IMPROVED GENE SELECTION FOR CLASSIFICATION OF MICROARRAYS

J. JAEGER[*], R. SENGUPTA[*†], W.L. RUZZO[*†]

[*]*Department of Computer Science & Engineering*
University of Washington
114 Sieg Hall, Box 352350
Seattle, WA 98195, USA

[†]*Department of Genome Sciences*
University of Washington
1705 NE Pacific St, Box 357730
Seattle, WA 98195, USA

In this paper we derive a method for evaluating and improving techniques for selecting informative genes from microarray data. Genes of interest are typically selected by ranking genes according to a test-statistic and then choosing the top k genes. A problem with this approach is that many of these genes are highly correlated. For classification purposes it would be ideal to have distinct but still highly informative genes. We propose three different pre-filter methods — two based on clustering and one based on correlation — to retrieve groups of similar genes. For these groups we apply a test-statistic to finally select genes of interest. We show that this filtered set of genes can be used to significantly improve existing classifiers.

1   Introduction

Even though the human genome sequencing project is almost finished the analysis has just begun. Besides sequence information, microarrays are constantly delivering large amounts of data about the inner life of a cell. The new challenge is now to evaluate these gigantic data streams and extract useful information.

Many genes are strongly regulated and only transcribed at certain times, in certain environmental conditions, and in certain cell types. Microarrays simultaneously measure the mRNA expression level of thousands of genes in a cell mixture. By comparing the expression profiles of different tissue types we might find the genes that best explain a perturbation or might even help clarify how cancer is developing.

Given a series of microarray experiments for a specific tissue under different conditions we want to find the genes most likely differentially expressed under these conditions. In other words, we want to find the genes that best explain the effects of these conditions. This task is also called feature selection, a commonly addressed problem in machine learning, where one has class-labeled data and wants to figure out which features best discriminate among the classes. If the genes are the features describing the cell, the problem is to select the features that

have the biggest impact on describing the results and to drop the features with little or no effect. These features can then be used to classify unknown data. Noisy or irrelevant attributes make the classification task more complicated, as they can contain random correlation. Therefore we want to filter out these features.

Typically, informative genes are selected according to a test statistic or p-value rank according to a statistical test such as the t-test. The problem here is that we might end up with many highly correlated genes. Besides being an additional computational burden, it also can skew the results and lead to misclassifications. Additionally, if there is a limit on the number of genes to choose we might not be able to include all informative genes. Our approach is to first find similar genes, group them and then select informative genes from these groups to avoid redundancy.

Besides t-like-statistics, there are many different techniques applicable. There are non-parametric tests like TNoM[1] (which calculates a minimal error decision boundary and counts the number of misclassifications done with this boundary) or Wilcoxon rank sum/Mann-Whitney[2] (which test statistic is identical[3] to the Park[4] score). It creates a minimal decision boundary too, but incorporates the distance from the boundary into the score. T-like statistics such as Fisher[5] and Golub[6] put different weights in the variance and number of samples. The Mutual Information score results from entropy and information theory, and the B-score[7] comes from Bayesian decision theory. Vapnik[8] describes an interesting method to optimize feature selection while generating support vector boundaries for SVMs.

In this paper we will compare classification done with five different test statistics: Fisher,[5] Golub,[6] Wilcoxon,[2] TNoM,[1] and t-test[2] on three different publicly available datasets, Golub,[6] Notterman[16] and Alon[9]. We will propose two algorithms based on clustering and one based on correlated groups to find similar genes. We then show that these prefiltering methods yield consistently better classification performance than standard methods using similar numbers of genes.

The rest of the paper is organized as follows: Section 2 will review important methods needed for our proposed approach. Section 2.1 will introduce Microarrays. Section 2.2 describes a method for evaluating different feature selection sets using support vector machines and a technique called leave-one-out cross-validation. In section 2.3 we will review clustering techniques used in this paper. In section 2.4 we will discuss how reducing redundancy in a dataset can help with the final classification process and elucidate why redundancy can cause problems for classification tasks. In section 2.5 we propose new methods to select genes from clusters using correlation, clustering and statistical information about the genes. In section 3 we will present results using our proposed approach on three publicly available data sets. Section 4 contains conclusions and future research directions.

