2].

Another common bias in reported cross-validated error rates arises from the consideration of multiple prediction models. Often, the predictive algorithm has one or more *tuning* parameters associated with it. For example, the PAM algorithm has a parameter that controls the degree of shrinkage of the class specific centroids [22]. Suppose one computes a proper cross-validated estimate of the error rate $\hat{e}(\lambda)$ for a range of values of the tuning parameter λ. Although $\hat{e}(\lambda)$ may be an unbiased estimator of the true value $e(\lambda)$, $\min\{\hat{e}(\lambda)\}$ (minimization with regard to λ) is not an unbiased estimator of $\min\{e(\lambda)\}$. Consequently, optimization of tuning parameters should be performed within each step of the cross-validation.

Radmacher et al. [14] discuss a paradigm for proper cross-validation of class predictors. They propose that the statistical significance of the cross-validated error rate be reported. The usual methods for determining statistical significance of an error rate are not valid with cross-validated error estimates. Because the leave-one-out training sets are not independent (they are almost completely overlapping), the number of cross-validated errors does not have a binomial distribution. Radmacher et al. [14] provide an algorithm for estimating the statistical significance of the cross-validated error estimate. They consider all possible (or a large number of random) permutations of the class labels that preserve the numbers of samples in each class. For each permutation of the class labels, the entire cross-validation procedure is repeated. They thus generate the permutation distribution of the cross-validated prediction error. The proportion of the permutations that give as small a cross-validated error rate as that obtained for the true data is taken as the statistical significance level for the cross-validated error rate.

There is considerable confusion about the proper use of cross-validation. You cannot cross-validate a model. The model to be used in the future will generally be the one fit to the entire set of data. The cross validation procedure does not utilize that model. It utilizes the model building algorithm, including the feature-

selection algorithm, to attempt to provide an unbiased estimate of the error rate of the complete data model. This point seems to be sometimes misunderstood by computer scientists and statisticians.

Cross-validation is a limited form of validation. Leave-one-out cross validation is known to provide an estimate of the error rate having large variance and other forms of cross-validation and bootstrap re-sampling provide smaller variance estimates [8]. Cross-validation does not provide as stringent a test of a model as would testing a model on a truly independent set of data from a different institution.

Some criticisms of cross-validation, however, seem invalid. Cross validation does provide an essentially unbiased estimate of the error rate of classification that would be obtained for samples from the same distribution as in the training set, in spite of some assertions to the contrary [20]. Some individuals point out that with tens of thousands of features, there will almost surely be a feature set that perfectly predicts both the training set and every split sample test set. Although this is true, it is not a valid criticism of cross-validation. A proper cross validation must apply the feature selection and model building algorithm to each leave one out training set and to select a single model for classifying the left out sample. The fact that there is a model that predicts perfectly, does not imply that some well defined algorithm will find it for each leave one out training set. Algorithms that overfit the data will generally not have a low cross-validated error estimate if the cross-validation is performed properly.

Class prediction problems in p>>n settings are increasingly important in medical applications of genomics, but provide serious challenges to the statistical and computational scientist. Conventional wisdom and routine practices for p<n prediction problems tend to give poor results in the p>>n setting. Guidelines commonly used to guide modeling, such as use of VP dimension are not useful for p>>n problems because they are measures of the complexity of the selected model, not of the space from which that model was selected.

Complex algorithms often perform more poorly than simpler algorithms for p>>n problems. Although the simpler models may not be sufficiently rich to describe the true relationship between predictors and outcome, there is not sufficient data to fit models that are more flexible. With orders of magnitude more candidate features than cases, huge training sets are needed to effectively utilize complex models.

Sample re-use methods such as cross-validation and bootstrap are frequently used improperly for p>>n problems, not only by experimental scientists, but also by statistical and computational scientists. Such methods, when used properly, are very valuable in p>>n problems.

6. Summary

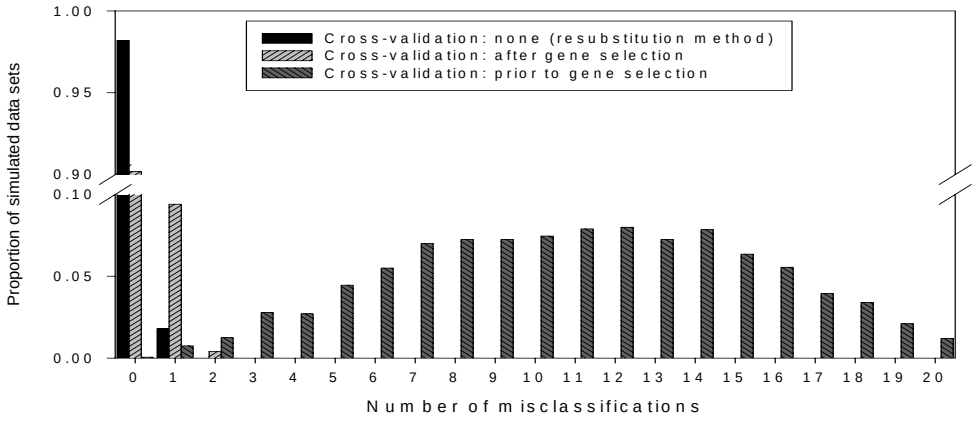

Figure 1. The effect of various levels of cross-validation on the estimated error rate for predicting with random data [19]

