STUDIA INFORMATICA 2006
Volume 27 Number 4 (69)

Katarzyna STĄPOR
Politechnika Śląska, Instytut Informatyki

Adrian BRÜCKNER, Paweł BŁASZCZYK
Uniwersytet Śląski, Instytut Matematyki

A COMPARATIVE REVIEW OF THE SELECTION METHODS FOR DISCOVERING DIFFERENTIALLY EXPRESSED GENES IN MICROARRAY EXPERIMENTS FOR CLASSIFICATION

Summary. In this paper the feature selection methods applied to discovering differentially expressed genes in microarray experiments are compared. This compareson includes both filter and optimal subset selection methods. The simulated and biological datasets are used as the microarray gene expression data, and the ability of selected genes for classification is also considered.

Keywords: feature selection, multiple hypothesis testing, microarray experiment, supervised learning

PRZEGLĄD PORÓWNAWCZY METOD SELEKCJI GENÓW RÓŻNICUJĄCYCH W EKSPERYMENTACH MIKROMACIERZOWYCH DLA KLASYFIKACJI

Streszczenie. W artykule porównano metody selekcji cech zastosowane do wykrywania genów różnicujących w eksperymentach mikromacierzowych. Porównanie zawiera zarówno metody statystyczne, jak i metody poszukiwania optymalnego podzbioru cech. Jako dane mikromacierzowe wykorzystano symulowane zbiory danych oraz dane biologiczne. Przedstawiono ponadto przydatność wyselekcjonowanych genów do klasyfikacji.

Słowa kluczowe: selekcja cech, wielokrotne testowanie hipotez, eksperyment mikromacierzowy, uczenie nadzorowane

1. Introduction

A common problem in genetics was how to discover genes which were responsible for some diseases. Nowadays microarrays technology was used to monitor expression levels of genes. The expression levels in samples were compared in order to discover the genes which separate the classes properly. There were three kinds of treatment to solve this problem: filter, wrapper and embedded methods. The filter method used only ordered individual feature filter to selected features. In the wrapper method features selection was connected with the predictor. The idea of the wrapper approach was simple. The induction algorithm was run on the dataset, usually partitioned into internal training and holdout sets with different sets of features removed from the data. The feature subset with the highest evaluation was chosen as the final set on which to run the induction algorithm. The resulting classifier was then evaluated on an independent test set that was not used during the search. In the embedded method, however, the features selection was precisely connected with the predictor.

In this article we concentrated on the filter and the wrapper methods. We compared 2 filter methods and 1 wrapper method. The filter methods were statistical methods while the wrapper method was the optimal subset selection method. We compared all of the methods on simulated and biological data. In statistical methods we used traditional two-sample t-test and raw p-value algorithm to estimated p-values. The p-values were corrected with Benjamini & Hochberg (BH) method and with Bonferroni (B) method et al. [6]. We applied FDR method of Dudoit et al [2]. These methods were compared with an optimal subset selection RFE method. We also discussed the results of classification on the selected genes.

None of these methods was restricted to any specific microarray technology. In the real gene data applications the gene expression level may had been suitably preprocessed. In this article we used both preprocessed simulated data and biological data.

2. Methods

Let $X = \left[x_{ij} : i = 1,...,M, j = 1,...,N \right]$ denote the gene expression levels matrix with the rows corresponding to genes and columns to individual microarray experiments (arrays), and let $y_1,...,y_N$, $y_i \in \{-1,1\}$ be the array class labels.

2.1. Filter methods

Let X_i denote the random variable corresponding to the expression level for gene i and let Y denote the response of covariate. Let the null hypothesis for that gene i be not differentially expressed (DEG), will be as:

H_i : There is no association between X_i and Y

It is important to choose appropriate test statistics. Because of the fact that we have only two kinds of samples in our dataset, we use the t-test. For gene i we used two sample t-statistic given with equation

$$t_i = \frac{\bar{x}_2(i) - \bar{x}_1(i)}{\sqrt{\frac{S^2{}_1(i)}{n_1} + \frac{S^2{}_2(i)}{n_2}}} \tag{1}$$

where $\bar{x}_j(i)$ is the mean expression of gene i in class j, n_j is the number of samples in class j and S^2_j(i) is the variance of gene i in class $j \in \{-1,1\}$.

The null hypothesis shows that the gene is not differentially expressed and such hypothesis is rejected for large values of $|t|$.