2 Methods and Approach

2.1 Microarrays

A typical microarray holds spots representing several thousand to several tens of thousands of genes or ESTs (expressed sequence tags). After hybridization the microarray will be scanned and converted into numerical data. Finally the data should be normalized. The purpose of this step is to counter systematic variation (e.g. difference in labeling efficiency for different dyes, compensation for signal spill over from neighboring spots[10]) and to allow a comparison between different microarrays[11]. The data we work with is already background-corrected and normalized, and we do not address these problems in this paper.

2.2 Validation, Classification

As we are proposing new gene selection schemes we want to measure their performance and allow a comparison. All schemes provide us with a set of informative genes that will be used for future classification. Assume we have n samples. Leave-one-out cross-validation (LOOCV) is a technique where the classifier is successively learned on n-1 samples and tested on the remaining one. This is repeated n times so that every sample was left out once. To build a classifier for the n-1 samples, we extract the most revealing genes for these samples, and use a machine learner. With this classifier we try to classify the remaining (left out) sample. Repeating this procedure n times gives us n classifiers in the end. Our error score is the number of mispredictions. We use support vector machines (SVMs[12]) as the classification method as these are very robust with sparse and noisy data.

2.3 Support Vector Machines

Supports Vector machines (SVMs) expect a training data set with positive and negative examples as input (i.e., a binary labeled training data set). Then they create a decision boundary (the maximal-margin separating hyperplane) between the two groups and select the most relevant examples involved in the decision process (the so-called support vectors). The construction of the hyperplane is always possible as long as the data is linearly separable. If this is not the case, SVMs can use 'kernels', which provide a nonlinear mapping into a higher dimensional feature space. If a separating hyperplane in this feature space is found, it can correspond to a nonlinear decision boundary in the input space. If there is noise or inconsistent data a perfectly separating hyperplane may not exist. Soft-margin SVMs[12] attempt to separate the training set with a minimal number of

errors. In this paper we used Bill Noble's SVM implementation 1.3 beta now called gist[13].

2.4 Clustering

Cluster analysis is a technique for automatically grouping and finding structures in a dataset. Clustering methods partition the dataset into clusters, where similar data are assigned to the same cluster whereas dissimilar data should belong to different clusters. Fuzzy clustering[14] deals with the problem that there is often no sharp boundary between clusters in real applications. Instead of assigning an element to one specific cluster there is a membership probability for each cluster. In doing so an element can be a member of several clusters. Fuzzy clustering can be seen as a generalization of k-means[15] clustering. We used the FCMeans Clustering MATLAB Toolbox V2-0.

2.5 Reducing Redundancy

Now that we have reviewed all the methods we need, we will return to the problem of feature selection. Table 1 shows a list of 7 genes from Notterman's Adenoma[16] dataset sorted by increasing p-value. For gene M18000 the expression value is generally higher in Adenoma than in Normals with the exception of Adenoma 1 and Normal 2. Looking at X62691 the same is true. Both genes have a very low p-value and would be pulled out by conventional methods, which focus on genes with the lowest p-values. Biologists are often interested in a small set of genes (for financial, personal workload or experimental reasons) that describes the perturbation as well as possible. Therefore we are limited in the number of genes to extract and e.g. assuming we could only extract 2 genes we would pull out the first two genes as they have the lowest p-value. However we would not get much additional information using the second gene as it shows the same overall pattern. It would be better to include a gene that provides us with extra information.

Table 1: Expression values for 7 selected genes of Adenoma and normal tissues, sorted by p-value.

Accession Number	Adenoma				Normal				t-test p-value
	1	2	3	4	1	2	3	4	
M18000	705.41	1227.27	959.35	951.56	359.83	711.08	485.33	431.19	0.014
X62691	387.91	577.57	578.45	546.54	227.26	436.65	306.94	239.33	0.016
M82962	91.85	16.27	12.61	61.62	187.44	76.90	181.38	186.53	0.017
U37426	0.47	7.05	6.30	3.40	-3.88	1.58	-2.99	-2.91	0.018
HG2564	2.33	0.54	1.58	3.82	-2.91	-2.11	1.00	-2.91	0.019
Z50853	35.43	26.03	51.49	41.22	27.68	15.80	12.46	15.99	0.022
M32373	-48.02	-28.20	-64.62	-56.95	-15.05	-16.86	-7.97	-34.88	0.022

