Made available by Hasselt University Library in https://documentserver.uhasselt.be

Performance of gene selection and classification methods in a microarray setting: A simulation study

Peer-reviewed author version

VAN SANDEN, Suzy; LIN, Dan & BURZYKOWSKI, Tomasz (2008) Performance of gene selection and classification methods in a microarray setting: A simulation study. In: COMMUNICATIONS IN STATISTICS-SIMULATION AND COMPUTATION, 37(2). p. 409-424.

Handle: http://hdl.handle.net/1942/8363

PERFORMANCE OF GENE SELECTION AND CLASSIFICATION METHODS IN A MICROARRAY SETTING: A SIMULATION STUDY

Suzy Van Sanden, Dan Lin, and Tomasz Burzykowski
Hasselt University,
Center for Statistics,
Agoralaan, gebouw D,
B-3590 Diepenbeek, Belgium

Corresponding author:
Suzy Van Sanden
Hasselt University,
Center for Statistics,
Agoralaan, gebouw D,
B-3590 Diepenbeek, Belgium
suzy.vansanden@uhasselt.be

Original data of receipt: 21/12/2006

Key Words: cDNA microarray; class prediction; gene selection; simulation study.

Running Head: Performance of Classification Methods

Primary AMS Mathematics Classification number: 62H30

Secondary AMS Mathematics Classification number: 68U20

ABSTRACT

In a previous paper we investigated the performance of several classification methods for cDNA-microarrays. Via simulations various experimental settings could be explored without having to conduct expensive microarray studies. For the selection of genes on which classification was based, one particular method was applied. Gene selection is however a very important aspect of classification. We would like to extend the previous study by considering several gene selection methods. Furthermore, the stability of the methods with respect to distributional assumptions is examined by also considering data simulated from a symmetric and asymmetric Laplace distribution in addition to normally distributed microarray data.

1. INTRODUCTION

Microarrays allow the monitoring of expression levels of thousands to tens of thousands of genes simultaneously in a given cell type. One of the uses of microarray data is to generate gene expression profiles which can discriminate between different known cell types or conditions. An important question is what method is most suitable for this purpose?

Van Sanden *et al.* (2007) compared the performance of several classification methods using simulated microarray data. The study supplements prior investigations that were based on the use of real-life data (Dudoit *et al.* 2002, Lee *et al.* 2005, and Statnikov *et al.* 2005). The latter capture the complexity of microarray measurements more adequately but simulated data are not limited to settings for which data are available. Furthermore, in simulations the true classification, as well as the set of truly differentially expressed genes, is known. Thus the misclassification rate can be accurately determined.

In this paper we extend the Van Sanden *et al.* (2007) study in two directions. First of all, the study was limited to one gene selection method prior to performing classification. Lee *et al.* (2005) found that the choice of the gene selection method has much effect on the performance of the classification procedures. They considered four gene selection techniques combined with a number of classification methods, and applied them to seven real-life datasets. In the present paper we incorporated 14 well-known selection methods into the simulation study to investigate their influence on 11 classification procedures under several

controlled scenarios. There are still other gene selection (linear models (Smyth, 2004), the TnoM score (Ben-Dor *et al.*, 2000), random forest based approach (e.g. Diaz-Uriarte and Alvarez de Andres, 2006)) and classification methods (LogitBoost (Dettling and Bhlmann, 2003), BagBoosting (Dettling, 2004), Regular Discriminant Analysis (e.g. Guo *et al.*, 2007)) that have been recently proposed. However, to keep the study feasible, we decided not include them in the simulations. We also did not consider dimension reduction methods (like MAVE (Antoniadis *et al.*, 2003) or PLS (Nguyen and Rocke, 2002; Boulesteix and Strimmer, 2007)). Secondly, in the Van Sanden *et al.* (2007) study the normal distribution has been used as the underlying distribution in the simulation procedure. It has been shown (Purdom and Holmes, 2005), however, that microarrays do not always follow the normal distribution. Thus, we considered the symmetric Laplace distribution as a long-tailed alternative. Moreover, we also used the asymmetric Laplace distribution.

To simulate the data a linear mixed effects simulation model that mimics a microarray context was used. The parameters of the model were chosen based on an analysis of a real-life experiment. For the analysis of this experiment, as well as for the simulation of the data, SAS 9.1 was used. The simulated data were analysed using R.

The paper is organized as follows. In Section 2 we shortly describe the investigated gene selection and class prediction methods. Section 3 contains details of our simulation study. Results are described in Section 4. A short discussion and conclusions are presented in Sections 5 and 6, respectively.

2. METHODS

In this section we give a brief description the gene selection and classification methods considered in the study. In most cases existing R functions were used for the implementation of the different methods. In what follows the necessary packages, as well as the parameter settings of the particular functions, are indicated.

2.1 GENE SELECTION

A typical microarray experiment involves many genes of which usually only a few are

differentially expressed. Therefore, one may want to select and base the classification on a subset of, say p, genes, majority of which would hopefully differentiate between the compared classes of samples. An important issue is how to select the genes? In the simulation study we assessed the use of several methods for this purpose.

Classical test statistics

We considered a basic parametric test statistic, the t-test (Ttest) and a basic non-parametric test statistic, the Wilcoxon Rank Sum test (Wilc). R-functions *stat.t2* from the `sma` package and *wilcox.test* from `stats` are used for this purpose.

Significance analysis of microarrays (SAM)

SAM is a method for analysing microarray experiments and detecting significant genes. It was proposed by Tusher *et al.* (2001). A score (modified t-statistic) is assigned to each gene based on change in gene expression relative to the standard deviation augmented by a small positive constant. This constant ensures that the variance of the score is independent of gene expression. Its value is chosen to minimize the coefficient of variation of the test statistic. The modified t-statistic for the case of two unpaired classes was calculated by the *samr* function from the `samr` package.