We used permutation resampling method to estimate p values because we did not know the distribution of the expression values in microarray experiments,. To calculate p values we used the raw p-value algorithm [4] which can be summarized in the following way:

For the each permutation b=1,…,B

1. Permute the n columns of the data matrix X
2. Compute the statistic $t_{1,b}$,…., $t_{M,b}$ for each hypothesis

After B permutations, the permutation p-values for hypothesis H_i is:

$$p^*{}_i = \frac{\#\{b : |t_{i,b}| \geq |t_i|\}}{B} \text{ for } i = 1,...,M \tag{2}$$

Assuming the type I error at the level α hypothesis H is rejected when $p \leq \alpha$. But this is a classical situation when only one null hypothesis is tested. A typical microarray experiment measures expression levels of thousands of genes simultaneously so it is a multiple testing problem. In this situation two types of errors can occur: type I error of false positive when a gene is declared to be DEG (differentially expressed genes) when it is not, and type II error or false negative, when the test fails to identify a truly DEG. To control the type I error we cannot use the threshold for p-values (obtained with the raw p-value algorithm) but there are many generalizations of this error to multiple testing situation in the literature. One possibility to define type I error in multiple hypothesis testing problems is the false discovery rate (FDR). FDR is an expected value of the proportion of type I errors among the rejected hypotheses, thus $E(V/R)$ where V is the number of false positives and R is the number of rejected hypotheses. There are many definitions of FDR in the case $R=0$, we choose the definition of Benjamini & Hochberg to put $V/R=0$ when $R=0$. In the consequence we have:

$$FDR = E\left(\frac{V}{R} \mid R>0\right)\Pr(R>0) \tag{3}$$

Multiple testing correction adjusts the individual p-value for each gene to keep the overall error rate below the specified threshold. Adjusted p-value for FDR is defined as:

$$\tilde{p}_i = \inf\{\alpha : H_i \text{ is rejected at FDR} = \alpha\} \tag{4}$$

We used a Benjamini & Hochberg procedure [5, 6] to calculate the adjusted p-values given below:

1. The p-values of each gene are ranked from the smallest to the largest
2. The largest p-value remains
3. The second largest p-value is multiplied by the total number of genes divided by its rank. If it is less than α, it is significant
4. Every next gene is multiplied as in previous step

Another type to define type I error in multiple hypothesis testing problems is the family-wise error rate (FWER). FWER is defined as the probability of at least type I error.

$$\text{FWER} = \Pr(\text{V} > 0) \tag{5}$$

where V is the number of false positives. Method which we used to calculate adjusted p-values for FWER was Bonferroni procedure [5, 6]. The procedure was described below:

1. The p-values of each gene is multiplied by the number of genes in the gene list.
2. $Corrected\_pvalue = pvalue \cdot M$
3. If the corrected p-value is still below the error rate, the gene will be significant

The probabilities and expectation given above are conditional on the true hypothesis

$$\text{H}_{\text{M}_0} = \bigcap_{\text{i}\in\text{M}_0}\{\text{H}_\text{i} = 0\}$$

where $\text{M}_0 = \{\text{i} : \text{H}_\text{i} = 0\}$.

2.2. Optimal subset selection

The recursive feature elimination (RFE) [7] is a backward feature elimination procedure, which iteratively removes non-discriminative features in the binary classification problem. The algorithm can be summarized as follows:

Let $X = \left[x_{ij} : i = 1,...,N, j = 1,...L \right]$ be the training set of size *L*, where x_{ij} is the value of *i*-th feature for a *j*-th sample and $y_1,...,y_L$, $y_i \in \{-1,1\}$ be the class labels. Moreover let $S = \left[1,2,...,N\right]$ denotes surviving feature list and $R = []$ feature ranking list.

while $s \neq []$ do:

1. Train the linear SVM classifier obtaining Lagrange multipliers vector $\alpha = \left[\alpha_1,...,\alpha_L\right]$ and compute the weight vector $w = \sum_j \alpha_j y_j x_j$.
2. Compute the ranking of all the remaining features according to the following criterion $c_i = \left(w_i\right)^2$
3. Eliminate from the list *S* the lowest ranked feature *k*, that is the feature with the lowest value of c_i and fix $R := \left[S(k), R\right]$.

Linear SVM is usually used in gene selection application because the gene expression samples tend to be linearly separable. To make the algorithm faster one can eliminate more than one feature every time.