Table 2: Correlation between Adenoma genes from table 1

	M18000	X62691	M82962	U37426	HG2564	Z50853	M32373
M18000	1.000						
X62691	0.961	1.000					
M82962	-0.944	-0.971	1.000				
U37426	0.973	0.975	-0.983	1.000			
HG2564	0.592	0.653	-0.553	0.529	1.000		
Z50853	0.514	0.616	-0.633	0.597	0.614	1.000	
M32373	-0.509	-0.590	0.602	-0.580	-0.619	-0.874	1.000

Looking at the correlation values in table 2 we can see that the first four genes have an absolute correlation greater than 0.94. Not surprisingly highly correlated genes show the same misclassification pattern and in fact we find that the first four genes also have the same pattern of consistent outliers in Adenoma 1 and Normal 2. In order to increase the classification performance we propose to use more uncorrelated genes instead of just the top genes. We expect the phenomenon illustrated by this example to be a general one. By just using the k best ranking genes according to a test-statistic we would select highly correlated genes. Correlation can be a hint that the two genes belong to the same pathway, are coexpressed or are coming from the same chromosome. In general we expect high correlation to have a meaningful biological explanation. If, e.g., genes A and B are in the same pathway it could be that they have similar regulation and therefore similar expression profiles. If gene A has a good test score it is highly likely that gene B will, as well. Hence a typical feature selection scheme is likely to include both genes in a classifier, yet the pair of genes provides little additional information than either gene alone. Of course we could just select more genes in order to capture all relevant genes. But not only would more genes involve higher computational complexity for classification but it also can skew the result if we have a lot more genes from one pathway. Furthermore if there are several pathways involved in the perturbation but one pathway has the main influence, we will probably select all genes from this pathway. If we then have a limit for the number of genes we might end up with genes only from this pathway. If many genes are highly correlated we could describe this pathway with fewer genes and reach the same precision. Additionally, we could replace correlated genes from this pathway by genes from other pathways and possibly increase the prediction accuracy. The same issue might be true when selecting a lot of genes as well, but it is more compelling when we have a limited budget of genes and can only select a few genes.

Our method for gene selection will therefore be to prefilter the gene set and drop genes that are very similar. For the remaining genes we will apply a common test statistic and pull out the highest-ranking genes. One way to find correlated

genes would be to calculate the correlation between all genes. In our first method we selected from the best genes (best according to a test statistic) those that have a pair-wise correlation below a certain threshold. A simple greedy algorithm accomplishes this selection – the k-th gene selected is the gene with highest p-value among all genes whose correlation to each of the first k-1 is below the specified threshold. This method is called "Correlation" in the figures below. Greedy algorithms are of course simple, but often give results of poor overall quality due to their myopic decision-making. As an alternative allowing a more global view of the data, we also consider clustering algorithms. Clustering is very versatile as it can use different distance functions (Euclidean, L_k, Mahalanobis, and correlation), and different underlying models, shapes and densities, which are not captured by just correlation. In this paper we compared clustering and correlation methods. We used a fuzzy clustering algorithm because it assigns a membership probability for a cluster for each gene and may therefore capture the fact that some genes are involved in several pathways. Although a cluster does not automatically correspond to a pathway it is a reasonable approximation that genes in the same cluster have something to do with each other or are directly or indirectly involved in the same pathway. Our basic approach is to cluster the genes, and then to select one or more representative genes from each cluster. The details how many genes from which cluster depend on the "quality" of each cluster and will be discussed below.

2.6   Assigning quality to cluster

Once we have done the clustering we know that genes in a cluster show similar expression profiles and might be involved in the same pathway. Since we want to have as many pathways as possible involved in our list of significant genes, we would like to sample from each cluster/pathway. But it would not be fair to treat each cluster and gene equally. The size of the clusters as well as the quality of a cluster play a role, i.e. how close together are the genes, how far away are they from the cluster center. If a cluster is very tight and dense it can be assumed that the members are very similar. On the other hand if a cluster has wide dispersion the members of the cluster are more heterogeneous. To capture the biggest possible variety of genes, it would therefore be favorable to take more genes from a cluster of bad quality than from a cluster with good quality. To determine the quality for the fuzzy clustering algorithm we used the membership probability for a gene. We said that an element belongs to the cluster to which it has the highest membership probability. The cluster quality is then assessed by looking at the average membership probability of its elements.