Prediction analysis for microarrays (PAM)

PAM fits a nearest shrunken centroid classifier to microarray data. The method, also referred to as soft-thresholding, was introduced by Tibshirani *et al.* (2002). It provides a list of significant genes whose expression best characterizes each class. The functions *pamr.train* and *pamr.listgenes* from the `pamr` package contain the implementation of the method.

Extreme-value distribution based gene selection (Extval)

Li *et al.* (2004) introduced gene selection based on comparison of the maximum likelihood of a logistic regression model applied to the original data and permutation datasets. To avoid using computational intensive procedures they propose to take advantage of the extreme-value distribution for the log likelihood ratios. From there follows a ranking of the genes which can be used to select a predefined number of genes. Li *et al.* (2004) also sug-

gest two criteria to determine the number of genes to be selected from the list, one based on the expected values (E-criterion) and one based on p-values (P-criterion). They were both included in our simulation study. The gene selection method and both criteria were implemented in R.

Other test statistics

There are a number of other statistics used for ranking and selection of genes. We considered a couple of well-known methods. Each time the p genes with the largest statistic were selected and used for classification. The BW ratio is the ratio of the between-treatment sum of squares and the within-treatment sum of squares (Dudoit *et al.* 2002). In a two group setting the BW-ratio reduces to the same statistic as the t-test. The prediction strength (PS) (Xiong *et al.* 2001) of a certain gene is defined as the ratio of the difference in mean log expression level between the two groups and the sum of the variances of the two classes. The between-class scatter score (BC-score) belongs to the class of correlation scores (Chai and Domenicioni 2004). Most of these scores are designed to handle multi class problems and reduce to one of the above statistics when applied to a two class problem. The BC-score is obtained by dividing the weighted squared difference of the class mean from the overall mean by the sum of the variances of the two classes. All the above mentioned test statistics were implemented in R.

Statistical impurity measures

In contrast to determining a test statistic, we can attempt to find a gene-specific threshold in the expression range. If a measured value for a particular gene is larger (resp. smaller) than this threshold, the sample is assigned to for instance class one (resp. two). Statistical impurity measures quantify the effectiveness of this method. There are several ways this can be done, leading to multiple impurity measures: twoing rule (Twoingr), information gain (Infgain), Gini index (Gini), max minority (Maxmin), sum minority (Summin), and sum of variances (Sumvar). A full description of them can be found in Murthy *et al.* (1994) and Su *et al.* (2003). The methods were implemented in R.

Empirical cumulative distribution function (ECDF)

The following method was introduced as a pre-screening technique by Boulesteix *et al.* (2003). Genes are selected based on how well the Empirical Cumulative Distribution Function (ECDF) of the two classes are separated. One chooses values α and β and retains genes for which there exist a point where the ECDF is less than α for one class and more than β for the other, or more than $1 - \alpha$ for one class and less than $1 - \beta$ for the other. Initially, α is set at 0.1 and β at 0.5. The value for α (resp. β) is increased (resp. decreased) to be able to select the desired number of genes. After the proper values of alpha and beta are found and the genes are selected, we can determine for each of those genes the interval of maximum width satisfying the above mentioned conditions. The width of the intervals can then be used to order the genes.

2.2 CLASS PREDICTION

We focused on the simplest case of discrimination between two different groups of samples. The class prediction procedures investigated in our study included tree methods, classical discrimination analysis techniques, and machine learning methods. For a more detailed description of the methods we refer to Van Sanden *et al.* (2007).

Classification trees

A classification tree is a binary recursive partitioning method developed by Breiman *et al.* (1984). It is implemented as the function *dorpart*, which is part of the `IBCLab4` package available at: http://bioinf.wehi.edu.au/marray/ibc2004/Rpackages/IBCLab4.html.

Aggregated classifiers combine tree classifiers to improve the accuracy of the class prediction. One such method is called *bagging* (Breiman 1996). It is based on taking bootstrap replicates (in our case 100) from the training dataset, constructing a tree for each of them and determining classification by majority vote. The method is implemented by the function *ipredbagg.factor* from the `ipred` package.

Boosting, proposed by Schapire and Freund (1999) is another form of aggregating classifiers. A series of classification trees is produced for the training dataset, each time with

different weights assigned to the samples. The idea is to give samples misclassified in the previous step more weight in the current one. The final outcome is a weighted majority vote of all created trees. The *gbm* function from the `gbm` package was first applied to the data in order to create 100 trees in the manner described above. When applying the function, Bernoulli distribution, a shrinkage parameter of 0.001 and the fraction of randomly selected observations for building a tree of 0.5 were used. The *gbm.more* function was used to create 1000 additional trees.

Random forests (Breiman, 2001) are constructed by independently drawing subsets of samples (with replacement) and genes (without replacement) from the training dataset. Classification is determined by majority vote. The method is implemented as the function *randomForest* from the `randomForest` package. The size of the samples that were drawn was set equal to the total number of samples. This is the default value in R. For the genes it is determined by a function specified in the help file of *randomForest*. The method was applied with the number of trees equal to 500 and 1000.

k nearest neighbors (kNN)

kNN (Ripley, 1996) is an intuitive method that classifies unlabeled samples to most frequent class label among the k closest samples (using the Euclidean distance measure) from the training set. The method is implemented as the function *knn* in the package `class`. *knn.cv* was used to determine the value of k.

Discriminant analysis

Linear discriminant analysis (LDA), a classical discriminant method, estimates linear discriminant functions for decision boundaries based on assumptions of Gaussian distribution and equal covariance matrices for the grouped data. Variants of the method assume equal (diagonal linear discriminant analysis, DLDA) or unequal (diagonal quadratic discriminant analysis, DQDA) diagonal covariance matrices for the considered classes. LDA is implemented as the function *lda* in the `MASS` package, while *stat.diag.da* from the `sma` package was used for DLDA and DQDA.

