

A Robust Method for Generating Discriminative Gene Clusters

Min Xu¹ and Louxin Zhang²

¹Institute of Molecular and Cell Biology, 30 Medical Drive, Singapore 117609. xumin@imcb.nus.edu.sg

²Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543.
matzlx@nus.edu.sg

Abstract

Motivation: Microarray technology is often used to identify the genes that are related to the tissue classes with the help of machine learning and statistical methods. Due to the small sample and high dimensionality nature of microarray data, it is not difficult to find small gene subsets which are highly discriminative. But these highly discriminative gene subsets may not be truly biologically relevant to the sample classes. Furthermore, the many existing identification process is very sensitive to the choice of training samples.

Results: We propose a novel approach for generating discriminative gene clusters. Our experiment on both simulation and real datasets show that our method can generate a series of robust gene clusters with good classification performance.

Availability: The program in MATLAB is available on request.

Contact: matzlx@nus.edu.sg

1 Introduction

Microarray has become an important tool for identifying discriminative genes for tissue classification because of its power of monitoring the expression levels of thousands of genes in one single experiment. Finding discriminative genes with microarray data is actually the feature selection problem in classification study. From the machine learning point of view, it is important since the microarray data have small samples but a large number of features. Good selected gene features can be used to build better classification and prediction tools. From the biological point of view, the selected feature genes can be further studied for finding the biological mechanisms that are responsible for class differentiation.

A lot of efforts have been put in finding gene selection methods (Speed, 2003; Bo and Jonassen, 2002; Ambrosie and McLachlan, 2002; Golub *et al.* 1999; Xing *et al.*, 2001; Guyon *et al.* 2002; Xiong *et al.*, 2001; Xiong, Fang, and Zhao, 2001; Ramaswamy *et al.*, 2001, Quackenbush, 2001). Due to the small-sample and high-dimension nature of the tissue classification problem, it is not difficult to find a small feature subset that can perfectly discriminate all the samples (Xu and Setiono, 2003). In fact, Cover (1965) showed that even for the non-informative, randomly generated dataset, the expected size of a

feature subset that can linearly discriminate all the n samples is just $(n+1)/2$. In microarray data analysis, there can be a large number of highly discriminative subsets containing only a couple of genes; and each individual gene in such a subset is not necessarily highly discriminative. For example, we observed by exhaustive search that there are as many as 10,173 perfect 3-gene subsets for classification with the weighted voting method proposed by Golub *et al.* (1999) and with their proposed training-test split; and these gene subsets cover 3,337 genes (93.4% of all the 3,571 genes in the datasets after preprocessing). This observation suggests that a method of finding a highly discriminative compact gene subset is not enough. The variability of the subsets found by such a method likely hinders the discovery of real interaction among the genes given that the method is usually sensitive to both the choice of samples and noise in microarray data.

The fundamental limit mentioned above motivates us to design more robust methods for identifying discriminative genes. In this paper, we find a series of discriminative gene clusters by running clustering and feature selection processes iteratively, where the centroids of the clusters are used to form predictors. The advantage of this approach is that genes grouped according to their expression would be more powerful in revealing regulation mechanisms. This is because genes perform their function interactively. This work also shows that the predictor constructed in this way is more robust to the choice of training samples.

Recently, Jornsten and Yu (2003) and Dettling and Buhlmann (2002) proposed supervised and unsupervised method combination for generating discriminative gene clusters. However, there are major differences between their methods and our method. We use a multivariate approach for cluster selection, while Dettling and Buhlmann (2002) employed a univariate approach. Univariate approach (see, for example, Liu and Motoda (1998)) assumes the independence of the contribution of clusters to classification. This hypothesis leads to computational advantage but may not reflect the complex biological interaction among gene clusters. We think a multivariate approach is more appropriate in the content of gene expression analysis since it accounts for the joint contribution of clusters to classification.

Our method is also different from Jornsten and Yu (2003) in the following two aspects although both works use multivariate approach. First, in their information-based approach, clustering and cluster selection are done simultaneously, resulting in a set of clusters optimizing the Minimum Description Length criterion. In comparison, our computation-oriented approach is a refinement process where clustering and cluster selection are performed alternatively in each iteration step. Secondly, the clusters generated with Jornsten and Yu's approach include active and inactive ones. Here, *active* clusters are those whose centroids are relevant to classification, and *inactive* ones are not. But, our approach iteratively eliminates the less active clusters and re-clusters remaining genes in the active clusters, which is essentially a backward approach (Ambroise and McLachlan, 2002).

Our approach outputs a series of cluster sets that have better and better discrimination power for training samples without losing prediction power on the test samples, as indicated in our experimental results. It outperformed the known methods just mentioned above for most of the tested datasets in our validation test. More importantly, our test shows that the centroids of the output clusters using different sets of training samples are stable and consistently significantly close to the global optimal gene clusters obtained by using all the samples. Another advantage of our method is that it provides researchers' flexibility to decide which cluster set should be chosen for their purpose. The main reason for proposing this approach is that a good discriminative cluster set for studying a real biological problem is highly problem dependent.