REFERENCES

[1] Alter, O., Brown, P. O. and Botstein, D. Singular value decomposition for genome-wide expression data processing and modeling. Proceedings of the National Academy of Science 97 (2000), 10101-6.

[2] Ambroise, C. and McLachlan, G. J. Selection bias in gene extraction on the basis of microarray gene-expression data. Proceedings of the National Academy of Science 99 (2002), 6562-66.

[3] Ben-Dor, A., Bruhn, L., Friedman, N., et al. Tissue classification with gene expression profiles. Journal of Computational Biololgy 7 (2000), 559-584.

[4] Bittner, M., Meltzer, P., Chen, Y., et al. Molecular classification of cutaneous malignant melanoma by gene expression profiling. Nature 406 (2000), 536-540.

[5] Bo, T. H. and Jonassen, I. New feature subset selection procedures for classification of expression profiles. Genome Biology 3 (2002), 0017.1-0017.11.

[6] Deutsch, J. M. Evolutionary algorithms for finding optimal gene sets in microarray prediction. Bioinformatics 19 (2003), 45-54.

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

[8] Efron, B. and Tibshirani, R. Improvements on cross-validation: The .632+ bootstrap method. Journal of the American Statistical Association 92 (1997), 548-560.

[9] Golub, T. R., Slonim, D. K., Tamayo, P., et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science 286 (1999), 531-537.

[10] Hastie, R., Tibshirani, R., Eisen, M., et al. Gene shaving as a method for identifying distinct sets of genes with similar expression patterns. Genome Biology 1 (2000), 0003.1-0003.21.

[11] Khan, J., Wei, J. S., Ringner, M., et al. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. Nature Medicine 7 (2001), 673-9.

[12] Kim, S., Dougherty, E. R., Barrera, J., et al. Strong feature sets from small samples. Journal of Computational Biology 9 (2002), 127-146.

[13] Ooi, C. H. and Tan, P. Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. Bioinformatics 19 (2003), 37-44.

[14] Radmacher, M. D., McShane, L. M. and Simon, R. A paradigm for class prediction using gene expression profiles. Journal of Computational Biololgy 9 (2002), 505-12.

[15] Ramaswamy, S., Tamayo, P., Rifkin, R., et al. Multiclass cancer diagnosis using tumor gene expression signatures. Proceedings of the National Academy of Science 98 (2001), 15149-15154.

[16] Ripley, B. D., Pattern recognition and neural networks, Cambridge University Press, Cambridge U.K., 1996.

[17] Rosenwald, A., Wright, G., Chan, W. C., et al. The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. New England Journal of Medicine 346 (2002), 1937-47.

[18] Simon, R., Korn, E. L., McShane, L. M., et al., Design and Analysis of DNA Microarray Investigations, Springer Verlag, New York, 2003.

[19] Simon, R., Radmacher, M. D., Dobbin, K., et al. Pitfalls in the use of DNA microarray data for diagnostic and prognostic classification. Journal of the National Cancer Institute 95 (2003), 14-18.

[20] Somorjai, R. L., Dolenko, B. and Baumgartner, R. Class prediction and discovery using gene microarray and proteomics mass spectroscopy data: curses, caveats, cautions. Bioinformatics 19 (2003), 1484-1491.

[21] Swets, J. Measuring the accuracy of diagnostic systems. Science 240 (1988), 1285-93.

[22] Tibshirani, R., Hastie, T. and al., e. Diagnosis of multiple cancer types by shrunken centroids of gene expression. Proceedings of the National Academy of Science 99 (2002), 6567-72.

[23] van't-Veer, L. J., Dai, H., Vijver, M. J. v. d., et al. Gene expression profiling predicts clinical outcome of breast cancer. Nature 415 (2002), 530-6.

[24] West, M., Blanchette, C. and Dressman, H. Predicting the clinical status of human breast cancer by using gene expression profiles. Proceedings of the National Academy of Science 98 (2001), 11462-67.

[25] Zhang, H., Yu, C. Y., Singer, B., et al. Recursive partitioning for tumor classification with gene expression microarray data. Proceedings of the National Academy of Science 98 (2001), 6730-35.