3 Gene list filtering for improved classification

In this chapter, I investigate which marker genes should be included in a diagnostic gene signature (question 2 from the preface). For this, I examine techniques for filtering univariate gene selection results. Univariate gene selection methods are computationally not demanding and especially helpful for finding small sets of genes that provide good classification accuracy. I suggest to prefilter the ranked list of univariately selected genes by removing highly correlated genes and show that this improves classification accuracy. The first proposed method uses correlation directly, the second and third method make use of clustering techniques.

In univariate gene selection strategies, usually a simple test statistic, e.g. the t-statistic, is used to select differentially expressed genes (see section 1.4.3). As demonstrated later on in this chapter, many of the top scoring genes according to such a test statistic show high correlation amongst each other. We will show that using less correlated but still differentially expressed genes improves classification accuracy.

A high correlation of two genes suggests that they belong to the same transcriptional module, are co-regulated or are coming from neighboring chromosomal locations. In general, high correlation is assumed to have a meaningful biological explanation (Massart-Leën and Massart, 1981). When two genes show a similar expression profile and are highly correlated to the class labels, a typical feature selection scheme is likely to include both genes in a classifier. Yet, the pair of genes provides little additional information for the classifier compared to using either gene alone. In the case of diagnostic chips there is a limitation on the number of genes that can be included. Here, excluding highly correlated genes saves space for other informative genes.

In this chapter, we compare gene selection for classification done with five different test statistics: Fisher (Bishop, 1995), Golub (Golub *et al.*, 1999), Wilcoxon (Wilcoxon, 1945), TNoM (Ben-Dor *et al.*, 2000), and t-score (Student, 1908) on three different publicly available datasets, Golub *et al.* (1999), Notterman *et al.* (2001) and Alon *et al.* (1999). We propose two algorithms based on clustering and one based on correlation to find similar genes. From these groups of similar genes the only representatives are used in a classifier and the rest is filtered out. We show that these prefiltering methods yield consistently better classification performance than standard methods using the same number of genes.

For a comparison of classification accuracy of the different gene selection schemes we measure their performance in leave-one-out cross-validation (LOOCV) (see section 1.4.2). Here, a 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. As classifier we use Bill Noble's SVM (see section 1.4.1) implementation "1.3β" now called `Gist` (http://microarray.genomecenter.columbia.edu/gist).

3.1 Reducing redundancy

We first show that top ranking genes selected by t-scores are highly redundant and conclude that a redundancy filter should be applied. Table 3.1 lists 7 genes from Notterman's Adenoma (Notterman *et al.*, 2001) dataset sorted by decreasing absolute t-scores. In this dataset the mRNA expression of approximately 6600 cDNAs and ESTs were measured in 4 colon adenomas and 4 paired normal colon samples. For gene M18000 and X62691 the expression value is higher in Adenomas than in Normals with the exception of the Adenoma sample in the first column and the Normal sample in the second column of the Normals. Both genes have a high t-score and are selected by methods that select genes by test scores. For example, when selecting the genes according to a high t-score one selects these two genes. Clearly, there is not much additional information for classification by including the second gene. We will show that it is better to include a gene that is not as highly correlated but can improve classification accuracy.

Accession Number	Adenoma				Normal				t-score
M18000	705.41	1227.27	959.35	951.56	359.83	711.08	485.33	431.19	3.55
M82962	91.85	16.27	12.61	61.62	187.44	76.90	181.38	186.53	-3.40
X62691	387.91	577.57	578.45	546.54	227.26	436.65	306.94	239.33	3.33
HG2564	2.33	0.54	1.58	3.82	-2.91	-2.11	1.00	-2.91	3.28
U37426	0.47	7.05	6.30	3.40	-3.88	1.58	-2.99	-2.91	3.27
Z50853	35.43	26.03	51.49	41.22	27.68	15.80	12.46	15.99	3.27
M32373	-48.02	-28.20	-64.62	-56.95	-15.05	-16.86	-7.97	-34.88	-3.16

Table 3.1: Expression values for 7 selected genes of Adenoma and normal tissues, sorted by t-scores.

By using the k highest ranking test statistics genes strongly correlated genes are selected. The correlation values in table 3.2 show that four genes have an absolute correlation greater than 0.94 to the top ranking gene M18000. Not surprisingly highly correlated genes show the same misclassification pattern and in fact these four genes also have the same misclassifications of Adenoma and Normal. In order to increase the classification performance we propose to limit pairwise correlation of signature genes.