The paper divides into four parts. In Section 2, we describe our backward cluster generation method. In Section 3, we validate our approach using simulated datasets as well as real microarray datasets. Finally, we conclude the paper with a few remarks in Section 4.

2 Methods

2.1 Algorithm

Here, we propose a backward approach for generating discriminative gene clusters. The method iteratively groups genes into clusters and eliminates the ‘less’ discriminative ones. In the clustering stage, we use the K-means method to group the genes into a constant number of active clusters.

In the elimination stage, we use a backward feature selection method. This stage involves cluster validation and evaluation of the discriminative ability of active clusters. To validate clusters, we use the Silhouette width (Kaufman and Rousseeuw, 1990) to measure their quality. Assume the given genes are partitioned into p clusters C_1, C_2, \dots, C_p . Given a gene g , let \bar{w}_g be the average Euclidian distance between g and another gene in the same cluster, and let \bar{b}_{gJ} the average Euclidian distance between g and a gene in a different cluster C_J . Then the Silhouette width ω_g of g is defined as

$$\omega_g = \frac{\min_J (\bar{b}_{gJ}) - \bar{w}_g}{\max \left(\min_J (\bar{b}_{gJ}), \bar{w}_g \right)},$$

and the Silhouette width of a cluster is defined as the average Silhouette width of all its members. It is easy to see that the Silhouette width of a cluster is in the range between -1 and 1. A good cluster has high Silhouette width.

To evaluate the discriminative ability of an active cluster, we adopt the idea of SVM-RFE method proposed by Guyon *et al.* (2002). Support Vector Machine (SVM) is a binary-class prediction method originated from statistical learning theory (Vapnik, 2000). A linear SVM first finds a decision hyperplane $y = \mathbf{w}^T \mathbf{x} + b$ that maximizes the separation between samples of two classes; and then it does class prediction according to the relative location of a new sample with respect to the hyperplane in the feature space. Note that the weight vector \mathbf{w} found by the linear SVM indicates the relative importance of the genes for the classification. Here, we iteratively train a linear SVM and eliminate the gene cluster according to both the weight and the Silhouette width instead of eliminating only one gene in the SVM-RFE. This makes the elimination more systematic.

Our method is summarized into the following algorithm. In the algorithm, Δ denotes the set of inactive gene clusters; A_i denotes the set of active clusters at each iteration i ; S_i denotes the set of genes under consideration at the beginning of the iteration i ; κ denotes the number of clusters partitioned at each iteration step. For simplicity, we set κ to be n_r , the number of training samples.

Algorithm

Input: A gene expression dataset and a set S of the genes.

- 1) $\Delta \leftarrow \phi, i \leftarrow 0, S_0 \leftarrow S$;
- 2) Apply K-means to S_i to obtain κ active clusters and store them in A_i ;
- 3) Calculate the Silhouette width for each active cluster in A_i ;
- 4) Train a linear SVM using the centroids of active clusters in A_i ;
- 5) Find a least active cluster C_t in A_i according to its Silhouette width and weight in the SVM, and put it into Δ ;
- 6) $S_{i+1} \leftarrow S_i - \{C_t\}$;
- 7) If $|S_{i+1}| \leq \kappa$ then output $\{A_1, \dots, A_i\}$ and quit, else $i \leftarrow i + 1$ and go to 2).

It is hard to determine how many clusters the genes should be grouped into for microarray data analysis. The algorithm generates $\kappa = n_r$, the number of the training samples, active clusters in each iteration. This is because the expected size of a feature subset that can linearly discriminate all the samples is only $(n+1)/2$ (Cover, 1965). Moreover, if we let the feature number to be too small, the clustering will lose its resolution.

Recall that the K-means clustering method starts with an initial partition of the genes. In order to make it deterministic in Step 2, we first select κ genes as follows:

- a) Find a furthest gene pair and form an initial gene set G , and then
- b) iteratively find a gene with largest average Euclidean distance from the genes in G and add it into G until $|G| = \kappa$.

We then partition all the genes into κ clusters by merging each gene with its nearest gene in G .

The calculation of the Silhouette width of each cluster in A_i takes all the clusters in both sets A_i and Δ into account. At the i th iteration, the algorithm groups the genes in the set S_i into κ clusters, forming the cluster set A_i , and then moves the least active cluster into cluster set Δ in Step 5 as follows.

There are two important factors for consideration to determine which cluster should be removed from S_i and added into Δ . One factor is the cluster's Silhouette width. Another factor is the cluster's discrimination ability in terms of its weight determined by the linear SVM constructed in Step 4. Here, we would like to eliminate a least discriminative cluster whose centroid is sufficiently representative (measured by the Silhouette width). Since these two factors are not always consistent, we adopt a multiple objective optimization technique appearing in (Zhou and Gen, 1997) to find a compromise between these two factors:

Given a set of clusters with the Silhouette width $[\bar{s}_1, \dots, \bar{s}_\kappa]$ and the rescaled weight magnitude $[\tilde{w}_1, \dots, \tilde{w}_\kappa]$, where $\tilde{w}_i = 2|w_i| - 1$, w_i is the weight of the corresponding

centroid in the SVM constructed in Step 4. We define the objective function $f(i) = \tilde{w}_i(\bar{s}_{i'} - \bar{s}_{i''}) + \bar{s}_i(\tilde{w}_{i''} - \tilde{w}_{i'})$ for each i , where $i' = \arg \min_i(\tilde{w}_i)$ and $i'' = \arg \min_i(\bar{s}_i)$. The algorithm identifies the cluster with smallest $f(i)$ as the least active cluster.