Support vector machines (SVM)

Support vector machines, first introduced by Cortes and Vapnik (1995) in the machine learning theory are used to solve two-group classification problems. The method is implemented as the *SVM* function in the package `e1071`. In the study we included the linear, polynomial and radial kernel. The regularization parameter was set equal to one. The other parameters were set at the default value of the R-function.

3. SIMULATION STUDY

We assumed the setting of an experiment using two-channel cDNA microarrays in a common reference design. In the experiment, two classes of samples (a treatment and a control group, say) were compared to the same reference group. No difference between expression levels for the control and reference groups was assumed. On the other hand, a subset of genes was assumed to be differentially expressed in the treatment group as compared to the reference group. Treatment and control samples were labeled with the same dye, while the reference samples were labeled with the other dye. The log ratios of the two channels (Control/Reference and Treatment/Reference) were used for classification purposes.

For every setting of interest 100 simulation datasets were created, each containing 160 arrays: 60 were used as a training dataset and the other 100 formed a test dataset. One half of the arrays in each set always contained the treatment sample, while the other half was for the control group. On each array we simulated 2000 genes, what corresponds to the number of genes on arrays used in the real-life experiment we were using as a basis for the simulation study (Van Breda*et al.* 2005). A linear mixed effects model was utilized to simulate observations subject to various systematic and random effects usually present in real microarray experiments (Kerr, Martine and Churchill 2000), (Wolfinger emphet al. 2001). An observation Y_{ijg}, assumed to be the mean signal intensity for array i ($i = 1, \ldots, 100$), dye j ($j = 1, 2$) and gene g ($g = 1, \ldots, 2000$), was generated by the following model:

$$log_2(Y_{ijg}) = \mu + A_i + G_g + AG_{ig} + DG_{jg} + TG_g + \varepsilon_{ijg}, \tag{1}$$

where μ is the overall mean, A_i stands for the overall array effect, G_g for the gene effect

and AG_{ig} for their interaction. DG_{jg} and TG_g represent, respectively, the gene specific dye and treatment effects, while ε_{ijg} is a random error. Some of these effects are fixed, while others are random and were drawn from a normal distribution. The details are displayed in Table 1. The chosen values were based on the estimated parameters of a ANOVA-model fitted to a real-life cDNA microarray dataset (Van Breda *et al.* 2005). The gene effects (G_g, DG_{jg}, TG_g and σ_g^2) for a specific gene were kept constant across datasets. We were thereby simulating the repetition of an experiment each time involving the same set of genes.

TABLE 1

FIGURE 1

Figure 1 shows histograms of the log intensity values for two arrays chosen from the real-life dataset. As can be seen from these graphs, for one of them the normal distribution may be a reasonable approximation. For the other a slightly asymmetric distribution with a long tail on the right side might be more appropriate.

Figure 1 also presents a normal QQ-plot of the log transformed gene-specific variances of the error terms ε_{ijg} obtained from the ANOVA-model fitted to the real-life dataset. In this particular case a log normal distribution seems to be a very good approximation for the distribution of the gene specific variance of ε_{ijg}. Consequently, it has been used in the simulation model.

In model (1) the gene specific dye effect is represented by the difference in the means of the normal distributions for DG_{jg}. Simulated arrays with even numbers contain control and reference group samples. Arrays with odd numbers contain treatment and reference samples. The latter arrays contain 20 genes that are differentially expressed. We considered two scenarios with respect to the treatment effect. In the first one, for all 20 genes a constant treatment effect of 0.5 on the log scale was assumed. In the second one, we assumed 4 groups of 5 genes with treatment effects equal to, respectively, 0.125, 0.250, 0.375 and 0.5 on the log scale.

Heterogeneity was incorporated into model (1) by allowing the random error term ε_{ijg} to have a gene-specific variance, selected randomly from a log-normal distribution (see Table 1), as observed in the real life-dataset (see Figure 1).

Laplace distribution

For the real-life dataset, which is the basis of the simulation model, the normality assumption for the error terms ε_{ijg} holds reasonably well. Often the distribution of microarray data has longer tails than the normal or is somewhat asymmetric. One of the examples in Figure 1 demonstrates these features to some extent. We wish to examine the effect of such a deviation on the performance of the selection and classification methods. Purdom and Holmes (2005) propose the use of the asymmetric Laplace distribution for microarray data. It is more peaked compared to the normal distribution. We included in the study data simulated from the symmetric and asymmetric Laplace distribution. This was achieved by replacing the normal distribution, as described in Table 1, by $L(0, \sigma_g)$ for the symmetric or $AL(0, \kappa, \sigma_g)$ for the asymmetric Laplace distribution. The density function of $AL(\theta, \kappa, \sigma)$ is given by

$$f(x) = \frac{\sqrt{2}}{\sigma} \frac{\kappa}{1+\kappa^2} \begin{cases} exp(\frac{-\sqrt{2}\kappa}{\sigma} |x-\theta|) & x \geq \theta \\ exp(\frac{-\sqrt{2}}{\sigma\kappa} |x-\theta|) & x < \theta \end{cases} \tag{2}$$

while that of the symmetric Laplace distribution is obtained when κ is put equal to 1.

The gene specific scale parameter (σ_g) is the same as for the case with the normal distribution (see Table 1). For the skewness parameter κ we chose values of 0.5 (skewed to the left) and 1.2 (skewed to the right). They are similar or a bit more extreme than estimates found by Purdom and Holmes (2005) for several microarrays from published microarray experiments. Their estimates range from 0.792 to 1.174. To make sure we can study the effect of skewness on the selection and classification methods we made the asymmetry to the left a little bit more severe.