3.2 Correlation based filtering

We propose to prefilter the ranked list of differential genes by removing highly correlated genes. One approach, called "Correlation" method, is based on a simple greedy algorithm with the following steps:

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

Table 3.2: Correlation between Adenoma genes from table 3.1

- Let t_i be the test statistic for gene $i, i \in G = \{1, .., P\}$. G is the set of all genes. $R = \emptyset$ is the initially empty set of filtered genes.
- do iterate
  - rank all genes in G according to the test statistic t_i
  - select the top ranking gene j in G, add it to R by setting $R = R \cup j$, and remove all indices of the genes from G that have an absolute correlation above a threshold $0 < c < 1$ to the gene j
- until k top ranking genes are found ($|R| = k$) or G is empty

These k genes are then by construction not correlated more than the threshold c and they are the genes with the highest test statistic fulfilling this criteria. Applying this algorithm with a given c, say $c = 0.8$, to the seven genes in table 3.1 yields the following:
Initially, G includes all genes $G = \{$ M18000, M82962, X62691, HG2564, U37426, Z50853, M32373 $\}$. In the first iteration, M18000 is found as the top ranking gene and added to R, $R = \{$M18000$\}$. Then, the genes M18000, M82962, X62691, and U37426 are removed from G as they have an absolute correlation of more than $c = 0.8$ to the top ranking gene M18000. G now holds the following genes, $G = \{$HG2564,Z50853,M32373$\}$. In the next iteration, the top ranking gene is HG2564 is added to R, $R = \{$M18000, HG2564$\}$. Only HG2564 itself is removed from G. Then, Z50853 is added to R, $R = \{$M18000, HG2564, Z50853$\}$. M32373 and Z50853 are removed from G, resulting in an empty set. Here, the algorithm stops. The top ranking genes fulfilling the condition of a correlation of less than $c = 0.8$ are $R = \{$M18000, HG2564, Z50853$\}$.

3.3 Clustering based filtering

As an alternative method for gene filtering we also consider clustering based algorithms. Cluster analysis is a technique for automatically grouping and finding structures in a dataset (see section 1.4.4). We used the FCMeans Clustering MATLAB Toolbox V2-0.

This is a fuzzy clustering algorithm, which extends k-means clustering by assigning a membership probability to each cluster for each gene.

First, we fuzzy clustered all genes to obtain groups of genes with a similar expression profile. As the optimal number of clusters was not known a priori we varied the number of clusters from 1 to 30. After clustering, each gene i had a membership probability m_{ij} for each cluster j. We defined the set of genes B_j as the genes with their highest membership probability to cluster j. Using B_j we furthermore used the membership probability to obtain a cluster quality score q_j. This score was defined as the average membership probability to cluster j of the genes in B_j. In order to select C genes for a diagnostic signature we selected genes proportionally to cluster size and inversely to cluster quality. From each cluster c_j genes were selected according to this criteria with the following constraints: no more than n_j (the total number of genes in cluster j) could be selected and at least one gene had to be selected from each cluster. Additionally, $\sum c_j$ should be as close to C as possible. Finally, the c_j genes with the highest test score were chosen from each cluster. We also implemented a second clustering method called "masked out clustering" where we did not require to select any genes from clusters with an overall bad test score.

Pseudo algorithm for filtering out a given number of $C = \sum c_j$ genes using fuzzy clustering:

- The fuzzy clustering of the genes yields a membership probability m_{ij} for each gene i and each cluster j.
- $G_i = \{\arg\max_j m_{ij}\}$ is the set of clusters one gene belongs to
- $B_j = \{i : j \in G_i\}$ is the set of genes belonging to cluster j, $n_j = |B_j|$
- $q_j = \dfrac{\sum_{i \in B_j} m_{ij}}{n_j}$ is the quality score of a cluster
- select from each cluster c_j genes: $c_j = min(max(round(1/q_j * n_j * s), 1), n_j)$, with

  $$s: \quad min|\sum_j min(max(round(1/q_j * n_j * s), 1), n_j) - C|,$$

  s is a scalar that ensures that $|\sum c_j - C| \rightarrow min$, and "round" rounds to the nearest integer.