2.3. Estimating error rates

For each dataset we classified the samples of the test dataset using linear support vector machines classifier with parameter setting the regularization parameter C equals 0,1. To estimate error rates we use the Jackknife resampling which is the special case of the bootstrap procedure et al. [4] and k-fold cross validation procedure et al. [4]. The Jackknife method we used as follows: we selected a single sample of the test dataset, learned the classifier on the remaining samples and classify the chosen sample obtaining the error rate. Error rate was equal to 0 if the sample was classified correctly or to 1 if it was not. This step was repeated for 1000 times. Every time we selected a sample from full set of samples. The bootstrap (jackknife) error rate ER_J was the mean error rate from each step. So the error rate was:

$$ER_J = \frac{1}{n} \cdot \sum_{i=1}^{n} er(i) \tag{6}$$

where *n* was the number of repetitions and $er(i)$ was the error rate in i-th iteration. In this article we used *n* being equal to 1000. The k-fold cross validation method we used as follows: we randomly divided dataset for *k* subsets of equal size. We trained classifier *k* times, each time leaved out one of the subsets from training, but using only the omitted subset to compute error rate. This step was repeated for every *k* subsets. The k-fold cross validation error rate ER_{CV} was the mean error for each *k* subsets. It could be expressed in the following form:

$$ER_{CV} = \frac{1}{k} \cdot \sum_{i=1}^{k} er(i) \tag{7}$$

where k was the number of subsets and $er(i)$ was the error rate in i-th iteration. In this article we used k equals to 5.

We used percentiles of the ER_{CV} and ER_J error distribution to find the end points of the confidence intervals. We used 1000 ER_{CV} and ER_J values estimated in the way given above to estimate the distribution. For the significant level α the confidence interval was bounded at $\alpha / 2$ and $1 - \alpha / 2$ percentiles so the confidence interval was defined by:

$$CI = (p_{\alpha/2}; p_{1-\alpha/2}) \tag{8}$$

where $p_{\alpha/2}$ was $\alpha / 2$ percentile.

3. Experimental results

3.1. Datasets

The methods described in the previous section were compared on several simulated DNA microarray datasets. Firstly, we prepared datasets consisting of two groups of 15 arrays with 2000 genes. Two datasets IS01, IS05 with 1% (20) and 5% (100) differentially expressed genes were generated according to the article by Broberg [1] using normal distributions with parameters given in Table 1. Only the last three rows represented differential expression.

Table 1

Means and standard deviations used in IS01, IS05

Group 1		Group 2	
μ_1	σ_1	μ_2	σ_2
−8	0,2	−8	0,2
−10	0,4	−10	0,4
−12	1	−12	1
−6	0,1	−6,1	0,1
−8	0,1	−8,5	0,2
−10	0,4	−11	0,7

We assumed an equal probability of every model from the first three rows for non-differentially expressed genes and from the second three rows of the table for DEGs. It means that if the gene was simulated as one of the DEGs (one of the 20 genes for IS01 and one of the 100 genes for IS05 dataset) we choosed one of the three last rows from the table with equal probability 1/3 and generate expression values for 15 arrays from group 1 and 15 arrays from group 2 using normal distributions with parameters given in the selected row from the table. For instance, let gene i be one of the DEGs and we choose the last row from the table to generate its expression values. It means that we generated 15 numbers for the first group from the N(−10; 0.4) distribution and 15 numbers for the second group from N(−11; 0,7) distribution. For non DEGs we chosed one of the first three rows from the table with equal probability 1/3 and generated expression values of all 30 arrays from the same normal distribution with parameters given in the selected row of the table.

We also generated another datasets including two groups of arrays with 21 arrays belonging to the first group and 19 arrays to the second one. Each array includes 2000 genes, the proportion of DEGs was set equal to 1% that is we have 20 differentially expressed genes in these datasets. Firstly, we independently generate each entry of the 2000×40 matrix from the standard normal distribution. Secondly, we add a value of 2 to the first 100 genes in the first group to model differentially expressed genes. Thus, first 100 genes in the first group were normally distributed with mean 2 and all the elements of the whole matrix were stochastically independent. Afterwords, we independently generated 40 random numbers $a_1, ..., a_{40}$ from the standard normal distribution. Then, for the fixed correlation value we applied the following transformation for each entry of the generated matrix: , where was the number of gene and was the number of sample, so that for any and j we had . Using the procedure described above, we generated training and test datasets called CS02 and CS06, with chosen correlation strength at the level of 0.2 and 0.6 respectively.

We also compared all of this methods on the leukemia dataset from [6] and available at http://www.broad.mit.edu. The dataset came from a study of gene expression in two types of acute leukemias: acute lybphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Training set contained 38 cases (27 ALL and 11 AML) and test dataset contained 34 cases (20 ALL and 14 AML) both with 7129 genes.