Finally, we can extend the algorithm to the multiple-class case by adopting the popular one-against-all approach. In this approach, given a training test split, both training and test samples of a dataset of $k > 2$ classes are transferred into k binary classification problems, each corresponding to classify samples from one class against samples from all remaining classes. Then our algorithm executed on the k problems results in k series of active cluster sets $A_{j,i}$, $j = 1, \dots, k$. Then classifiers are constructed using $k \times \kappa$ clusters from the k active cluster sets $A_{1,i_1}, \dots, A_{k,i_k}$ by selecting i_1, \dots, i_k such that $|S_{i_1}|, \dots, |S_{i_k}|$ are roughly identical. Given the centroids of the above $k \times \kappa$ clusters, a multi-class linear SVM is trained using training samples and tested on test samples.

2.2 Evaluation

We validate our method in terms of its classification performance and clustering performance. The classification performance includes classification accuracy on training or testing samples. We use the SVM as the classifier to evaluate the generated gene clusters. Classification accuracy θ_{test} on the test samples is defined as the percentage of the correctly classified samples. However, we define classification accuracy θ_{train} on training samples as the average accuracy of the 10-fold cross validation on the training samples as suggested by Ambroise and McLachlan (2002) for less biased estimation of classification performance.

The clustering performance is measured by (a) the average Silhouette width $\bar{\omega}(A_i)$ of active clusters in A_i produced in iteration i , (b) the average Euclidean distance $\bar{\delta}(A_i)$ between the centroids of active clusters in A_i and (c) the ‘average’ Euclidean distance $\bar{d}(A_i, \Delta')$ between the centroid of an active cluster in each A_i and the centroid of a cluster in a reference cluster set Δ' (the construction can be found in Section 3). Assume $A_i = \{C_1, \dots, C_\kappa\}$ and $\Delta' = \{D_1, \dots, D_\kappa\}$. First, for l from 1 to κ , find recursively $C'_l \in A_i$ and $D'_l \in \Delta'$ such that $d(x(C'_l), x(D'_l)) = \min_{C'_l \in A_i - A'', D'_l \in \Delta' - \Delta''} d(x(C'_l), x(D'_l))$, where $x()$ denotes the centeroid of a cluster, $d(,)$ the Euclidean distance between two vectors, $A'' = \{C'_1, \dots, C'_{l-1}\}$ and $\Delta'' = \{D'_1, \dots, D'_{l-1}\}$. Then, the ‘average’ Euclidean distance $\bar{d}(A_i, \Delta')$ is defined as $\bar{d}(A_i, \Delta') = \frac{1}{\kappa} \sum_{1 \leq l \leq \kappa} d(x(C'_l), x(D'_l))$.

We measure the statistical significance of average distances in both case (b) and (c) at each iteration i against the pairwise distances of all genes in the input gene set S in terms of the p-value $\rho_S(i)$, and

against all the genes in the dataset in terms of the p-value $\rho_{all}(i)$. The p-value is calculated in terms of the empirical distribution of the pairwise distances in each case.

3 Validation results

We implemented the algorithm as MATLAB functions. It runs on a PC with the Windows operating system. The SVM program written by Cawley was downloaded from the website <http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox>. In this section, we present the detailed test results on both simulated and real datasets including Leukemia AML/ALL dataset (Golub *et al.* 1999) and compared the performance of our algorithm with those reported in the previous literature.

3.1 Simulated datasets

We generated 100 simulated binary classification datasets using a simple stochastic model. Each simulated dataset contains 100 samples evenly split into two classes. Both training and test samples contains 25 samples in each class.

Each dataset contains of 400 genes evenly divided into four gene clusters. Two of the four clusters are relevant to classification and these two discriminative clusters C_1 and C_2 contribute to classification independently. Their centroids $x(C_1)$ and $x(C_2)$ are generated according to the sample class labels. Each component of $x(C_1)$ in a position is generated according to normal distributions $N(1, 0.5)$ or $N(-1, 0.5)$ depending on whether the corresponding sample is in class 1 or class -1, while each component of $x(C_2)$ generated according to $N(-1, 0.5)$ if the sample is in class 1 and $N(1, 0.5)$ otherwise. Similarly, the centroids of the non-discriminative clusters C_3 and C_4 are generated according to the normal distribution $N(1, 1)$ and $N(-1, 1)$ regardless of the samples' class. For each $i=1, 2, 3, 4$, the expression values of a gene in the cluster C_i are generated according to the multivariate normal distribution $N(x(C_i), \frac{d_i}{4})$, where $d_i = \min_{j \neq i} d(x(C_i), x(C_j))$.

