Statistical analysis on microarray data: selection of gene prognosis signatures

Kim-Anh Lê Cao[1] and Geoffrey J. McLachlan[2]

[1]ARC Centre of Excellence in Bioinformatics, Institute for Molecular Bioscience, University of Queensland, 4072 St Lucia, QLD, Australia

[2]Department of Mathematics and Institute for Molecular Bioscience, University of Queensland, 4072 St Lucia, QLD, Australia

Abstract

Microarrays are being increasingly used in cancer research for a better understanding of the molecular variations among tumours or other biological conditions. They allow for the measurement of tens of thousands of transcripts simultaneously in one single experiment. The problem of analysing these data sets becomes non-standard and represents a challenge for both statisticians and biologists, as the dimension of the feature space (the number of genes or transcripts) is much greater than the number of tissues. Therefore, the selection of marker genes among thousands to diagnose a cancer type is of crucial importance and can help clinicians to develop gene-expression based diagnostic tests to guide therapy in cancer patients.

In this chapter we focus on the classification and the prediction of a sample given some carefully chosen gene expression profiles. We review some state-of-the-art

machine learning approaches to perform gene selection: recursive feature elimination, nearest-shrunken centroids, and random forests. We discuss the difficulties that can be encountered when dealing with microarray data, such as selection bias, multiclass, and unbalanced problems. The three approaches are then applied and compared on a typical cancer gene expression study.

1. Introduction

Microarray data allow the measurement of expression levels of tens of thousands of genes simultaneously on a single experiment. The biological aim of these experiments is to better understand interactions and regulations between genes, which are spotted on the array in some given conditions. For example, in the context of cancer data, there are several types of statistical problems that can be considered:

-to identify new tumour classes using gene expression signatures (*e.g.* cluster analysis, unsupervised learning).

-to classify samples into known cancer classes (*e.g.* discriminant analysis, supervised learning).

-to identify marker genes that characterize one or several cancer types (*i.e.* feature selection).

Considering this last point, feature selection or *gene selection* may allow for the development of diagnostic tests to detect diseases and, in the particular case of cancer data, the selected genes can give more insight into the tumours characteristics. These genes are called *prognosis signatures* or *gene signatures*.

From a statistical point of view, the number of genes is usually often greater than the number of arrays, which renders the problem non-standard to solve. The selection of a relevant subset of genes enables one to improve the prediction performance of classification methods and to circumvent the curse of dimensionality. It also enables one to reduce computation time and allows for an understanding of the underlying biological process that generated these data.

Statistical analysis of microarray data involves several important steps, such as normalization and pre-processing; see McLachlan et al. (2004), Chapter 2 and Li et al. (2003). In this chapter, the focus is solely on the analysis of microarray data and the selection of genes using classification methods.

1.1. Notation

In this chapter, we will adopt the following notation. A microarray data set consists of the quantitative measurements of p genes (called *predictor variables*) on n tissues (called *samples*). These data are summarized in a $p * n$ matrix $\boldsymbol{X} = x_{ij}$, where x_{ij} is the expression of gene i in the j^{th} microarray ($i = 1, \dots, p; j = 1, \dots, n$).

In the context of classification, we can represent the a $p * N$ matrix $\boldsymbol{X}$ of gene expressions as

$$\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n),$$

where the feature vector $\boldsymbol{x}_j$ (the expression signature) contains the expression levels of the p genes in the j^{th} tissue sample ($j = 1, \dots, n$).

2. Supervised classification

In the context of supervised classification, each tissue belongs to a known biological class k, $k = 1, \ldots, g$. In the following, we let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ denote the feature vectors and $z_1, \ldots, z_n$ the corresponding vectors of zero-one indicator variables defining the known class of each sample. The collection of the data

$$\boldsymbol{t} = (\boldsymbol{x}_1^T, \boldsymbol{z}_1^T, \ldots, \boldsymbol{x}_n^T, \boldsymbol{z}_n^T)^T$$

will be referred to as the *training* data.

In supervised classification, the aim is to construct a rule $r(\boldsymbol{x}; \boldsymbol{t})$ based on the training data $\boldsymbol{t}$ with feature vectors for which the true class is known. Based on this rule, the final aim of supervised classification approaches is to predict the class label of a new tissue sample.

Such problems are ubiquitous and, as a consequence, have been tackled in several different research areas. As a result, a tremendous variety of algorithms and models have been developed for the construction of such rules. In the sequel, we will describe some classification methods, such as Support Vector Machines, Shrunken centroids, and classification trees. We will show that these classifiers can be included in some machine learning approaches to perform variable selection.