3.2. Results

We applied exactly the same experimental scheme for each simulation dataset. Firstly, we calculated raw p-values from the t-test. We used 10000 permutations to estimate them. Secondly, we controlled FDR at level α equals to 0,01. Next we applied Benjamini & Hochberg correction on this p-values. We also applied Bonferroni correction on estimated p-values and compare the results with results after Benjamini & Hochberg correction. On the basis of the results we built a linear SVM classifier and estimated the error rate with two methods: bootstrap and k-fold cross validation with k equals to 5. We estimated confidence interval at the level 0,95. For optimal subset selection method (RFE) we applied this method on the original dataset and we also built exactly the same classifiers.

For the leukemia dataset the procedure was as in the case of the simulation dataset.

3.2.1. Simulation data

For the IS01 dataset, after raw p-value calculation 34 genes had p-values less than 0,01. In the second dataset IS05, p-value of 176 genes was less than 0,05. These results show that the correction of the p-values was necessary. We applied two different correction methods Benjamini & Hochberg and Bonferroni. After we applied the Benjamini & Hochberg correction in the first dataset 18 genes were discovered as significant genes, assuming that FDR was equal to 0,01. Two of these 18 genes were incorrectly recognized as significant genes. Four genes were not recognized at all. In the second dataset IS05 67 genes were discovered as significant genes after Benjamini & Hochberg correction. Three of these genes were incorrectly recognized as significant genes, 37 genes were not recognized at all. We compared this result with results after Bonferroni correction. On the IS01 dataset Bonferroni method recognize only 11 genes. All of them were recognized properly. Nine genes were not recognized at all. Similar situation was in the case of the second dataset IS05. Fifty four genes were recognized as significant genes. We noted that these two methods have different stringent level. The Bonferroni (B) method was more stringent that Benjamini & Hochberg. We also noticed that in the Bonferroni method less false positives are allowed. Whereas in Benjamini & Hochberg method less false negatives are allowed. The 11 genes recognized by Bonferroni method in the IS01 dataset were also discovered by Benjamini & Hochberg in the same dataset. In IS05 dataset we had exactly the same situation. All genes recognized by B method were also discovered by BH method. It was known that which the results correspond with. These two methods control different types of error. Benjamini & Hochberg control FDR whereas Bonferroni controls FWER. In both cases the control was strong it means that the control is for every possible choice of where .

For the CS02 datasets, after raw p-value calculation 43 genes had p-values less than 0,01. In the second dataset CS06 p-value of 43 genes was less than 0,01. After both Benjamini & Hochberg and Bonferroni correction, 19 of 20 genes were discovered as differentially expressed in the CS02 dataset. Thus we observed one false negative and there were no false positives. When the correlation was stronger (for CS06 dataset) these algorithms (BH, B) gave only 14 of 20 DEGs. There were no false positives as well, but the number of false positives raised to 6. Note that in CS02 dataset, where the results were better than in CS06 dataset, more genes had p-values less than 0,01. It could suggested that the correlation values have an impact on the number of p-values which were under a specified threshold. Also the number of permutation, which were used to calculate raw p-values, could be too small to discover the difference between BH and B methods or some preprocessing should be applied before we started. We want to examined it in the feature work. To built the classifier on these dataset we tuned the regularization parameter C in the validation procedure. In the IS01 dataset regularization parameter C equaled 100 for B method and 10 for BH method. In IS05 dataset for both methods regularization parameter C equaled 1. For the IS01 dataset the results were better than we had expected. For Bonferroni (B) method the best error rates estimated by 5-fold cross validation were for 6÷8 genes. The error rate equaled 0 and the confidential interval was also 0. We noted the mean error rate for two genes and it equaled 0,0303. The worst error rate was for one gene and it equaled 0,233 in confidence interval [0,2; 0,3]. In Jackknife case the results were even better. For 4 genes the error rates equaled 0. We also noted the mean error rate for two genes and it was 0,033, and the worst was for one gene and it equaled 0,266 in the confidence interval equaled [0,238; 0,293]. For Benjamini & Hochberg (BH) method, in both error estimation cases the best error rate was noted for 6 and 8 genes and it equaled 0. The worst error rates, in both cases, were for one gene and it equalled 1 and 0,522 for Jackknife and 5-fold cross validation respectively. It could be explained by the fact that in the results, there could be genes which were not significant genes but had small p-value. For the IS05 dataset the results were similar to IS01 dataset. For Bonferroni (B) method the best error rates estimated by 5-fold cross validation were when the numbers of a gene was bigger than 6 genes. The error rate equaled 0 and the confidential interval was also 0. The mean error rate equaled 0,04. The worst error rate was for one gene and it equaled 0,1 in confidence interval [0,066; 0,166]. In Jackknife case the results were even better. For 6 genes the error rates were equaled 0. We noted the mean error rate for two genes and it was 0,003, and the worst was for one gene and it equaled 0,066 in the confidence interval equaled [0,051; 0,083]. For Benjamini & Hochberg (BH) method in both error estimation cases we also noted the best error rate for 6 and 8 genes and it equaled 0. The worst error rates in both cases were for one gene and it equaled 0,2 and 0,167 for Jackknife and 5-fold cross validation respectively.