We run our algorithm with the input gene set S contains all the 400 genes for each of the 100 simulated datasets. The performance results are summarized in Figure 1. We observed that the classification performance of the generated clusters keeps on increasing as the iteration process goes. The average classification accuracy θ_{test} of these tests is from 0.756 up to 0.848 (Figure 1.a); and the classification accuracy θ_{train} on training samples goes up from 0.720 to 0.989 (Figure 1.b).

We also observed that there are more and more truly discriminative genes remaining in the active clusters as the algorithm proceeded. Since the genes in the discriminative clusters are known in each simulated dataset, we considered the ratio $p_{sim}(i) = \frac{|S_i \cap (C_1 \cup C_2)|}{|S_i|}$ of the truly discriminative genes over all the genes in S_i for each iteration i . For the active clusters output just before the algorithm terminates, $p_{sim}(i)$ is about 0.784 (Figure 1.c). Recall that, at each iteration i , the algorithm generates

$\kappa = 50$ active gene clusters since $n_r = 50$ for each simulated dataset. We found that at each iteration i , the centroids of two active clusters are very close to $x(C_1)$ and $x(C_2)$, the centroids of the discriminative clusters in the model. This is reflected by the indistinguishably small p-value $\rho_S(i)$ calculated based on $\bar{d}(A_i, \Delta')$, where $\Delta' = \{C_1, C_2\}$.

In the same time, the centroids of active clusters are more and more distinct from each other, increasingly close to the average pairwise distance of all 400 genes, which is reflected by the p-value $\rho_S(i)$ going from 0.228 up to 0.460 (Figure 1.d), which is calculated based on $\bar{\delta}(A_i)$. Meanwhile, the Silhouette width $\bar{\omega}(A_i)$ of active clusters in A_i increases from 0.826 to 0.981.

3.2 Leukemia dataset

Leukemia AML/ALL dataset (Golub et. al, 1999) contains the expression values of 6,817 human genes in 47 *acute lymphoblastic leukemia* (ALL) tissue samples and 25 *acute myeloid leukemia* (AML) tissue samples. After performing the threshold filtering and logarithmic transformation procedure, we obtained a reduced dataset with only 3,571 genes.

Here, we validate our algorithm by using three-fold cross validation. In each run, we randomly selected two third of the samples as the training samples and the rest as the testing samples. The samples of different classes are kept proportional in the training and test samples. The resulting dataset was further normalized by rescaling the variance of expression values of each gene to 1 in the training samples, and then applying the same rescaling factor to the expression values of that gene in the test samples.

We conducted the three-fold cross validation 100 times. In each run, the algorithm starts with the input gene set S consisting of the 357 genes (10% of all the 3,571 genes) that are highly correlated with the training samples' classification in terms of the correlation metric proposed in Golub *et al.* (1999).

Figure 2 summarizes the values of the different performance indicators. The average classification accuracy θ_{train} on the training samples ranges from 0.994 up to 1 (Figure 2.b); and the average classification accuracy θ_{test} on the test samples increase slightly from 0.966 to 0.972 (Figure 2.a). These results show that the centroids of the clusters generated in different iteration steps discriminate the training samples better and better without significant decrease of its generalization ability.

For the evaluation of our algorithm, we searched for perfect 3-gene subsets, which can be used to perfectly classify all 72 samples using the weighted voting classifier trained on all the samples. This search resulted in 9,722 perfect subsets. We selected 48 genes g_i ($1 \leq i \leq 48$) with highest occurrence frequency to form the cluster set $\Delta'_1 = \{\{g_i\} | 1 \leq i \leq 48\}$ for comparison with the clusters generated by our algorithm.

We also evaluate our algorithm using another cluster set Δ'_2 , the final set of active clusters generated by our algorithm with S' as the input gene subset and with all the 72 samples as the training samples,

where S' is the set of the 357 genes (10% of all the 3,571 genes) that are highly correlated with the AML/ALL classes in terms of the correlation metric proposed in Golub *et al.* (1999).

Probably because the selection bias of the correlation metric of Golub *et al.* (1999), the gene sets to our algorithm that are selected according to different training-test splits do not have too many genes in common. In all the 100 validation experiments, only 120 genes appearing in every input gene set S . This number is quite small compared with 1,071, the number of the genes appearing in some input gene sets. In contrast, the centroids of clusters in the set A_i generated in each of iterations of our algorithm in different runs are significantly similar to the selected discriminative genes in Δ'_1 and Δ'_2 at most iteration steps. This is indicated by the very small p-values $\rho_S(i)$ computed based on $\bar{d}(A_i, \Delta'_1)$ and $\bar{d}(A_i, \Delta'_2)$, which range from 4.11×10^{-2} to 6.12×10^{-3} (Figure 2.c) and from 5.62×10^{-3} to 2.38×10^{-3} (Figure 2.d) respectively.