2.1 Linear classifier

As an introduction to prediction rules, we first consider the basic linear function in the case where $g = 2$ (binary problem). For any feature vector, here denoted $\boldsymbol{x}$, its label is assigned to class 1 or class 2 if

$$r(\boldsymbol{x}; \boldsymbol{t}) = 1, \text{ if } c(\boldsymbol{x}; \boldsymbol{t}) > 0,$$

$$= 2, \text{ if } c(\boldsymbol{x}; \boldsymbol{t}) < 0,$$

where $c(\boldsymbol{x};\boldsymbol{t}) = \beta_0 + \boldsymbol{\beta}^T\boldsymbol{x} = \beta_0 + \beta_1 (\boldsymbol{x})_1 + \cdots + \beta_i (\boldsymbol{x})_i + \ldots + \beta_p (\boldsymbol{x})_p$,

and $(\boldsymbol{x})_i$ denotes the i^{th} element of the feature vector $\boldsymbol{x}$ $(i = 1, \ldots, p)$.

The function $c(\boldsymbol{x};\boldsymbol{t})$ is a linear combination of the features (or genes) with different weights β_i $(i = 1, \ldots, p)$. Once the rule $r(\boldsymbol{x};\boldsymbol{t})$ is constructed on the training data $\boldsymbol{t}$, we can use it to predict the class label of a new feature vector.

2.2 Support vector machines

The Support Vector Machine (SVM) is a powerful machine learning tool that has often been applied to microarray data (Vapnik, 2000). We briefly describe the formulation of a soft margin SVM, that is, when classes are linearly non-separable. In this section, we assign a label $y_j \in \{1, \ldots, g\}$ for $j = 1, \ldots, n$ to each tissue sample to indicate the known class of each sample.

In the case where $g = 2$, the SVM learning algorithm with a linear kernel aims to find the separating hyperplane

$$\boldsymbol{\beta}^T\boldsymbol{x} + \boldsymbol{\beta}_0 = 0,$$

that is maximally equidistant from the training data of the two classes. In this case of $g = 2$, it is conveninent if we let the class label y_j be *1* or *-1* to denote membership of class *1* or class *2*. When the classes are linearly separable, the hyperplane is located so that there is maximal distance between the hyperplane and the nearest point in any of the classes. This distance is called the margin and is equal to $2/\boldsymbol{\beta}^T\boldsymbol{\beta}$. The aim is to maximize this margin, that is, to minimize $\boldsymbol{\beta}^T\boldsymbol{\beta}$.

When the data are not separable, the margin is maximized so that some classification errors are allowed. In that case, the so-called *slack* variables ξ_j are used $(j = 1, \ldots, n)$. The quadratic optimization problem to solve is:

$$\min_{\boldsymbol{\beta},\beta_0,\boldsymbol{\xi} \in \mathbb{R}^{p+1+n}} \ \boldsymbol{\beta}^T\boldsymbol{\beta}, \qquad (2.1)$$

subject to

$$y_j\left(\boldsymbol{\beta}^T \boldsymbol{x_j} + \boldsymbol{\beta}_0\right) \leq 1 - \xi_j, \tag{2.2}$$

where $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_N)^T$ is the vector of so-called slack variables.

The cases $\boldsymbol{\beta}^T \boldsymbol{x} + \boldsymbol{\beta}_0 = \pm(1 - \xi_j)$ are the *support vectors* which define the solution. The Lagrangian dual formulation is finally

$$\min \frac{1}{2} \sum_{j,k}^{n} \alpha_j \alpha_k y_j y_k \, \boldsymbol{x_j}.\boldsymbol{x_k} - \sum_{j}^{n} \alpha_j,$$

$$\text{subject to} \qquad 0 \leq \alpha_j \leq C \quad \text{and} \quad \sum_j \alpha_j \, y_j = 0, \tag{2.3}$$

where C corresponds to a penalty for misclassified cases and the α_j $(j = 1, \ldots, n)$ are the Lagrange multipliers corresponding to the constraints (2.2). We call the *support vectors* the cases where $\boldsymbol{\alpha_j \neq 0}$. The use of this 'soft' margin enables the misclassification of outliers during training and avoids overfitting.