For BH and B methods in the CS02 and CS06 dataset parameter C equaled 0,1. The genes selected with BH and B algorithms provide proper classification results for the CS02 dataset. For the CS02 dataset error rates obtained with the Jackknife method for 1 to 20 genes and the obtained confidence intervals were the best for 12 and 13 and equaled 0. Confidence interval is also equaled zero. The mean error rate in CS02 dataset was 0,027 for 15÷17 genes. We noted the worst results for 3 genes. The error rate was 0,18 and the confidence interval was [0,1; 0,25]. Error rates estimated by k-fold cross validation were similar. We also noted the best error rate for 13 genes. The confidence interval equaled [0; 0,05]. We also noted the mean error rate for 15÷17 genes which were bigger and equaled 0,064. The worst results as in the case of Jackknife procedure were for 3 genes and equaled 0,175. Confidence interval equaled [0,125; 0,225]. In the CS06 dataset the results were not as good. In both error estimation method we had very similar results. The best error rate was for 7 and 12 genes and equaled 0,17 for Jackknife and for 5-fold cross validation, in the confidence interval [0,1; 0,25], the worst error rate was for one gene and equaled 0,35 in the confidence interval [0,27; 0,44] for Jackknife and 0,3 in confidence interval [0,2; 0,4] for 5-fold cross validation. Also mean error rate was very similar in both cases and equaled 0,213 and 0,211 for Jackknife and 5-fold cross validation respectively.

The method of the optimal subset selection group was RFE (Recursive Feature Elimination). The result was not satisfactory. On the dataset IS01 in first 20 genes only 10 genes was discovered properly. In the first 1000 genes 19 genes were discovered properly. The last gene was on the 1478th place! There was a similar situation in IS05 dataset. In the first 100 genes only 41 were discovered properly. In the first 200 genes only 60 genes were discovered properly. RFE discovered properly each gene in the first 1760 genes! There was a better situation in CS02 and CS06 datasets. Recursive feature elimination algorithm (RFE) found all DEGs in the first 50 selected genes, but in the first 20 selected genes (notice that in the CS datasets there were only 20 truly DEG) only 14 and 11 genes were selected correctly for the CS02 and CS06 dataset respectively. It means that for CS02 in the first 20 genes there were 6 false positives and for the CS06 there were 9 such genes. Note that in the RFE algorithm the ranking criterion was computed using information about single gene but the top ranked genes were not necessarily the ones that are individually most relevant.

To build the classifier on these dataset we tuned the regularization parameter C in the validation procedure. In the IS01 dataset regularization parameter C equaled 100 for RFE method. In IS05 dataset for both methods the regularization parameter C equaled 1. For IS01 dataset and 5-fold cross validation method we noted the best error rate for 5 and 12 genes and it equaled 0,03 and 0,06 with confidence interval [0; 0,10] and [0; 0,13] respectively. For Jackknife method the error was similar but the confidence interval was smaller. For 5 genes, we noticed error rate equaled 0,067 with confidence interval [0,051; 0,083]. The worst error rate for number of genes was bigger than 15. In both cases the error rate was equaled 0,1 in confidence interval equaled [0,083; 0,19]. The mean error rate was for 6 – 8 genes and it equaled 0,051 in both methods. For IS05 dataset the results were even better. For a number of genes bigger than 39 we noted that error rates equaled 0 in confidence interval also equaled 0. The mean error rate in both cases was 0,015 in confidence interval [0,070; 0,02]. We noticed the worst error rate for one gene. In both cases it equaled 0,233 in confidence interval [0,2; 0,295]. Although RFE did not discover more than 50% genes, the error rate was satisfactory. It can be explained by the fact that RFE select these genes for which predictor performance is the best, not these which are the most statistically significant.