A high cluster quality means low dispersion, and the closer the quality gets to 0 the more scattered the cluster becomes. In our first clustering algorithm we decided that no matter how bad the quality and how small the size of the cluster

we should get at least one element from each cluster. Our reasoning is as follows. First, if the cluster size is very small but there is a very good gene in it, we do not want to miss that cluster. Second, by eliminating a cluster we lose all the information of that pathway, so getting at least one representative plays a role like pseudocounts. Third, if we have a cluster that is extremely correlated, we would have a very high quality score and therefore may not pick any gene from that cluster. But this one gene from that cluster might have had a very good contribution to the discrimination process.

The drawback is that a cluster might represent a pathway that is totally unrelated to the discrimination we look for. If the cluster then has a bad quality we might pick a lot of genes from that cluster even though they are not informative. To counteract this problem we implemented the possibility to mask out and exclude clusters that have an average bad test statistic p-value (this method is called "Masked out Clustering" in the figures, whereas "Clustering" refers to the method where we look at all clusters and do not mask out any). Lastly we want to have genes that have a high discriminatory power, i.e. can explain the symptoms. This can be achieved by using an appropriate test statistic.

3   Results and Discussion

For our experiments we selected three different publicly available microarray datasets, Alon[9](40 Adenocarcinoma and 22 normal samples), Golub[6](47 ALL and 25 AML leukemia samples) and Notterman[16](18 tumor and 18 normal samples). We compared five different test statistics: Fisher[5], Golub[6], Park[4], TNoM[1], and t-test and ran our three different filtering algorithms described above: Correlation, Clustering, Masked out Clustering. The performance of the feature selection was calculated using SVM and LOOCV scores.

For each possible combination of test statistic and clustering algorithm we evaluated the performance varying the number of clusters between 1 and 30 and the number of selected features between 2 and 100. We used the Euclidean distance metric and a fuzzy clustering softness of 1.2 (where 1 would be hard clustering and infinity is everything belonging to all clusters). For SVM we chose an RBF kernel function and used data normalization in the feature space.

We also calculated the LOOCV performance using all available data (i.e. not reducing the number of genes but using all of them). The result was that LOOCV produced 6 false classifications in the Alon's colon dataset, resulting in an error of 9.7%. In Golub's leukemia dataset LOOCV produced 2 false classifications, resulting in an error of 2.8% and in Notterman's carcinoma dataset LOOCV made 1 false classification, resulting in an error of 2.8%.

Figure 1 shows a 3d plot of the LOOCV performance varying the number of clusters selected between 1 and 10 and the number of features chosen between 10

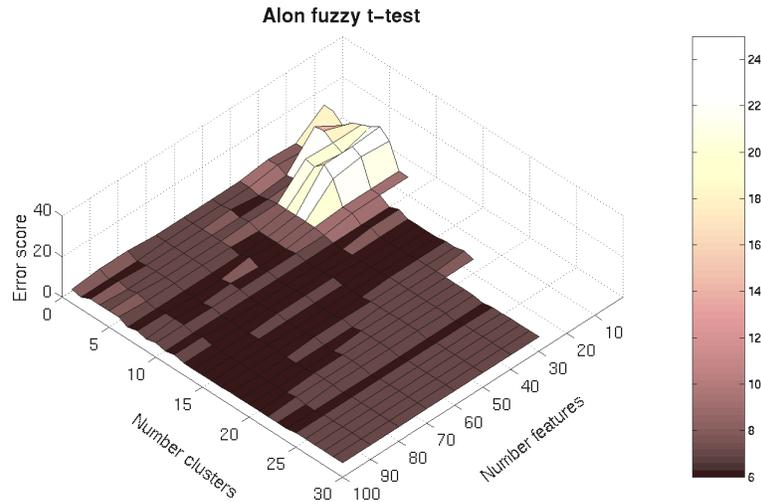and 100 in steps of 10. The plot shows the clustering algorithm without masking out.