3.3 The performance analysis on other real datasets

We also tested our algorithm on the following datasets:

- (1) ALL T/B Cell dataset. The 47 ALL samples in Leukemia ALL/AML dataset in Golub *et al.* (1999) are further divided into 39 T-cell samples and 9 B-cell samples.
- (2) Breast cancer dataset (West *et al.*, 2001). The dataset comprises 7,129 genes and 49 samples divided into two classes according to their estrogen receptor (ER) responses: 25 for ER positive and 24 for ER negative.
- (3) Carcinoma dataset (Notterman *et al.*, 2001). It contains the expression levels of about 6600 genes in 18 tumor and 18 normal tissues.
- (4) Colon dataset (Alon *et al.*, 1999). It consists of the expression levels of 6,500 human genes in 40 tumor and 22 normal tissues.
- (5) Diffuse Large B-Cell Lymphoma (DLBCL) dataset (Shipp *et al.*, 2002). It consists of expression values of 6,817 genes in 58 DLBCL and 19 Follicular Lymphoma.
- (6) The Melanoma dataset (Bittner *et al.*, 2000): The dataset consists of the expression ratios of 6,971 human genes in 12 Unclustered Cutaneous Melanomas and 19 Cutaneous Melanomas samples.
- (7) Prostate dataset (Singh *et al.*, 2001). The dataset consists of the expression levels of 52 prostate and 50 normal samples of 6,744 human genes.
- (8) The Small, round blue cell tumors (SRBCT) dataset (Khan *et al.*, 2001). It consists of 88 samples divided into five classes: neuroblastoma (NB) (18 samples), rhabdomyosarcoma (RMS) (25 samples), Burkitt lymphomas (BL) (11 samples), the Ewing family of tumors (EWS) (29 samples) and others (5 samples). The dataset has 2,308 genes. Since we consider the binary classification problem, we derived four binary classification datasets from this dataset using the one-against-all rule: SRBCT-NB, SRBCT-RMS, SRBCT-BL SRBCT-EWS.

We preprocessed each dataset by filtering with threshold and logarithm transformation if necessary. For each dataset, we run our algorithm 100 times by choosing random training-test splits in the same way as the Leukemia dataset described in the last subsection. The performance of our method is summarized in Table 1. The classification accuracy θ_{test} on the test samples shows that among 9 of 12 datasets, the prediction performance of active clusters in A_i increases slightly from the start to the end of each execution, which are highlighted in the table. The value of θ_{test} for the remaining three datasets (Breast, Colon and Carcinoma) decrease slightly. The above observations indicate that for all datasets we tested, there is no significant decrease in the generalization ability of the active clusters in A_i obtained in each iteration step. The classification performance θ_{train} on the training samples increases in all the 12 datasets, which indicates that the separation of the training samples increases for all datasets.

The all the 100 input gene set S 's vary a lot in different runs for each dataset. There are only 1.1% to 5.1% of all the genes appearing in all the 100 input gene set S 's, while at least 23.8% to 51.7% genes appear in some input gene sets. By contrast, the centroids of clusters in A_i generated by our algorithm at each iteration step i are stably close to the optimal centroids of clusters in Δ'_2 as reflected by the p-values $\rho_S(i)$ ranging from 2.99×10^{-4} to 8.75×10^{-2} at the first iteration step and those ranging from 6.42×10^{-5} to 3.20×10^{-2} in the last iteration step. The consistent closeness of the clusters generated in different repeats can also be reflected in the standard deviation of $\rho_S(i)$, which are limited from 0.32 to 0.96 times of the absolute values of $\rho_S(i)$ in the first iteration step and 0.24 to 1.37 times at the last iteration step.

During the generation process, the p-values $\rho_{all}(i)$ of average pairwise distance $\bar{\delta}(A_i)$ among centroids of clusters in A_i keeps increasing for all 12 datasets (ranging from 0.088 to 0.750 at the first iteration step and from 0.252 to 0.755 in the last step), and the average Silhouette width of active clusters $\bar{\omega}(A_i)$ keeps increasing for all the 12 datasets (ranging from 0.230 to 0.698 at the first iteration step and from 0.964 to 0.989 in the last iteration step). This indicates the clusters in A_i are more and more distinct in general.

In summary, our test shows that on these real datasets, our algorithm is able to generate clusters that separate the training samples better and better, without significantly loss of generalization ability and closeness to known optimal clusters. This behavior is consistent with what we have observed on simulated datasets.

3.4 Comparing the classification performance to other studies

In this section, we compare the cross validation performance of our method with previous works in Dettling and Buhlmann (2002), Jornsten and Yu (2003), Shipp *et al.* (2002), and Jaeger *et al.*, (2003). For the purpose of comparison, we converted the classification performance from the classification accuracy θ_{test} into the error rate. The comparison is summarized in Table 2.