3.4 Applications

For the evaluation of our method we selected three different publicly available microarray datasets: Alon *et al.* (1999), who examined 40 Adenocarcinomas and 22 normal samples with an Affymetrix Hum6000 microarray; Golub *et al.* (1999), who examined 47 ALL and 25 AML leukemia samples with an Affymetrix HU6800 GeneChip; and Notterman *et al.* (2001), who examined 18 adenocarcinomas and 18 normal samples with an Affymetrix HU6500 GeneChip and 4 adenomas versus 4 normal tissue samples with a HU6800 microarray chip. We compared five different test statistics: Fisher, Golub, Wilcoxon, TNoM, and t-score and ran the three filtering algorithms described above: "Correlation", "Clustering", and "Masked out Clustering". The performance of the feature selection was calculated using SVM with leave-one-out cross-validation (LOOCV).

To assess the general classification complexity of each dataset we first calculated the classification accuracy of all datasets with SVM LOOCV when no feature selection was applied and all data was used in the SVM. In Alon's dataset 56 samples were correctly and 6 falsely classified (9.7% error). In Golub's leukemia dataset 70 samples were correctly and 2 falsely classified (2.8% error). Notterman's carcinoma dataset achieved the same classification error of 2.8% with 1 false classification and 35 correct class assignments.

Next, we evaluated the performance for all combinations of the five test statistics and three prefiltering algorithms. Hereby, we varied the number of clusters between 1 and 30 and the number of selected features between 10 and 100. Choosing only one cluster naturally selects all genes into this one cluster. Since we select genes from one cluster according to the test statistic rank, choosing one cluster is equivalent to standard methods without prefiltering that just select genes according to their rank. For the clustering parameters we used the Euclidean distance metric and in fuzzy-clustering a fuzzy clustering softness of 1.2 (where 1 is hard clustering and infinity is everything belonging to all clusters). For SVM we chose an RBF kernel function and used data normalization to zero mean and unit variance in the feature space.

Alon's dataset evaluation – We clustered the Alon data set once with a k-means and once with a fuzzy-k-means clustering. For k-means we used the ratio of the mean without cluster distance to the mean within cluster distance as the cluster quality measure. A noticeable performance loss appeared when using few genes (Fig. 3.1). If no prefiltering was done (which is equal to clustering everything into one cluster) the best performance was achieved using 40-80 genes. For less than 40 features the performance dropped off strongly. A possible explanation is that using a t-score selects highly correlated and therefore redundant genes. The average correlation of the top 10 genes selected with the t-score was 0.85. Clustering, however, in general provided superior results especially in the range between 10 and 20 clusters. Though, this was more pronounced for fuzzy k-means clustering. Comparing k-means results with fuzzy-k-means, the former showed more variation and fuzzy clustering generally achieved a smaller LOOCV error score.

Comparing different test statistics on a given data set using the fuzzy clustering algorithm showed that Fisher and Golub test statistics behaved similarly as did Wilcoxon and TNoM statistics (see section 1.4.3 for a detail of the test scores), but the t-statistic showed a performance drop especially for lower number of genes (Fig. 3.2). Fisher and Golub test statistics had a higher variance in classification but their best classification performance was similar to t-score. They achieved their best results with 6-25 clusters. TNoM and