4. RESULTS

In this section we present the results of the various selection and classification methods in different settings, starting with data simulated from the normal distribution and with a

constant treatment effect. Further on, we investigate changes to the results when considering other distributions or varying treatment effects.

4.1 NORMALLY DISTRIBUTED DATA WITH A CONSTANT TREATMENT EFFECT

First, a comparison of the performance of the gene selection methods is made. It can be evaluated in two ways: by the number of genes selected that are actually differentially expressed or by the misclassification rate when combining the selection methods with different classification procedures. This rate is calculated based on the number of misclassified arrays divided by total number of arrays (100) in the test dataset.

The gene selection methods were applied to the simulated training datasets to obtain subsets of genes of varying sizes (p=2, 5, 10, 20, 40, 200). Table 2 shows the median numbers of the truly differentially expressed genes among the p=10 selected genes. Though many methods result in similar numbers, some patterns can be observed. Certain impurity measures, especially Maxmin, do not perform as well as some of the other methods. ECDF seems by far the least desirable method. Furthermore, there is no obvious winner, though SAM appears to perform slightly better than all other methods.

TABLE 2

In a next step we evaluate the gene selection methods when combined with different classification procedures. For demonstration purposes the results for p=10 and 200 (Figure 2) were chosen. The figure contains box plots of the misclassification rates, computed over the 100 simulated datasets. First of all a comparison between the results for p=10 and 200 reveals the importance of proper gene selection. When a lot of noise (genes that are not differentially expressed) is present, the performance of most classification methods (except for tree methods) weakens. This conclusion is supported by Van Sanden *et al.* (2007). Furthermore, no gene selection method is clearly outperforming all the others.

FIGURE 2

For the classification trees all gene selection methods are performing similarly, except for SAM, BC-score and ECDF. ECDF leads in general to poor results. The other two fail

mainly when a relatively low number of genes is selected. The choice of the best gene selection method seems to depend on the classification method used and the number of selected genes. When the value of p becomes larger, the difference between the methods fades.

The results for kNN are similar. SAM, BC-score and ECDF are not recommendable. There is however more variation between the other methods, also for large values of p. The best methods appear to be Wilc, PS, PAM, Ttest (=BW) and Extval.

For the different forms of discriminant analysis and SVM the best methods are also Wilc, PS, PAM, Ttest (=BW) and Extval, while Maxmin, SAM, BC-score and ECDF perform rather poorly. However, in contrary to the tree methods and kNN, ECDF improves on several methods when p is quite large ($p \geq 200$), while the performance of Extval weakens.

Next to gene selection, we can also re-examine the performance of the classification methods, particularly for the optimal selection procedures (Figures 3). In the previous paper (Van Sanden *et al.*, 2007) we only considered one SVM method based on a linear kernel. In the current study two other kernels are considered, the polynomial and radial. It appears that polynomial SVM gives the poorest results of the three. Radial SVM mostly outperforms linear SVM, except when a large number of genes is used for classification. The difference between radial and linear SVM fades or even reverts when p increases to 2000 (data not shown). The bad performance of polynomial SVM becomes then even more clear.

FIGURE 3

In general radial SVM does not outperform some of the other classification methods. Considering all gene selection procedures, DLDA is one of the best methods. Random Forrest is not doing so bad either, especially when the number of genes is large ($p \geq 200$, data not shown). In that particular setting it is outperforming all other methods. Furthermore LDA, DQDA and radial SVM work reasonably well.

4.2 OTHER SETTINGS

The performance of the selection and classification methods was also investigated for data with a proportionally increased treatment effect, and with other distributional assumptions

for the error term ε_{ijg}. Table 2 displays the mean percentage of truly differentially expressed genes for all settings. The comparison between different selection methods appears quite similar under the different assumptions. There is however some variability over the settings with different distributional assumptions.

The size of the log ratios of the two channels depends, besides on the dye and on the treatment effect, on the difference between the error terms of the model that generates the ratios (equation 1). When this difference has the sign opposite of the treatment effect, it will make it more difficult or even impossible to pick up the treatment effect. Table 3 contains some basic properties (variance and kurtosis) of $f_{X-Y}(z)$. When X, Y $\sim L(0,\ \sigma)$ or $AL(0,\ 1.2,\ \sigma)$, $f_{X-Y}(z)$ has a similar variance but a higher kurtosis than when X, Y $\sim N(0,\ \sigma^2)$. Therefore the distribution in question is more peaked with the same spread. It has more weight near zero and in the tails. However, the probability of getting a value in the tails is quite small, and values close to zero have less chance of diminishing the treatment effect to the point where it is not detected anymore. This explains why error terms with a symmetric or right skewed asymmetric Laplace distribution allow for a better detection of the significant genes than when they follow the normal distribution. The opposite is true when X, Y $\sim AL(0,\ 0.5,\ \sigma)$: $f_{X-Y}(z)$ is more peaked than when X, Y $\sim N(0,\ \sigma^2)$, but also has a lot more variability. The chance of getting values in the tails of this distribution is not so small in this case. It thus explains why fewer of the significant genes were picked up.

TABLE 3

Figure 4 displays median misclassification rates for certain combinations of gene selection and classification methods under different settings. The shift between the curves representing a certain setting is directly related to the variability over the different settings seen in Table 2 and discussed above. Only small differences are noticeable when comparing profiles of different settings. However, in general the results are not substantially or consistently affected by proportionally increasing the treatment effect, or by simulating data from the symmetric or asymmetric Laplace distribution. Similar results were found for the other combinations of gene selection en classification methods (data not shown).

FIGURE 4

4.3 E- AND P-CRITERIA

Figure 5 displays the misclassification rates for various classification methods combined with all considered gene selection methods, including the E- and P-criteria. For the methods that do not determine the optimal number of genes (p) to select, several values of p were considered. The E-criterion generally outperforms the P-criterion, except for some classification methods where both give almost the same result. The E-criterion also works quite well compared to other selection methods. It leads to the lowest median misclassification rate, except when combined with a tree method. In that case its performance is still very close to that of the best choice of gene selection method and value of p.