Figure 1: LOOCV performance for Alon's data set using clustering and conventional methods

There are no values for 10 features and more than 10 clusters, as well as 20 features and more than 20 clusters, as one of our constraints is that each cluster has at least one member in the feature set. So we can never have more clusters than features selected.

In the leftmost ribbon (starting in the lower left corner and going up to the top left corner) we can see the performance varying the number of features and using just one cluster (i.e. this is our standard comparison line, as this reflects just selecting features by test-statistic). The whole plot seems very flat once we have more than 30 features. But notice the darker spots in the middle (between 10 and 20 clusters), that reflect very low LOOCV scores. One reason that there is a high peak for less than 30 features is that the t-test selects highly correlated and therefore redundant genes, which makes it hard for the underlying SVM to learn a good classifier. The average correlation of the top 10 genes selected with the t-test is 0.85.

Figure 2 compares different test statistics on a given data set using the first clustering algorithm. Notice that Fisher and Golub behave very similarly as do Park and TNoM, but t-test has (besides the big bulk at only a few features) a very flat and robust behavior. Fisher and Golub seem to have a higher variance in class-
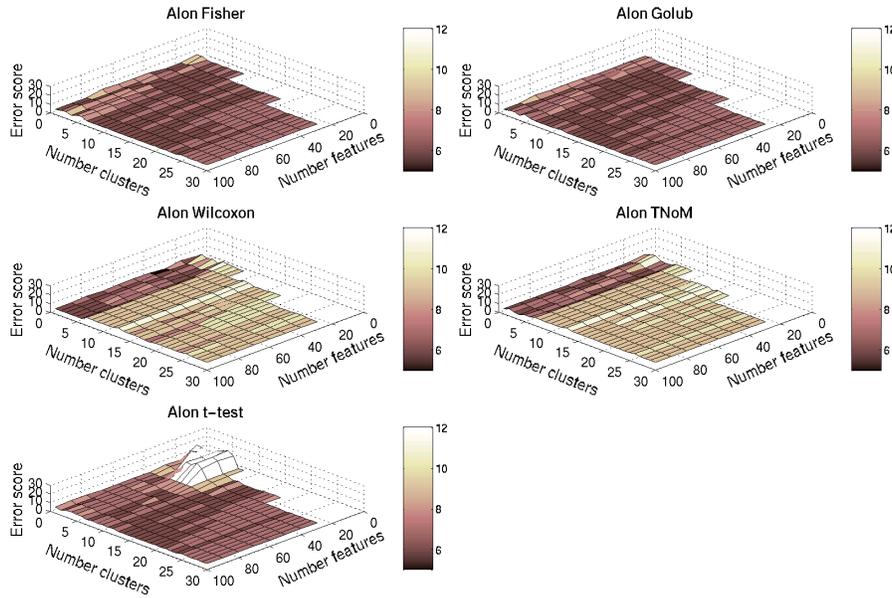
Figure 2: Comparison of different test statistics

ification but their best classification performance is similar to t-test. They achieve their best results with 6-25 clusters. TNoM and Park achieve their best results for fewer clusters (in the range of 1-6) and in fact they seem not to benefit from the clustering as much as t-test, Fisher or Golub do.

For Notterman's carcinoma dataset the standard classification process already achieves 0% error when using more than 10 features but we can still improve the classification when using 10 features or less. Using only that few features we still manage to have a 0% error with most of the test statistics. Due to space restrictions figures are not shown here but can be accessed online[17].

Now consider how the LOOCV performance of our clustered result compares to the conventional methods (depicted as normal). In figure 3 we plot the normal score, the clustered scores (the minimum error score over all trials with cluster size from 2 to 30), the clustered scores with masking out and the correlation scores for the LOOCV performance for each of the five test-statistics. Here we did not plot the number of errors (plots for this are available online[17]), that reflect the efficiency of the classification but a ROC[18] (receiver operator curves) score (i.e., the area under the ROC graph, which takes both false negative and false positive errors into account and reflects the robustness of the classification). We can see that almost always the filtered performance is better than the conventional method. Another noticeable fact is that without clustering, TNoM would have on average the best LOOCV performance of the five scores for Alon's colon dataset.