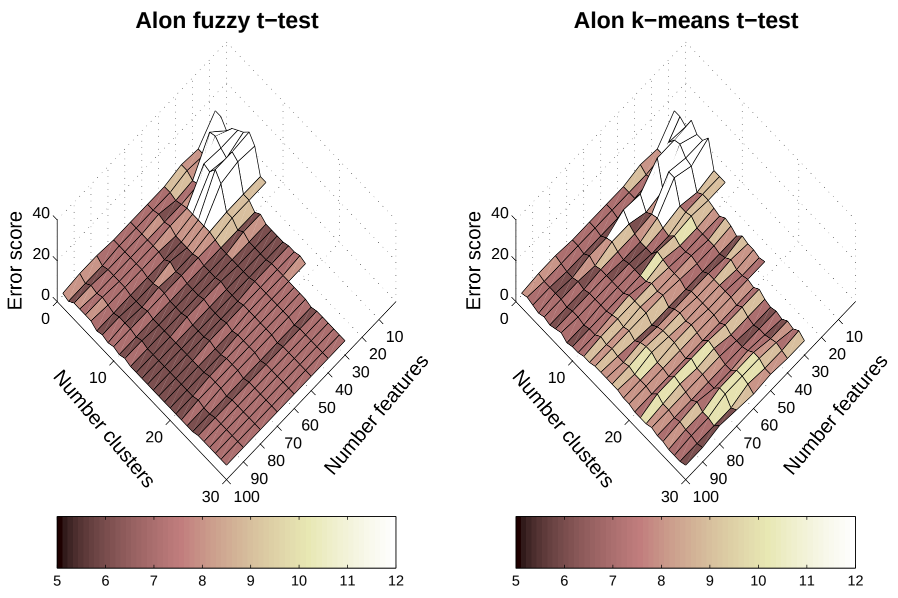

Figure 3.1: LOOCV performance for Alon's data set when using k-means prefiltering (left plot) compared to a fuzzy clustering prefiltering (right plot). The number of clusters were varied between 1 and 30 and the number of features between 10 and 100 in steps of 10. The plots show the clustering algorithm without masking out. There are no values for 10 features and more than 10 clusters, as well as 20 features and more than 20 clusters, since we did require to have at least one member of each cluster in the feature set. So we can never have more clusters than features selected.

Wilcoxon statistics achieved their best results for fewer clusters (in the range of 1-6) and in fact they did not benefit from the clustering as much as t-score, Fisher or Golub test statistics did.

We now consider how the LOOCV performance of our results compared to the conventional gene selection methods on the example of Alon's dataset. Hereby, five methods were compared. The performance achieved with standard gene selection methods ("Normal"), the performance achieved with fuzzy clustering methods (where the minimum error score over all cluster sizes from 2 to 30 was chosen), the performance achieved with clustered methods allowing cluster masking, and the performance of the correlation based method for each of the five test-statistics are compared (Fig. 3.3). Here, we show the area under receiver operator curves (ROC, Metz (1978)), which takes both, false negative and false positive errors into account. The prefiltered gene selection methods did generally perform better than conventional gene selection methods without filtering. Without gene filtering methods TNoM had on average the best LOOCV performance of the five scores for Alon's colon dataset. An explanation is that TNoM, as a nonparametric test, extracts less correlated genes.

The top 30 t-score genes in Alon's dataset had an average correlation of 0.60, whereas the top 30 TNoM genes had only an average correlation of 0.23. Wilcoxon (also a non-parametric test) achieved a similar result (when comparing 30 features) and reduced the absolute LOOCV error to 5.17. The average correlation of the top 30 Wilcoxon genes was 0.23.