FIGURE 5

5. DISCUSSION

Given the complexity of microarray data, one probably should not expect that a single gene selection or classification method will always outperform all the others. It is therefore paramount to investigate relative merits of different procedures to see in which settings which methods might be expected to work reasonably well.

Lai *et al.* (2005) compared a number of univariate and multivariate gene selection algorithms across several cancer diagnostic problems. They did not detect any significant improvement when employing multivariate gene selection techniques. They argue that this finding could be due to very limited sample size. However, in our previous study (Van Sanden *et al.*, 2007) we simulated correlation between certain genes. We also did not find this to have an effect on the performance of the classification methods. With these findings in mind and to keep the size of the study manageable, we decided not to include multivariate methods at this time. It is a topic for further research.

Simulations allow to overcome limitations related to the use of real-life data. For instance, we were able to investigate the performance of the gene selection and classification

methods in many controlled settings. Also, using real-life data we can only obtain a point estimator for the misclassification rate of a certain classification method. To get an estimate of the variance, re-sampling techniques would be necessary. With simulated data we directly obtain the distribution of the misclassification rate, with the precision depending on the number of simulated datasets. Additionally, the true classification, as well as the set of truly differentially expressed genes, are known. Hence we can evaluate the performance of the gene selection methods directly and study the link between the performance of the classification method and the number of genes used for it that are truly differentially expressed. On the other hand, an important issue is whether the simulated data adequately capture the complexity of microarray measurements. The real-life data used as a basis for our simulations could be approximated by data simulated from a linear mixed model. In general, this may not be the case. To deal with this issue, we considered the use of non-normal distributions, the symmetric and asymmetric Laplace distribution.

Another issue is related to the computational complexity of such a study. Simulating the data and performing all the classification procedures on them is time consuming. Extending the simulations by including more methods or simulating data for more genes, more datasets or more settings is therefore not trivial. For this reason, in our study we chose for a limited, yet relevant from a practical point of view (as documented by Van Breda (2005)), setting. In particular, the study was designed for cDNA microarrays. Although it is difficult to predict to what extent the conclusions can be generalized to oligonucleotide arrays, both platforms have common features that we expect to influence the gene selection and classification methods in a similar way. For instance, in both cases a large number of genes is available for building a classifier and a lot of them are uninformative. It is therefore expected that the conclusions regarding the influence of the number of chosen genes might apply to oligonucleotide arrays as well.

6. CONCLUSIONS

Though there are differences between the gene selection methods, there is not a single one which surpasses all others. Wilc, PS, PAM, Ttest (=BW) and Extval lead to quite good

results. The E-criterion is also worth noting. It works well most of the time and does not require predefining the number of genes to select. On the other hand Maxmin, SAM and BC-score are not performing as well as some of the other methods. ECDF is found to be the least interesting. The ECDF method was proposed to look for genes that might jointly discriminate between classes. For this reason, it allows for a larger overlap between the classes, than other gene selection procedures. Its poorer performance seen in our simulations is therefore expected and understandable.

When comparing the effect of different gene selection methods, it is not just a matter looking at how many truly differentially expressed genes a certain method is able of detecting. The method that leads to the largest percentage of truly differentially expressed genes (SAM) is not the method leading to the lowest misclassification rate. It seems important which particular genes (differentially expressed or noise) are included in the subset on which classification is based.

The best choice of the classification method does not appear to depend strongly on the gene selection procedure. In general DLDA and Random Forest give the lowest misclassification rates. DLDA is performing slightly better than Random Forest, except when a large number of genes is used. Radial SVM is clearly the best machine learning method. It is however in most cases not good enough to compete with DLDA.

In our simulations introduction of longer tailed or asymmetric distributions for the microarray data or specification of a proportionally increased, instead of a constant, treatment effect did not substantially affect the comparison between the different gene selection and classification methods.

ACKNOWLEDGMENT

We gratefully acknowledge support of the IAP research network P5/24 of the Belgian State (Belgian Science Policy).

REFERENCES

Antoniadis, A., Lambert-Lacroix, S., and Leblanc, F. (2003). Effective dimension reduction

methods for tumor classification using gene expression data. *Bioinformatics*, **19**, 563–570.

Ben-Dor, A., Friedman, N., and Yakhini, Z. (2000). Scoring genes for relevance. Technical Report 2000-38, School of Computer Science and Engineering, Hebrew University, Jerusalem.

Boulesteix, A.L., Tutz, G., and Strimmer, K. (2003). A CART-based approach to discover emerging patterns in microarray data. *Bioinformatics*, **19**, 2465–2472.

Boulesteix, A.L., and Strimmer, K. (2007). Partial least squares: a versatile tool for the analysis of high-dimensional genomic data. *Briefings in Bioinformatics*, **8**(1), 32–44.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984). *Classification and regression trees*. New York: Chapman & Hall.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, **24**, 123–140.

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

Chai, H., and Domeniconi, C. (2004). An evaluation of gene selection methods for multi-class microarray data classification. In: Proc. 2nd European workshop on data mining and text mining in bioinformatics, 3–10.

Cortes, C., and Vapnik, V. (1995). Support-Vector Networks. *Machine Learning*, **20**, 273–297.

Dettling, M., and Bühlmann, P.B. (2003). Boosting for tumor classification with gene expression data. *Bioinformatics*, **19**(9), 1061–1069.

Dettling, M. (2004). BagBoosting for tumor classification with gene expression data. *Bioinformatics*, **20**(18), 3583–3593.

Diaz-Uriarte, R., and Alvarez de Andres, S. (2006). Gene selection and classification of microarray data using random forest. *BMC Bioinformatics*, **7**:3.