For RFE methods in the CS02 and CS06 dataset parameter C equaled 0,1. The genes selected with RFE algorithm provide proper classification results for the CS02 dataset. For the CS02 dataset we obtained error rates with the Jackknife method for 1 to 20 genes. The obtained error rates were the best for 17 and equaled 0. Confidence interval is also equal to zero. The mean error rate in CS02 dataset was 0,053 for 12 genes. We noted the worst results for one gene. The error rate was 0,15 and the confidence interval was [0,08; 0,22]. The error rates estimated by 5-fold cross validation were similar. We also noted the best error rate for 17 and it equaled 0. The confidence interval equaled [0, 0,025]. We also noted the mean error rate for 7-8, 15 genes which was bigger and equaled 0,059. The worst results, as in the case of in Jackknife procedure, were for one gene and equaled 0,150. Confidence interval equaled [0,100; 0,200]. In the CS06 dataset the results were not as good. We obtained very similar results in both error estimation methods. The best error rate was for 14-15 genes and equaled 0,025 for Jackknife and for 5-fold cross validation, in the confidence interval [0; 0,075], the worst error rate was for one gene and equaled 0,25 for Jackknife and 0,225 for 5-fold cross validation in the confidence interval [0,20; 0,275] for Jackknife and [0,17; 0,34] for 5-fold cross validation. What is more, the mean error rate was very similar in both cases and equaled 0,097 and 0,098 for Jackknife and 5-fold cross validation respectively. The details are shown on the figures below. The solid line indicate error rates estimated by the Jackknife method. The dash line indicate the 5-fold cross validation method.

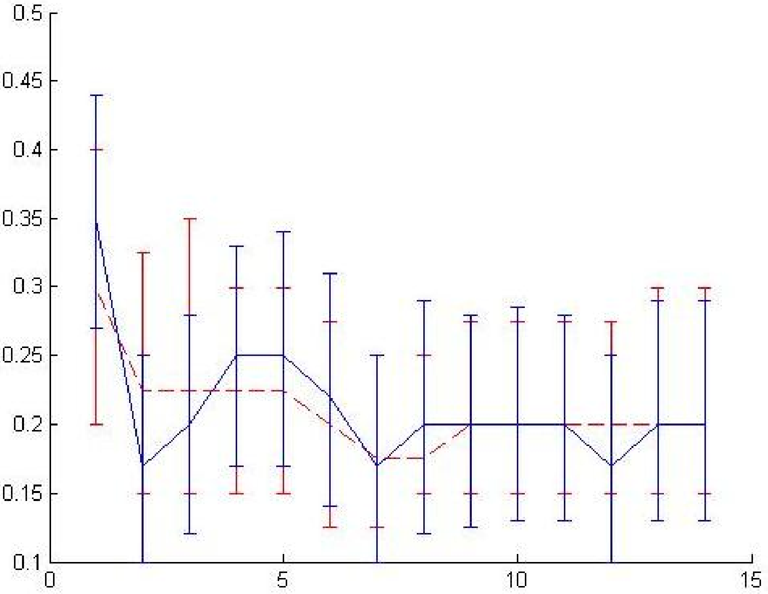

Fig. 1. Error rates for CS06 and BH method
Rys. 1. Błąd klasyfikacji dla zbioru CS06 i metody BH

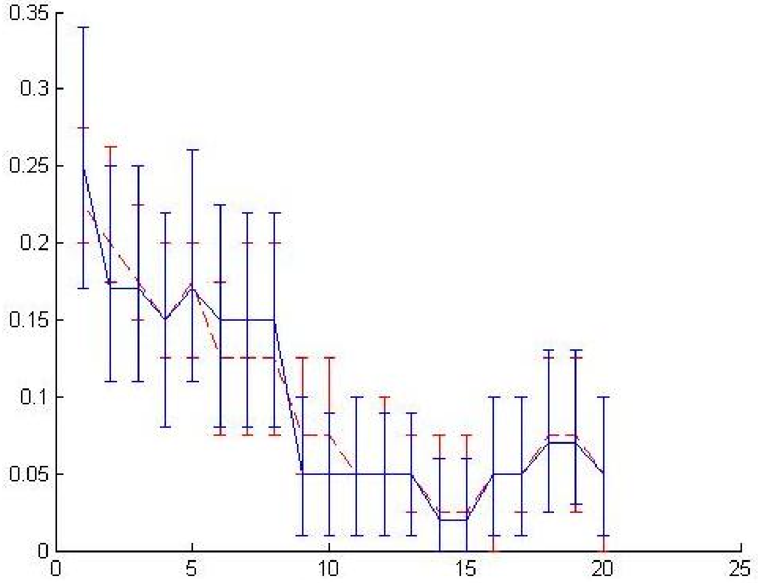

Fig. 2. Error rates for CS06 dataset and RFE method
Rys. 2. Błąd klasyfikacji dla zbioru CS06 i metody RFE