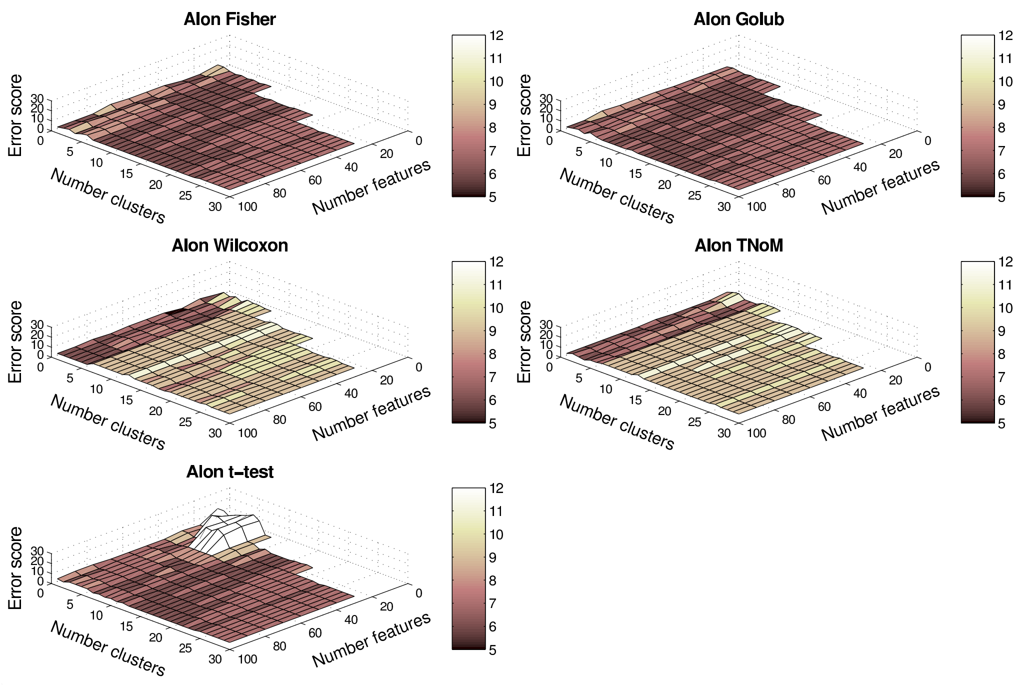

Figure 3.2: Comparison of different test statistics when applying fuzzy-k-means prefiltering to the Alon dataset. The color and z-axis depict the LOOCV error score. Darker colors generally indicate lower error. The x and y-axis denote the number of cluster centers used for the fuzzy-k-means and the number of features used in classification, respectively.

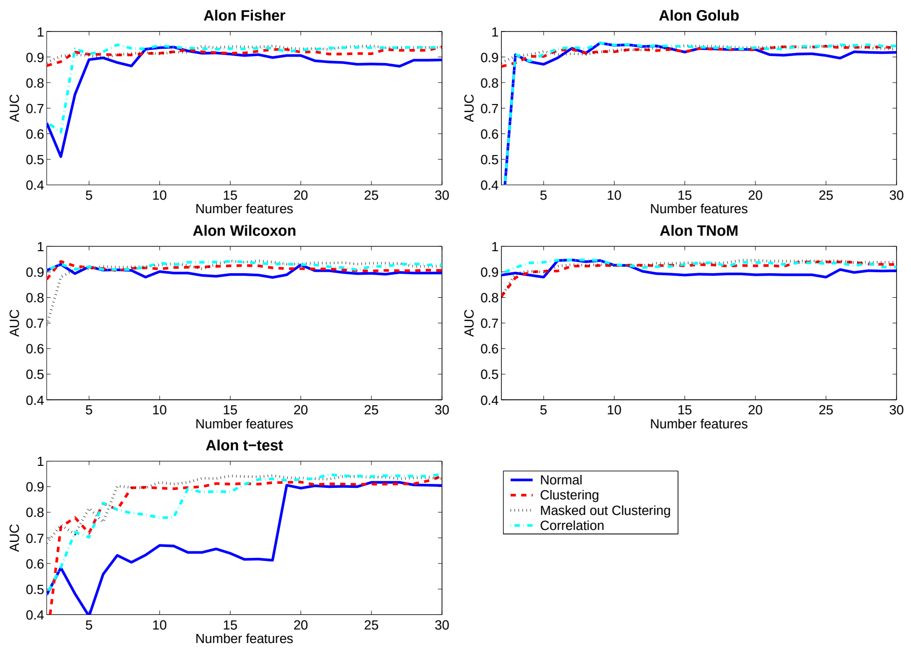

Figure 3.3: Comparison of different prefiltering methods for Alon's data set depending on the number of features. "Normal" depicts conventional gene selection according to the test statistic without filtering. "Clustering" depicts fuzzy k-means clustering. "Masked out clustering" depicts fuzzy k-means clustering where no genes are selected from clusters with average bad test scores. "Correlation" depicts the correlation based prefiltering method. The graphs show the AUC (area under receiver operator curves (ROC)), which combines sensitivity and specificity of classification.

Doing the same comparison on Golub's dataset showed similar results (Fig. 3.5). In Alon's dataset TNoM was on average the best test statistic whereas in Golub's leukemia dataset Wilcoxon performed best. Still, clustering lowered the error in most of the cases. Using clustering with t-score filtering the classification error was 0 when using more than 50 features compared to 2 errors when using all the data.

Notterman data analysis – Analyzing Notterman's data showed that using standard gene selection methods it was already possible to achieve 0% LOOCV error when using more than 10 genes. However, our proposed methods still improved the classification when using 10 features or less. Using only 10 features still resulted in 0% error with most of the test statistics (Fig. 3.4).