Dudoit, S., Fridlyand, J., and Speed, T.P. (2002). Comparison of discrimination methods for

the classification of tumors using gene expression data. *Journal of the American Statistical Association,* **98**, 77–87.

Freund, Y., and Schapire, R.E. (1999). A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, **14**, 771–780.

Guo, Y., Hastie, T., and Tibshirani, R. (2007). Regularized linear discriminant analysis and its application in microarrays. *Biostatistics*, **8**(1), 86–100.

Kerr, K.M., Martin, M., and Churchill, G.A. (2000). Analysis of variance for gene expression microarray data. *Journal of Computational Biology*, **7**, 819–838.

Lai, C., Reinders, M.J.T., and Wessels, L.F.A. (2005). Multivariate gene selection: does it help? IEEE Computational Systems Biology Conference, Stanford, California, USA, August 8-12 2005.

Lee, J.W., Lee, J.B., Park, M., and Song, S.H. (2005). Extensive comparison of recent classification tools applied to microarray data. *Computational Statistics and Data Analysis*, **48**, 869–885.

Li, W., Sun, F., and Grosse, I. (2004). Extreme value distribution based gene selection criteria for discriminant microarray data analysis using logistic regression. *Journal of Computational Biology*, **11(2/3)**, 215–226.

Murthy, S.K., Kasif, S., and Salzberg, S. (1994). A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, **2**, 1–33.

Nguyen, D., and Rocke, D. (2002). Tumor classification by partial least squares using microarray gene expression data. *Bioinformatics*, **18**, 39-50.

Purdom, E., and Holmes, S.P. (2005). Error distribution for gene expression data. *Statistical Applications in Genetics and Molecular Biology*, **4**, 1, article 16.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge .

Smyth, G. K. (2004). Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical Applications in Genetics and Molecular Biology*, **3**, 1, Article 3.

Statnikov, A., Aliferis, C.F., Tsamardinos, I., Hardin, D., and Levy, S. (2005). A comprehensive evaluation of multicateogry classification methods for microarray gene expression cancer diagnosis. *Bioinformatics*, **21**, 631–643.

Su, Y., Murali, T.M., Pavlovic, V., Schaffer, M., and Kasif, S. (2003). Rankgene: a program to rank genes from expression data. *Bioinformatics*, **19**, 1578–1579.

Tibshirani, R., Hastie, T., Narasimhan, B., and Chu, G. (2002). Diagnosis of multiple cancer types by shrunken centroids of gene expression.*Proceedings of the National Academy of Sciences of the United States of America*, **99**, 6567–6572.

Tusher, V., Tibshirani, R., and Chu, G. (2001). Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, **98**, 5116–5121.

Van Breda, S., van Agen, E., van Sanden, S., Burzykowski, T., Kleinjans, J., and van Delft, J. (2005). Vegetables affect the expression of genes involved in anticarcinogenic processes in the colonic mucosa of C57BL/6 female mice. *Journal of Nutrition*, **135(8)**, 1879–1888.

Van Sanden, S., Lin, D., and Burzykowski, T. (2007). Performance of classification methods in a microarray setting: a simulation study. *Biocybernetics and Biomedical Engineering* (accepted for publication).

Wolfinger, R.D., Gibson, G., Wolfinger, E.D., Bennett, L., Hamadeh, H., Bushel, P., Afshari, C., and Paules, R.S. (2001). Assessing gene significance from cDNA microarray expression data via mixed models. *Journal of Computational Biology*, **8**, 625–637.

Xiong, M., Li, W., Zhao, J., Jin, L., and Boerwinkle, E. (2001). Feature (gene) selection in gene expression-based tumor classification. *Molecular Genetics and Metabolism*, **73(3)**,

239–47.

Table 1: Simulation model parameters

Parameter	Value/Distribution	Parameter	Value/Distribution
μ	$= 9$	TG_g	$= 0.5 \times I(g \leq 20, i = \text{odd}, j = 1)$
A_i	$\sim N(0.5,\ 0.1)$	DG_{jg}	$\sim N\{1 \times I(j = 1),\ 0.2\}$
G_g	$\sim N(0,\ 5)$	ε_{ijg}	$\sim N(0; \sigma_g^2)$
AG_{ig}	$\sim N(0,\ 0.5)$	$log(\sigma_g^2)$	$\sim N(-2,\ 0.5)$

Table 2: Median number of truly differentially expressed genes among a subset of size p=10 selected by different methods. (C=constant and P=prop. increased treatment effect)

$\epsilon_{ijg} \sim$		Ttest	Wilc	SAM	PAM	PS	BC-score	ECDF	Infgain	Twoingr	Summin	Maxmin	Gini	Sumvar	Extval
$N(0,\ \sigma^2)$	C	9	9	10	9	9	9	4	9	9	8	7	9	9	9
$N(0,\ \sigma^2)$	P	7	7	7	6	7	6	2	6	6	5	4	6	6	7
$L(0,\ \sigma)$	C	10	10	10	10	10	10	7	10	10	10	10	10	10	10
$AL(0,\ 0.5,\ \sigma)$	C	9	9	9	9	9	9	3	8	8	8	7	8	8	9
$AL(0,\ 1.2,\ \sigma)$	C	10	10	10	10	10	10	5.5	10	10	10	10	10	10	10

Table 3: Variance and kurtosis of $f_{X-Y}(z)$.