Dettling and Buhlmann (2002) reported the three-fold cross validation classification error rate of their algorithm for different datasets. They employed nearest neighbors and aggregated trees as the classifiers. For the leukemia AML / ALL dataset, our algorithm seems achieved a slightly lower error rate than theirs. In the Colon and Prostate datasets, the error rate of our algorithm lies between theirs. For the Breast dataset, the error rate is significantly higher than Dettling and Buhlmann's. However, we obtained the performance using all the original 49 samples. The range of cross validation is equivalent to 7.89 to 6.90 errors. According to (West *et al.* 2001), at least 7 out of the 49 samples are inherently erroneous. In comparison, Dettling and Buhlmann (2002) used the 38 good samples selected by West *et al.* (2001), and the range of average errors is 1.14 to 0.10. The 38 samples used by Dettling and Buhlmann (2002) consists none of the above 7 erroneous samples. Thus, we believe the performances of ours and Dettling and Buhlmann's are still comparable for the Breast dataset.

For the DLBCL dataset, the leave-one-out performance of Shipp *et al.* (2002) lies in the error-rate interval of ours method. For Carcinoma dataset, Jaeger *et al.*, (2003) achieved perfect leave-one-out performance, and our best performance can match theirs. For the Colon dataset, both ours and Dettling and Buhlmann's error rate are higher than Jornsten and Yu's.

We also test the performance of the multiple-class version of our method against other methods. For the Leukemia three-class dataset, Jornsten and Yu's method has error rate lying in the error-rate interval of our method. However, for the SRBCT multi class dataset, our algorithm seems achieved a slightly higher range of error rate than Dettling and Buhlmann's.

4 Conclusion

Due to the small-sample-high-dimension nature of the microarray dataset, it is not difficult to find highly discriminative gene subsets of small size. However, if a gene selection processes is unstable with the choice of training samples, the biological significance of the resulting gene subsets are often not guaranteed. In this paper, instead of finding individual discriminative genes or gene subsets, we purpose a novel backward approach for generating a series of highly discriminative gene cluster sets. Genes grouped in clusters can provide more support to the gene interactions that are relevant to the sample classes.

Regarding to the classification performance, the gene clusters produced by our approach can generally achieve good cross validation performance than the existing methods for the most of datasets we tested. Our test experiments show that regardless of choice of training samples, the centroids of the clusters generated are stable and significantly close to the known optimal gene clusters found using all the samples. All these indicate that our approach is promising. However, the current version of our algorithm is time consuming. In the future, the computational efficiency will be investigated. Furthermore, more suitable clustering and backward feature selection method needs to be exploited so that the gene clustering and cluster selection process can be integrated better.

Our approach for generating discriminative gene clusters is generic. There are many modifications that can customize the algorithm to a particular problem. For example, we can replace the k-means method with the Diana method (Kaufman and Rousseeuw, 1990) in our algorithm. Diana is a divisive clustering method. The Diana-based algorithm will be different from our current algorithm in the following two points: At the initialization stage of the algorithm, S is divided into κ clusters by iteratively dividing the worst cluster in terms of the Silhouette width using Diana. In the cluster

generation process, instead of re-cluster all the genes in S_i , we use Diana to split the worst cluster in A_i to form new clusters. Since in each iteration only genes in one cluster in A_i are re-clustered, the cluster generation process integrating Diana runs faster than the one presented in Section 2 that re-cluster all genes in A_i .

The basic principle of SVM is to find a vector such that the projection of the binary-class samples on the vector can maximize the extremal margin criterion. The weight of the features can be derived directly from the vector. Use the same idea of finding the vector but replacing the discrimination criterion, it is possible to integrate other backward feature selection methods such as Fisher's Linear Discriminate (Webb, 2002) into our algorithm by replacing SVM-RFE method. Different cluster eliminate strategies can also be used. For example, factors other than the discrimination ability and cluster quality can also be used in the elimination process. More than one cluster can also be eliminated in each step to accelerate the elimination process.

5 Acknowledgement

This work was partially supported by the Singapore BioMedical Research Council research grant BMRC01/1/21/19/140. M. Xu would also like to thank Rudy Setiono and Jinrong Peng for their helpful conversations.

REFERENCE

- Alon U, *et al.*, 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci USA*. **96**:6745-6750.
- Ambroise C and McLachlan GJ, 2002. Selection bias in gene extraction on the basis of microarray gene-expression data. *Proc Natl Acad Sci USA*. **99**:6562-6.
- Bittner M, *et al.*, 2000. Molecular classification of cutaneous malignant melanoma by gene expression profiling. *Nature*. **406**:536-40.
- Bo T and Jonassen I, 2002. New feature subset selection procedures for classification of expression profiles. *Genome Biol*. **3**:RESEARCH0017.
- Cover T, 1965. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans. Elec. Comp*. **14**:326-334.
- Dettling M and Buhlmann P, 2002. Supervised clustering of genes. *Genome Biol*. **3**:RESEARCH0069.
- Golub TR, *et al.*, 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*. **286**:531-537.
- Guyon I, Weston J, Barnhill S and Vapnik V, 2002. Gene selection for cancer classification using support vector machines. *Mach. Learn*. **46**:389-422.
- Jaeger J, Sengupta R, Ruzzo WL, 2003. Improved gene selection for classification of microarrays. *Pac Symp Biocomput*. 53-64.