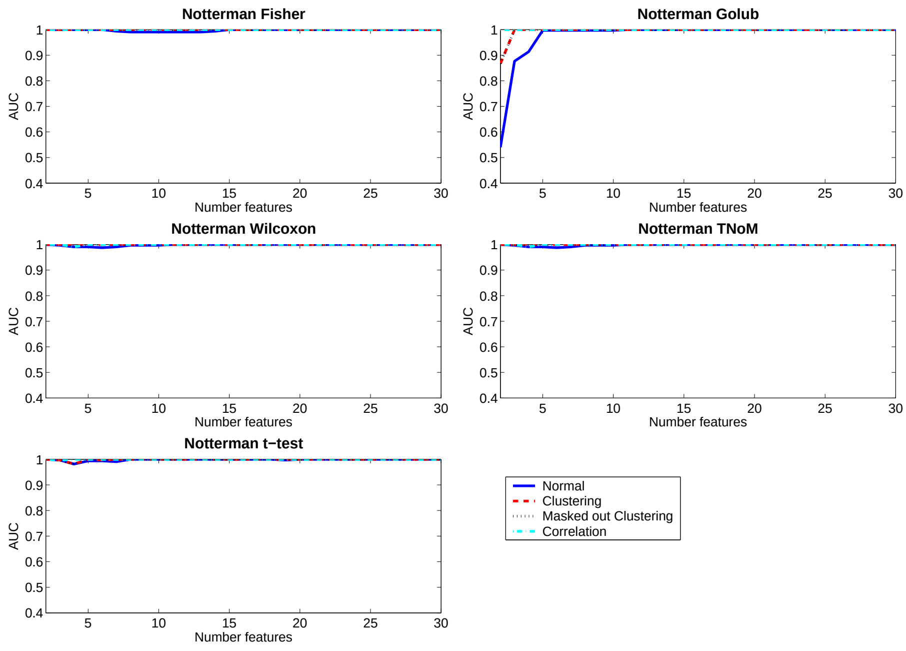

Figure 3.4: Comparison of different prefiltering methods for Notterman's data set depending on the number of features. See Fig. 3.3 for legend description.

Although clustering for feature selection generally improved the LOOCV error, the least improvements were obtained in conjunction with the Wilcoxon rank sum test and the best performance improvement was achieved using it together with t-score statistics. As illustrated above one reason for that is that the t-score generally finds more correlated genes. The nonparametric rank based tests do not use the numeric values directly but calculate their scores purely based on rank information what seems to have a positive effect on selecting fewer correlated genes.

3.5 Discussion

Gene selection for microarray improves classification performance of microarray data and reduces the number of variables used in a classifier. If all features are used, irrelevant

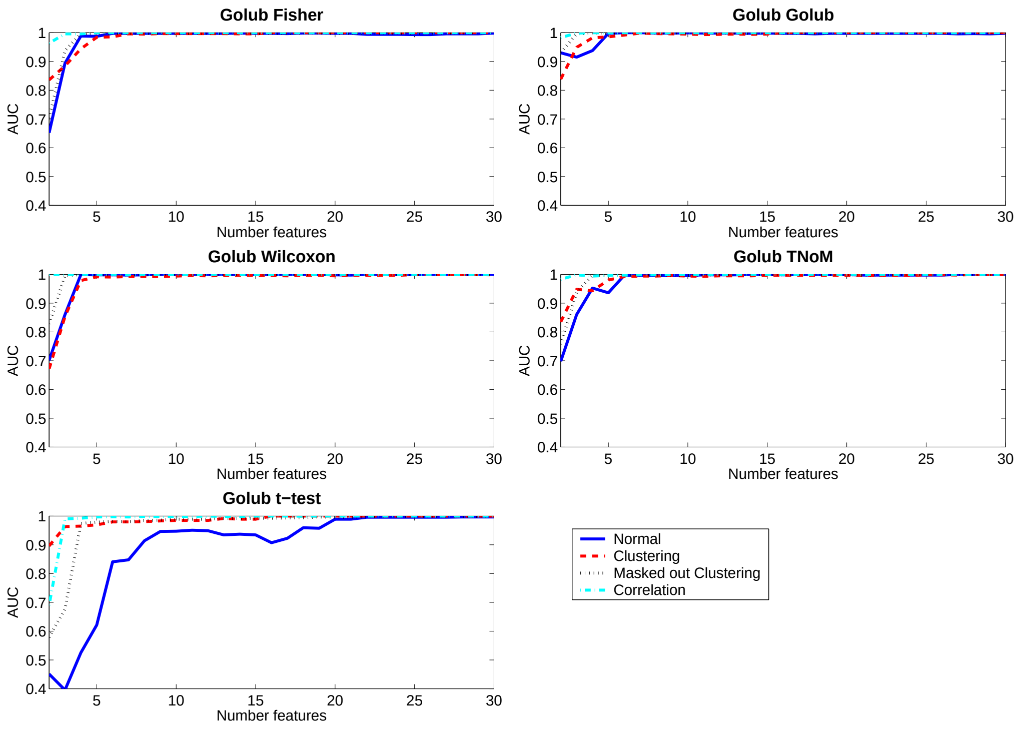

Figure 3.5: Comparison of different prefiltering methods for Golub's data set depending on the number of features. See Fig. 3.3 for legend description.