X, Y $\sim$	Variance	Kurtosis
$N(0,\ \sigma^2)$	$2\sigma^2$	0
$L(0,\ \sigma)$	$2\sigma^2$	1.5
$AL(0,\ 0.5,\ \sigma)$	$4.26\sigma^2$	2.67
$AL(0,\ 1.2,\ \sigma)$	$2.14\sigma^2$	1.68

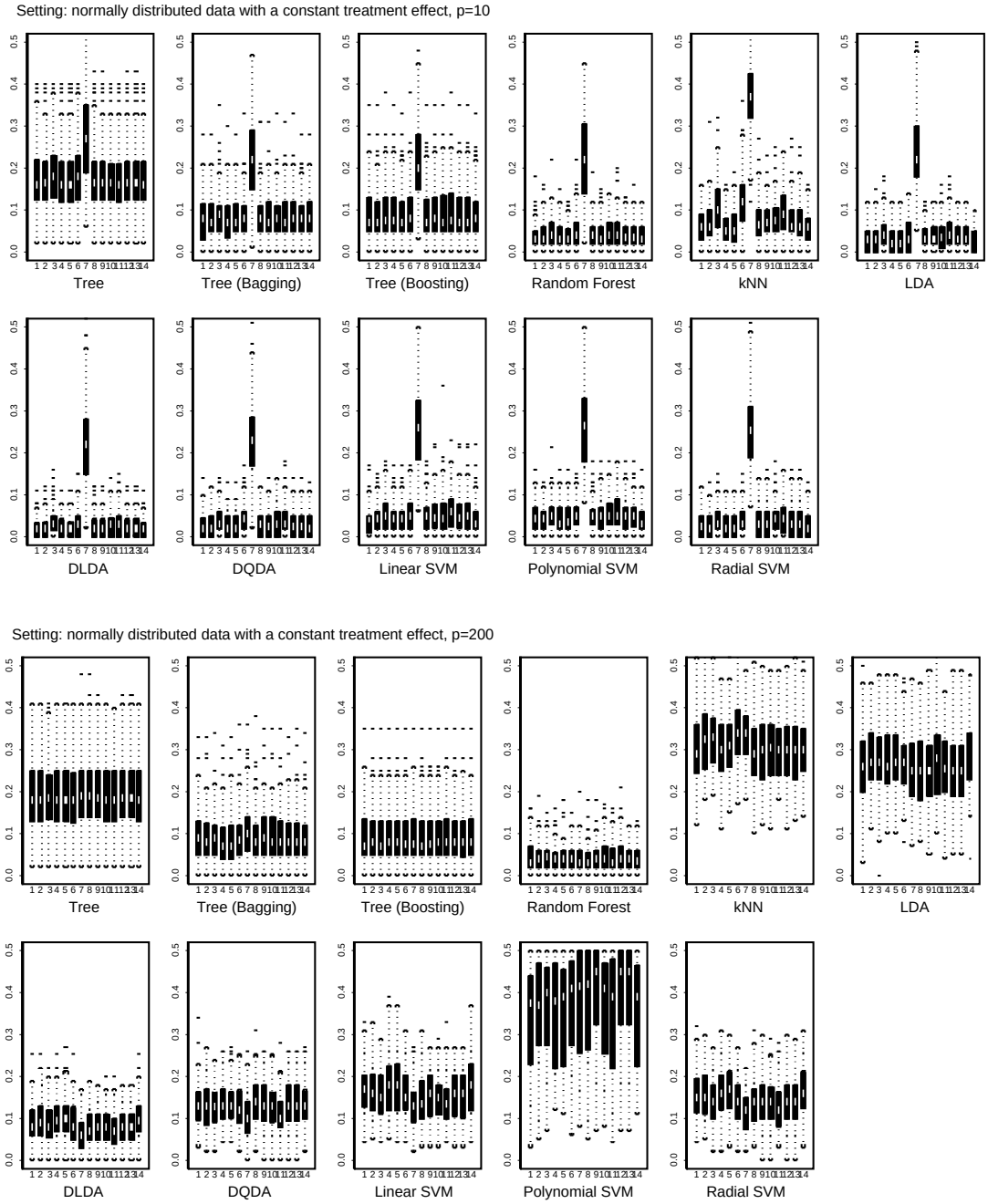

Figure 2: Box plots for the misclassification rates (vertical axis) of various classification methods using p=10 (resp. p=200) genes selected by different selection methods (horizontal axis). [1: Ttest, 2: Wilc, 3: SAM, 4: PAM, 5: PS, 6: BC-score, 7: ECDF, 8: Infgain, 9: Twoingr, 10: Summin, 11: Maxmin, 12: Gini, 13: Sumvar, 14: Extval]

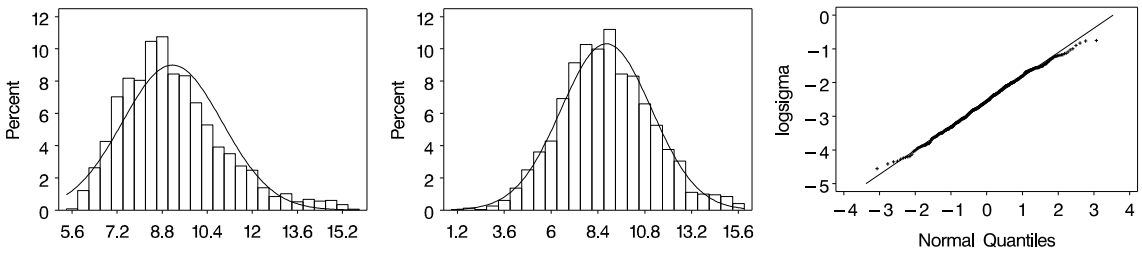

Figure 1: Histograms with normal density curve of the log transformed mean signal measurements for two arrays of the real-life cDNA microarray dataset and normal QQ-plot of the log transformed gene-specific variances of the error terms obtained from the ANOVA-model fitted to the real-life cDNA microarray dataset.