- Jornsten R and Yu B, 2003. Simultaneous gene clustering and subset selection for sample classification via MDL. *Bioinformatics*, **19**:1100-1109.
- Kaufman L and Rousseeuw PJ, 1990. *Fitting Groups in Data. An Introduction to Cluster Analysis*. Wiley, New York.
- Quackenbush, J. 2001. Computational analysis of cDNA microarray data. *J. Nature Reviews* 2(6):418-428
- Khan J, *et al.*, 2001. Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature. Med.* **7**:673-679.
- Liu H, and Motoda H, 1998. *Feature Selection for Knowledge Discovery and Datamining*. Kluwer Academic Publishers.
- Notterman DA *et al.*, 2001. Transcriptional gene expression profiles of colorectal adenoma, adenocarcinoma, and normal tissue examined by oligonucleotide arrays. *Cancer Res.* **61**:3124-30.
- Ramaswamy S. *et al.*, 2001. Multiclass cancer diagnosis using tumor gene expression signatures. *Proc. Natl. Acad. Sci.*, **98**:15149-15154.
- Shipp MA, *et al.*, 2002. Diffuse large B-cell lymphoma outcome prediction by gene expression profiling. *Nat Med.* **8**:68-74.
- Singh D, *et al.*, 2002. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell.* **1**:203-209.
- Speed T. 2003. *Statistical Analysis of Gene Expression Microarray Data*. Chapman & Hall, USA.
- Vapnik VN, 2000. *The Nature of Statistical Learning Theory*, 2nd edition, Springer, New York.
- Webb A, 2002. *Statistical Pattern Recognition*, 2nd edition, John Wiley and Sons.
- West M, *et al.*, 2001. Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc Natl Acad Sci USA.* **98**:11462-11467.
- Xing EP, Jordan MI, and Karp RM, 2001. Feature selection for high-dimensional genomic microarray data. *Proceedings of the Eighteenth International Conference on Machine Learning.* 601-608.
- Xiong M *et al.*, 2001. Feature (gene) selection in gene expression-based tumor classification. *Molecular Genetics and Metabolism*, **73**: 239-247.
- Xiong M, Fang X and Zhao J. Biomarker identification by feature wrappers. *Genome Research*, **11**: 1878-1887.
- Xu M and Setiono R, 2003. Gene selection for cancer classification using a hybrid of univariate and multivariate feature selection methods. *Applied Genomics and Proteomics.* **2**:79-91.
- Zhou G and Gen M, 1997. Evolutionary Computation on Multicriteria Production Process Planning Problem. *Proceedings of the 1997 IEEE International Conference on Evolutionary Computation.* 419-424.

TABLES

Datasets	θ_{test}		θ_{train}		$\rho_S(i)$ based on $\bar{d}(A_i, \Delta'_2)$		$\rho_{all}(i)$ based on $\bar{\delta}(A_i)$		$\bar{\omega}(A_i)$	
	Leukemia ALL T/B cell	0.970	0.977	1.000	1.000	1.72E-02	8.28E-03	0.088	0.331	0.406
Breast	0.843	0.842	0.989	1.000	1.33E-02	8.36E-03	0.142	0.421	0.351	0.974
Carcinoma	0.983	0.981	1.000	1.000	2.96E-02	3.20E-02	0.194	0.252	0.382	0.966
Colon	0.814	0.806	0.836	0.941	2.43E-02	2.06E-02	0.750	0.755	0.673	0.978
DLBCL	0.896	0.929	0.970	1.000	8.75E-02	1.99E-02	0.441	0.514	0.716	0.982
Melanoma	0.913	0.921	0.993	1.000	1.71E-02	2.25E-02	0.129	0.463	0.272	0.957
Prostate	0.889	0.916	0.932	0.992	4.79E-02	2.27E-02	0.495	0.541	0.680	0.987
SRBCT-BL	1.000	1.000	1.000	1.000	3.63E-04	7.52E-05	0.314	0.322	0.682	0.984
SRBCT-EWS	0.956	0.986	0.986	1.000	5.06E-04	9.17E-05	0.297	0.408	0.634	0.984
SRBCT-NB	0.989	0.996	0.997	1.000	2.99E-04	6.42E-05	0.321	0.436	0.665	0.986
SRBCT-RMS	0.974	0.980	0.988	1.000	4.82E-04	8.18E-05	0.304	0.347	0.630	0.989
Lukemia AML / ALL	0.966	0.972	0.995	1.000	5.62E-03	2.38E-03	0.212	0.398	0.627	0.980

Table 1: Summary of the performance of the algorithm for different datasets. In the table, there are two columns for each performance measure, indicating to the average values of the corresponding measures at the first and last iteration step of our algorithm. Because the exhaustive search of the highest frequent globally optimal genes for constructing Δ'_1 is time-consuming, we only compare the active clusters with Δ'_2 , the set of active clusters generated by our algorithm at the last iteration step with all the samples as the training samples in the same way as described for the leukemia ALL/AML dataset in Section 3.2.