An explanation might be that TNoM, as a nonparametric test, extracts less correlated genes and therefore already does a good job in selecting different genes.
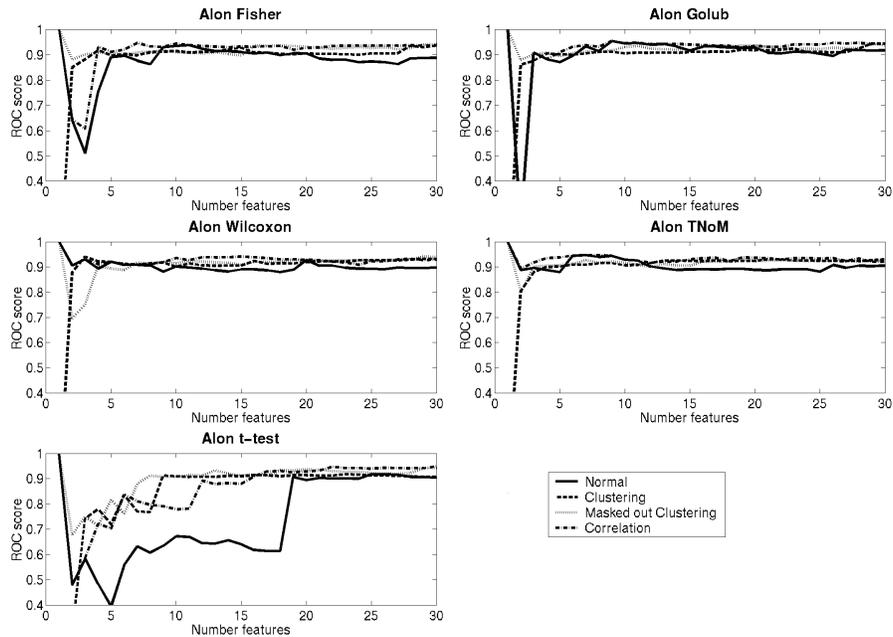

Figure 3: Comparison of test-statistics and different prefilter methods for Alon's data set

In the top 30 t-test genes in Alon's dataset, we have an average correlation of 0.79, whereas in the top 30 TNoM genes, we have only an average correlation of 0.56. Wilcoxon (also a nonparametric test) achieves the best result of all the tests (when comparing 30 features) and can reduce the absolute LOOCV error to 5.[17] The average correlation of the top 30 Wilcoxon genes is 0.43.

Doing the same comparison on Golub's dataset yields figure 4. In Alon's dataset it seemed that TNoM was on average the best test statistic whereas in Golub's leukemia dataset Wilcoxon performs best. We still see that clustering can lower the error in most of the cases. A remarkable fact is that with clustering we achieve 0% error with the t-test when using more than 50 features. Notice that we had 2 errors when using all the data. Here, not only can feature selection reduce the number of genes to find, but it can also decrease the error.

Although clustering for feature selection generally seems to improve the LOOCV error, the least improvements were obtained using it in conjunction with the Wilcoxon rank sum test and the best performance improvement was achieved using it together with t-tests. As illustrated above the reason for that could be that the t-test generally finds more correlated genes. The nonparametric tests do not

take values into account and calculate their scores purely based on rank information what seems to have a positive effect on selecting fewer correlated genes.
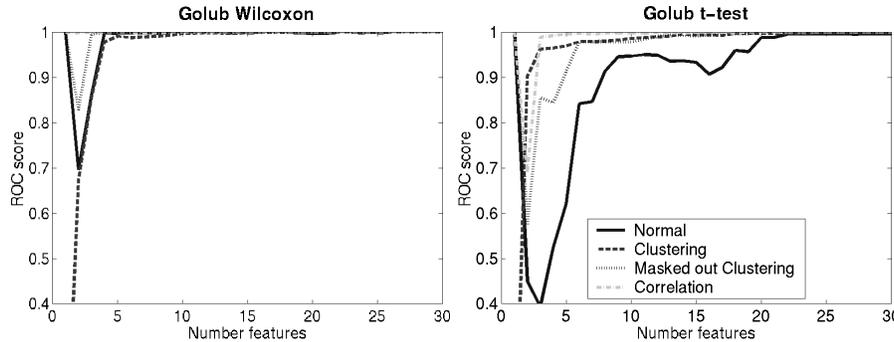