3.2.2. Biological data

For the biological dataset we applied exactly the same methods as in the case of simulated dataset. First, we calculated raw p-values form the t-test estimated from 10 000 permutation, corrected by Benjamini & Hochberg correction algorithm with the significance level α which equaled 0,01 and with Bonferroni correction method with the same level α. Before we applied the correction methods, in the dataset there were 861 genes of the p-value less than 0,01. After Benjamini & Hochberg correction 193 genes were discovered as significant genes. We had another situation when we applied Bonferroni correction. Only 112 genes were discovered as differentially expressed in the leukemia dataset. All of these genes were also discovered by BH method. When we applied RFE method in the first 50 genes, there were only 6 genes discovered previously by BH or B methods. In the first 193 genes there were 36 genes discovered previously by BH and B. It is gene X59350_at (no. 4324). In first 1 000 genes there were 117 common genes for BH and RFE and 70 common genes for B and RFE. Tables of common genes for B,BH and RFE are given in a table 2.

For BH and B we noted the best classification result for 24 genes and it equaled 0,029 in confidence interval equaled [0,02; 0,04]. The mean error rate was not that satisfactory. It equaled 0,136 in confidence interval [0,117; 0,156]. The worst error rate was for 3 genes and equaled 0,41 in confidence interval [0,38; 0,443]. For RFE classification, the result was the best for the first 24 genes. The error rate equaled 0,03 in confidence interval [0,019; 0,04]. We noted the worst result for 3 genes. The error rate equaled 0,413 in confidence interval [0,383; 0,442]. We noted an average error rate for the first 17÷21 genes. The error rate equaled 0,136 in the confidence interval [0,117; 0,156]. The result for k-fold cross validation was very similar. The classification rates were shown in the fig. 3.

Table 2

Common genes in B, BH and RFE methods for leukemia dataset

No.	Id.	Gene name
1	804	'HG1612-HT1612_at'
2	1133	'J04990_at'
3	1249	'L08246_at'
4	1674	'M11147_at'
5	1704	'M13792_at'
6	1882	'M27891_at'
7	1909	'M29696_at'
8	2121	'M63138_at'
9	2288	'M84526_at'
10	2354	'M92287_at'
11	2402	'M96326_rna1_at'
12	2642	'U05259_rna1_at'
13	4196	'X17042_at'
14	4211	'X51521_at'
15	4328	'X59417_at'
16	4847	'X95735_at'
17	5191	'Z69881_at'
18	5772	'U22376_cds2_s_at'
19	6200	'M28130_rna1_s_at'
20	6201	'Y00787_s_at'

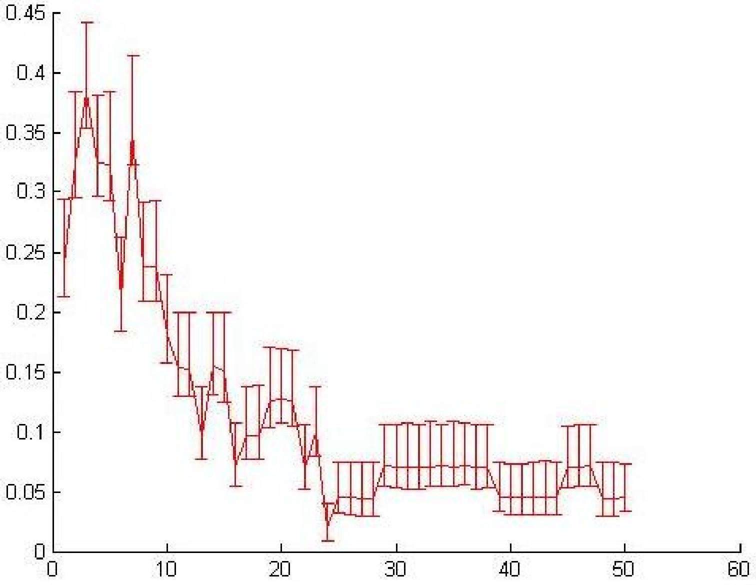

Fig. 3. Error rates for leukemia dataset and RFE method
Rys. 3. Błąd klasyfikacji dla zbioru leukemia i metody RFE

4. Conclusions

To sum up, this paper compared some existing methods for discovering differential expressed genes, from both filter and optimal subset selection groups, applied to the simulated and biological datasets. Each method gave a different set of features. For the leukemia dataset the difference between methods was more visible. Only 6 genes were common for all methods. It suggested of using some kind of preprocessing. For these 6 genes the classification error rates were very good, so it can implied that these genes were truly differential expressed genes.