Datasets	Our algorithm	Dettling and Buhlmann (2002)	Jornsten and Yu (2003)	Shipp et al. (2002)
Lukemia AML / ALL	3.43 - 2.57	6.58 - 2.71		
Leukemia three classes	13.8 - 9.3		12.6	
Breast	16.14 - 14.11	3.00 - 0.75		
Carcinoma	5.6 - 0.0			
Colon	19.41 - 18.23	23.35 - 15.95	13.6	
DLBCL	8.7 - 7.4			7.8
Prostate	11.09 - 8.36	16.47 - 6.91		
SRBCT multi class	5.92 - 4.27	5.76 - 0.43		

Table 2: Comparison of our algorithm (of both binary and multi-class versions) with others by the cross validation error rates. The DLBCL and Carcinoma datasets are validated using leave-one-out validation; and the remaining datasets are validated using three-fold cross validation.

FIGURES

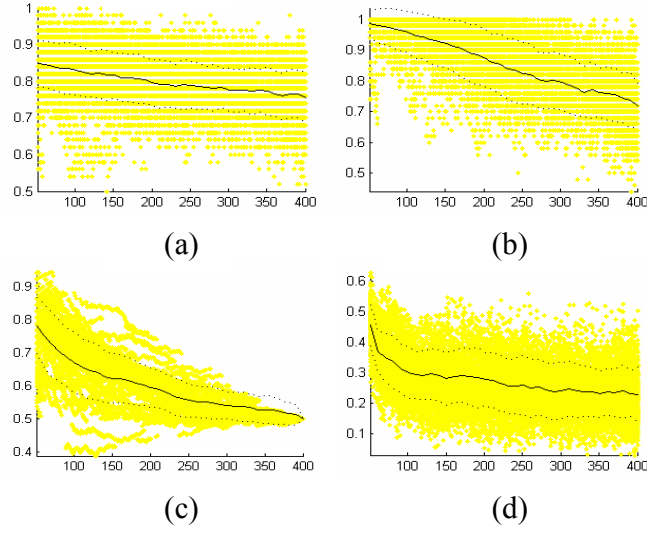


Figure 1: The performance analysis on simulated datasets. The dots indicate the performance values in individual tests. The real lines indicate the average values; and dotted lines indicate one standard deviation from the averages. The X-axis represents the number of genes in S_i . Note that, when the generating process goes, the number of genes in S_i decreases. (a) The classification accuracy θ_{test} on the test samples. (b) The classification accuracy θ_{train} on the training samples. (c) The percentage $p_{sim}(i)$ of truly discriminative genes in S_i ; (d) The p-value $\rho_S(i)$ based on $\bar{\delta}(A_i)$.

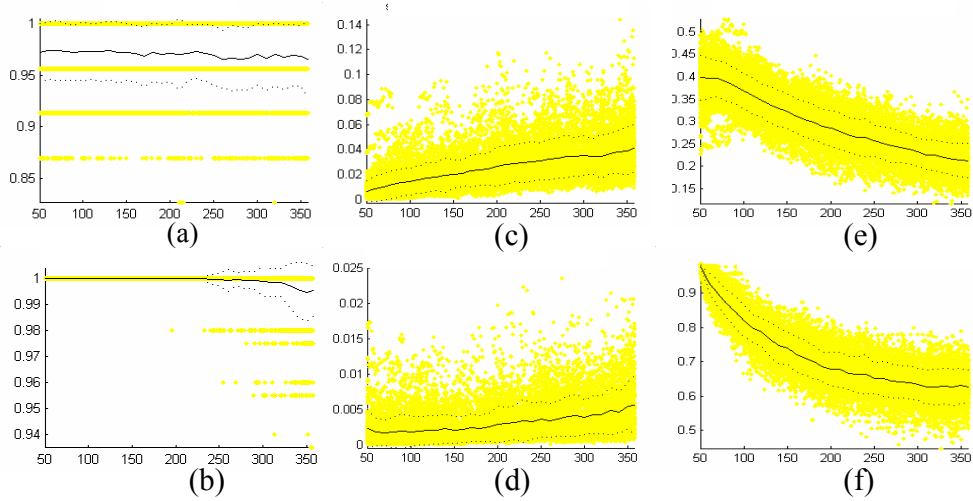


Figure 2: The analysis of the three-fold cross validation performance of the algorithm on the Leukemia dataset. The dots indicate the performance values in individual tests. The real lines indicate the average values; and the dotted lines indicate one standard deviation from the averages. The X-axis represents the number of genes in S_i . (a) The classification accuracy θ_{test} on the test samples. (b) The classification accuracy θ_{train} on the training samples. (c) The p-values $\rho_S(i)$ based on $\bar{d}(A_i, \Delta_1)$. (d) The p-values $\rho_S(i)$ based on $\bar{d}(A_i, \Delta_2)$. (e) The p-value $\rho_{all}(i)$ based on $\bar{\delta}(A_i)$. (f) The average Silhouette width $\bar{\omega}(A_i)$ of active clusters in A_i .