Figure 4: Comparison of different prefilter methods for Golub's data set

4   Conclusion and further research

In this paper we presented three novel prefilter methods to increase classification performance with microarray data. There is no clear winner between the three proposed methods and it depends largely on the dataset and parameters used.   All the proposed feature selection methods find a subset that has better LOOCV performance than the currently used approaches. One question not addressed here is how to find the correct number of clusters. It is pretty expensive to try all possible numbers for clusters to find a setting that provides us with a good LOOCV performance. One direction for future work would be to estimate the number of clusters using a BIC[19] (Bayesian Information Criterion) score or switching over to model based clustering[20].

We addressed the problem of feature selection and outlined why feature selection has to be done and how it can be done without losing crucial information. The question not answered here is how many features/genes should be chosen in the end. One could argue to choose exactly that many genes as necessary to achieve the lowest LOOCV error. But in the end it comes down to a tradeoff between false positives and false negatives. The more genes we have in our set of interest (the feature set) the more genes might also be real candidates. The extreme would be to take all genes. Then we definitely have all candidates but also a very high percentage of false positives. The other extreme would be to take no gene at all. Then we would not mispredict anything to be a real gene but would have a very high false negative rate. The final answer on how many genes to select can only be answered by the biologists who must judge how much time

they can invest in examining these genes further and which false positive/negative rate they will accept.

We feel, however, that for any fixed size the methods outlined here are likely to identify sets of genes that are stronger predictors than sets found by standard methods, which should be of significant value for diagnostic purposes as well as for guiding further exploration of the underlying biology.

Acknowledgment

References

1   A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. *RECOMB*, (2000).
2   J.L. Devore, Probability and Statistics for Engineering and the Sciences, 4th edition, Duxbury Press, (1995).
3   Own unpublished results
4   P.J. Park, M. Pagano, M. Bonetti: A nonparametric scoring algorithm for identifying informative genes from microarray data. *PSB*:52-63, (2001).
5   C.M. Bishop: Neural Networks for Pattern Recognition, *Oxford University Press*, (1995)
6   T.R. Golub, D.K. Slonim, et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, **286**:531-537, (1999).
7   I. Lonnstedt and T. P. Speed. Replicated Microarray Data. *Statistical Sinica*, (2002).
8   J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs, *Advances in Neural Information Processing Systems* **13**. MIT Press, (2001).
9   U. Alon, N. Barkai, D.A. Notterman, K. Gish, S. Ybarra, D. Mack and A.J. Levine: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays **PNAS 96**:6745-6750, (1999).
10 Y.H. Yang, M.J. Buckley, S. Dudoit, T.P. and Speed: Comparison of methods for image analysis on cDNA microarray data. *Technical report* (2000)
11 Y. H. Yang, S. Dudoit, P. Luu and T. P. Speed. Normalization for cDNA Microarray Data. *SPIE BiOS*, (2001).
12 C. Cortes and V. N. Vapnik. Support vector networks. *Machine Learning*, **20**:273-297, (1995).
13 http://microarray.cpmc.columbia.edu/gist/
14 J.C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters", *Journal of Cybernetics*, **3**:32--57, (1973).
15 A.K. Jain, R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, (1988).
16 D.A. Notterman, U. Alon, A.J. Sierk, A.J. Levine: Transcriptional Gene Expression Profiles of Colorectal Adenoma, Adenocarcinoma and Normal Tissue Examined by Oligonucleotide Arrays, *Cancer Research* **61:**3124-3130, (2001).
17 http://www.cs.washington.edu/homes/jj/psb
18 C.E. Metz, Basic principles of ROC analysis, *Seminars in Nuclear Medicine*, Vol **8**, No. **4**, 283-298, (1978).
19 G. Schwarz: Estimating the dimension of a model. *Annals of Statistics*, **6:**461-464 (1978).
20 K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery and W. L. Ruzzo, Model-based clustering and data transformation for gene expression data, *Bioinformatics* **17**:977-987 (2001).