There was no simple answer which multiple testing procedure to use because the results were very similar to them. Some of the DEGs were not discovered, especially in the IS05 datasets. It could be because the corrections of the p-values were too strong. We also investigated the performance of the methods for the classification. The obtained classification rates across the compared filter methods were very similar. The classification rates using genes obtained from the optimal subset selection group were worse. The classification rate was surprisingly low when we used the genes selected by RFE algorithm for the IS01 dataset.

In conclusion, there were a large number of methods selecting differential expressed genes in the literature. We compared only some of them, we obtained very similar results for the filter methods. Of course the results could depended on the test used. We used the standard t-test, which had some limitations. The main disadvantage was the restriction of only two classes in the dataset. In the real microarray experiments we often had to find DEGs in the datasets where the number of classes was more than two. In the future we would like to compare some methods for doing so, and propose some improvements.

At the end we would like to thank prof. Andrzej Świerniak for valuable comments.

BIBLIOGRAPHY

1. Broberg P.: Statistical methods for ranking differentially expressed genes. Genome Biology 2003, 4:R41.
2. Dudoit S., Fridlyand J., Speed T. P.: Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data. Journal of American Statistical Association, 2002, Vol.97, No. 457.
3. Dudoit S., Yang Y. H., Callow M. J., Speed T .P.: Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. Technical Report 578, Department of Statistics, UC Berkeley, CA, 2000.

4. Efron B.: Estimating the error rate of prediction rule improvement on cross-validation. Journal of the American Statistical Association, 1983, Vol. 78, No. 382.
5. Ge Y., Dudoit S., Speed T. P.: Resampling-based multiple testing for microarray data analysis. Technical Report 663, Department of Statistics, UC Berkeley, CA, 2003.
6. "Multiple Testing Corrections", Silicon Genetics 2003.
7. Guyon I., Weston J., Barnhill S.: Gene Selection for cancer classification using support vector machines. Machine Learning, 2002, Vol. 46, pp. 389÷422.
8. Kohavi R., John G. H.: Wrappers for feature subset selection Artificial Intelligence, 1997 pp. 273÷324.
9. Storey J. D., Tibshirani R.: Statistical significance for genome wide studies. PNAS 2003, Vol. 100, No. 16.

Recenzent: Dr inż. Krzysztof Fujarewicz

Wpłynęło do Redakcji 4 września 2006 r.

Omówienie

W artykule przedstawiono porównanie metod selekcji cech zastosowanych do wykrywania genów różnicujących w eksperymentach mikromacierzowych. Wśród omawianych metod są metody statystyczne, takie jak: ranking cech wykorzystujący test T z poprawką Benjamini & Hohberga [7], zapewniającą kontrolę błędu pierwszego rodzaju FDR na określonym poziomie dla wielokrotnego testowania hipotez. Ponadto, porównanie uwzględnia metody poszukiwania optymalnego podzbioru cech, takie jak algorytm Recursive Feature Elimination (RFE) opisany w [6]. Oceniono również przydatność wybranych przez omawiane metody genów dla klasyfikacji szacując błąd i przedziały ufności dla błędu dwoma metodami: bootstrapową (Jackknife) oraz k-fold cross validation. Porównania dokonano przy użyciu specjalnie symulowanych zbiorów mikromacierzowych, zawierających różne liczby genów różnicujących czy zakładających zależności korelacyjne pomiędzy genami na ustalonym poziomie. Na zbiory symulowane składają się poziomy ekspresji 2000 genów w 30 próbkach dla zbiorów IS01, IS05 natomiast dla zbiorów CS02 i CS06 w 40 próbkach, pochodzących z dwóch klas. Wykorzystano również dostępne pod adresem http://www.broad.mit.edu dane biologiczne, dotyczące dwóch typów białaczki: acute lybphoblastic leukemia (ALL) and acute myeloid leukemia (AML). Choć wybrane geny różnicujące za pomocą omawianych metod często się różnią, nie zaobserwowano istotnej przewagi któreś z metod selekcji dla klasyfikacji, błędy dla różnych metod były na podobnym poziomie, ponadto otrzymane metodą Jackknife i Cross Validation nie różniły się znacznie.

Adresses

Katarzyna STĄPOR: Politechnika Śląska, Instytut Informatyki, ul. Akademicka 16,
44-100 Gliwice, Polska, katarzyna.stapor@polsl.pl

Paweł BŁASZCZYK, Uniwersytet Śląski, Instytut Matematyki, ul. Bankowa 14, 40-007
Katowice, Polska, pblaszcz@math.us.edu.pl

Adrian BRÜCKNER, Uniwersytet Śląski, Instytut Matematyki, ul. Bankowa 14, 40-007
Katowice, Polska, abruckner@math.us.edu.pl