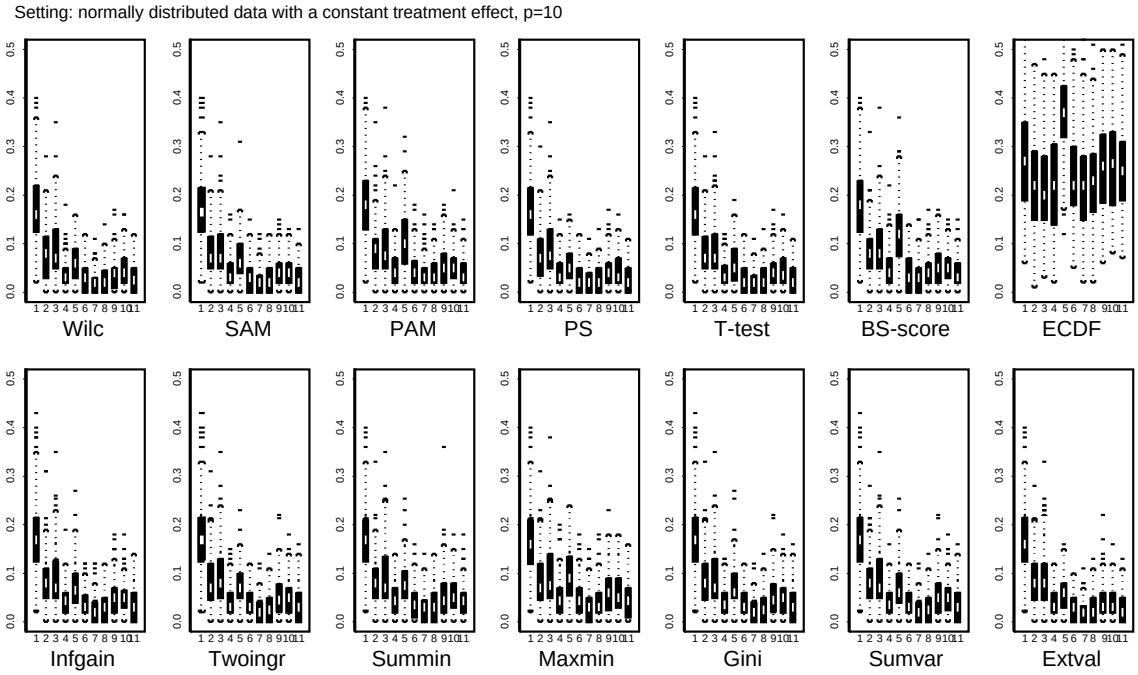

Figure 3: Box plots for the misclassification rates (vertical axis) of various classification methods (horizontal axis) using p=10 genes selected by different selection methods. [1: Tree, 2: Tree (Bagging), 3: Tree(Boosting), 4: Random Forest, 5: kNN, 6: LDA, 7: DLDA, 8: DQDA:, 9: Linear SVM, 10: Polynomial SVM, 11: Radial SVM]

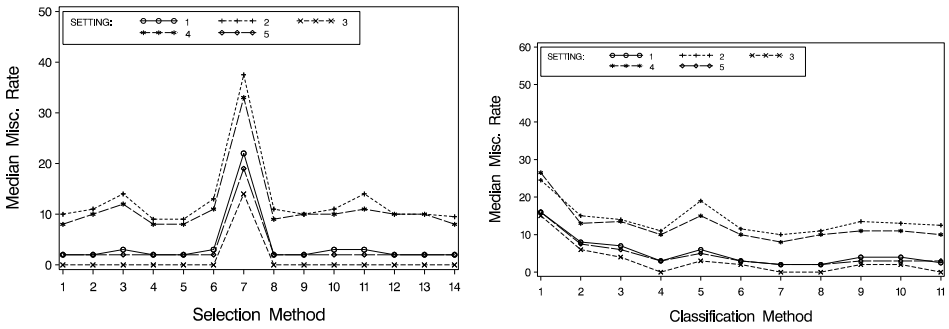

Figure 4: Misclassification rates for certain combinations of gene selection and classification methods. Setting 1 and 2: normally distributed data with constant (resp. proportionally increased) treatment effect; setting 3, 4 and 5: symmetric (resp. asymmetric with skewness parameter κ=0.5 and κ=1.2) Laplace distributed data. Left: DLDA [1: Ttest, 2: Wilc, 3: SAM, 4: PAM, 5: PS, 6: BC-score, 7: ECDF, 8: Infgain, 9: Twoingr, 10: Summin, 11: Maxmin, 12: Gini, 13: Sumvar, 14: Extval]; Right: Ttest [1: Tree, 2: Tree (Bagging), 3: Tree(Boosting), 4: Random Forest, 5: kNN, 6: LDA, 7: DLDA, 8: DQDA:, 9: Linear SVM, 10: Polynomial SVM, 11: Radial SVM];

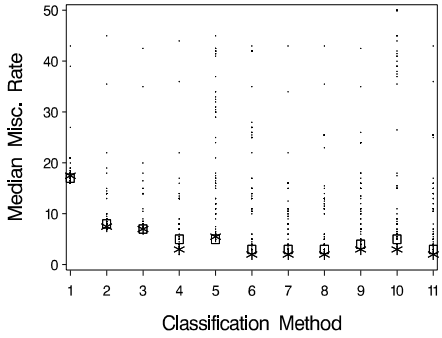

Figure 5: Misclassification rates of various classification methods (horizontal axis). For every classification method,he dots represent Ttest, Wilc, SAM, PAM, PS, BC-score, ECDF, Infgain, Twoingr, Summin, Maxmin, Gini, Sumvar and Extval and for values of p=2, 5, 10, 20, 40, 200 and 2000. Stars correspond to the E-criteria and squares to the P-criteria. [1: Tree, 2: Tree (Bagging), 3: Tree(Boosting), 4: Random Forest, 5: kNN, 6: LDA, 7: DLDA, 8: DQDA:, 9: Linear SVM, 10: Polynomial SVM, 11: Radial SVM]