or noisy features drive the classifier to overfit the data and it will not generalize well to future samples. This is especially true for simple classifiers like kNN but it was also shown that even SVMs do not perform well without feature selection (Weston *et al.*, 2001). Gene selection, on the other hand, does not only help to provide better generalization performance but it also offers computational speed up and can guide the selection of genes that should be put on custom made diagnostic microarrays. The set of selected genes has then already shown to provide a good basis for the discrimination of the groups of interest and can be the basis for a custom diagnostic microarray chip.

Gene selection can be divided in univariate gene selection, looking at each gene separately, or multivariate gene selection considering the influence of multiple genes. In this chapter I have explored a prefilter method for univariate gene selection results aiming for increased classification performance in microarray data analysis. I exemplified its improvement on several univariate test statistics. There is no clear winner between the three proposed prefiltering methods and it depends largely on the dataset and parameters used. However, all the proposed feature selection methods find a subset that has better LOOCV performance than the currently used univariate approaches. Recently, Yu and Liu (2004) proposed a similar approach. They also first ranked their genes according to a relevance score and then filtered out the redundant genes. However, they did not use correlation as a redundancy measurement but an information gain criteria (Quinlan, 1993). They also conclude that this approach clearly outperforms other methods. The prefiltering method is generally applicable to any univariate feature selection method and has the potential for improved classification accuracy without high computational costs. For a review of further feature selection methods see section 1.4.3.

Univariate feature selection methods have a strong limitation. They only look at each gene separately and do not evaluate groups of genes together. Another approach is to evaluate several genes together. This is also called multivariate gene selection. Even though it is often claimed that multivariate gene selection strategies provide better classification accuracy this is not always the case (Lai *et al.*, 2006). Additionally, multivariate approaches suffer from their high complexity and brute force approaches are infeasible (Lai *et al.*, 2005). Therefore, several heuristic solution have been proposed. Bø and Jonassen (2002) could improve univariate gene selection by looking at each gene pair instead. Recursive feature elimination (RFE) is a wrapper method, where recursively in each round the features are scored together and the worst feature is eliminated through a backward sequential selection (Guyon *et al.*, 2002). As the classifier has to be retrained in each round in a complete cross validation manner, RFE is up to $p * c$ times slower (where p is the number of genes and c is proportional to the classifiers complexity) than plain univariate feature selection. It is possible to speed up recursive feature elimination (RFE) by eliminating several features at a time without sacrificing significance (Guyon *et al.*, 2002). The relevance of feature subsets in SVMs was further explored by Rakotomamonjy (2003) and an algorithm based on the SVM support vector weights was introduced. Xu and Setiono (2003) proposed a hybrid method method using a univariate maximum likelihood method (LIK) together with multivariate RFE. Recently, Duan *et al.* (2005) suggested a backward elimination procedure similar to support vector machine with recursive feature elimination (SVM-RFE) that trained multiple linear SVMs on subsamples of the original data and constructed a feature ranking score from the analysis of their weight vectors. Campbell *et al.* (2001) proposed a Bayesian approach for feature selection, an automatic relevance determination classification algorithm. They showed that it is simpler than support vector machine (SVM) RFE but performed only marginally worse.

Feature selection can serve to discover diagnostically relevant genes for a specific disease. But often many genes that are involved in a disease are already known and this prior knowledge can be incorporated. Kostka *et al.* (unpublished) suggested a LASSO approach (Tibshirani, 1996; Roth, 2004; Krishnapuram *et al.*, 2004) that puts initial weights on the genes and then optimizes an L1-classifier. It thus shrinks the weights of many genes to zero and removes them from the classifier. Hereby, prior knowledge of biologically interesting genes can be combined with the relevance of genes for classification. The genes that remain in the end are those genes that drive the classification.

I addressed the problem of feature selection and outlined why feature selection has to be done and how it can be done without loosing crucial information. For any fixed size the methods outlined here identify sets of genes that are stronger predictors than sets found by standard methods, which should be of significant value for diagnostic purposes.