Let *S* the set of indices of the Support Vectors and $\boldsymbol{x}_s$ any Support Vector case, then given the solution to the problem (2.1), the corresponding discriminant rule is

$$r(\boldsymbol{x};\, \boldsymbol{t}) = \text{sign}(y_s \sum_{j \in S} \alpha_m \, y_m \, \boldsymbol{x}_m \, . \boldsymbol{x}_s + \boldsymbol{\beta}_0).$$

By using the '*kernel trick*' and the scalar product in the Lagrangian formulation (2.3), this standard SVM can be extended to nonlinear decision functions to map the data into a higher, possibly infinite, dimensional space. The user will then need to specify the kernel function to use. More details about the SVM methodology can be found in the tutorial of Burges (1998) and Cristianini & Shawe-Taylor (1999).

2.3 Nearest centroid

The nearest centroid rule assigns the feature vector $\boldsymbol{x}$ to the class whose mean centroid is closest in Euclidian distance. For the classes $k = 1, \ldots, g$, let C_k be indices of the n_k samples in class k. The i[th] component of the centroid for class k is $\bar{\mathbf{x}}_{ik} = \sum_{j \in C_k}^{n} \mathrm{x}_{ij}/n_k$, which is the mean expression value in class k for the gene i. The i[th] component of the overall centroid is $\bar{\mathbf{x}}_i = \sum_{j=1}^{n} \mathrm{x}_{ij}/n$.

Nearest centroid classification takes the gene expression profile of a new sample, and compares it to each of these class centroids. The class whose centroid that it is closest to, in squared distance, is the predicted class for that new sample.

Note that in contrary to SVM, nearest centroid classifiers can be naturally generalized to multiclass problems ($g > 2$).

In the case of high dimensional microarray data, Tibshirani et al. (2002) proposed the '*nearest-shrunken centroid*' rule that 'shrinks' each of the class centroids toward the overall centroid for all classes by moving the centroid towards zero by a *threshold*. It also takes into account the different gene variances. This approach has two advantages: 1) it can make the classifier more accurate by reducing the effect of noisy genes, 2) it performs automatic gene selection (see Section 3.3).

2.4 Classification and regression trees

Tree-based methods such as Classification And Regression trees (CART, Breiman et al., 1984) are conceptually simple and easy to interpret. In our case we will focus on binary classification trees only, that is, when a binary split is performed at each node of the tree.

The construction of CART requires one to choose:

i. the best split for each node, *i.e.* the best predictor (gene) to split the node and the best threshold among this predictor.

ii. a rule to declare a node '*terminal'*, *i.e*. when to stop splitting.

iii. a rule to affect a class to each terminal node.

The best split criterion (*i*) relies on a *heterogeneity* function, so that the cases or samples that belong to the same class land in the same node. *Gini* index or *entropy* index is an example of such heterogeneity functions; see Breiman et al. (1984).

When applying classification trees to noisy data like microarray data, a major issue concerns the decision when to stop splitting (*ii*). For example, if 9 splits are performed (*i.e.* with AND/OR rules for each split) with only 10 observations, then it is easy to perfectly predict every single case. However, if new cases run this tree, it is highly likely that these cases will land in a terminal node with a wrong predicted class. This issue is called '*overfitting*', that is, the model applied on the data does not generalize well to new data because of random noise or variation. The way to address this issue with CART is to stop generating new split nodes when subsequent splits only result in very little overall improvement of the prediction. This is called '*pruning*'. The tree is first fully grown and the bottom nodes are then recombined or pruned upward to give the final tree, where the degree of pruning is determined by cross-validation (see Section 2.5.2) using a cost complexity function.

The class of the terminal node (*iii*) is determined as the majority class of the cases that land in the same terminal node. Details of the CART methodology can be found in Breiman et al. (1984).

Trees are different from other previously considered classification methods as they are learning and selecting features simultaneously (embedded approach, see Section 33.1. However, one of the major problems with trees is their high variance. Indeed, a small

change in the data can result in a very different series of splits and hence a different prediction for each terminal node. A solution to reduce the variance is to consider bagging (Breiman, 1996) as was done in Random Forests, see Section 03.4.

2.5 Error rate estimation

Given a discriminant rule $r(\boldsymbol{x}; \boldsymbol{t})$ constructed on some training data $\boldsymbol{t}$, we now describe some techniques to estimate the error rates associated with this rule.

2.5.1. Apparent error rate

The apparent error rate, also called *resubstitution* error rate, is simply the proportion of the samples in $\boldsymbol{t}$ that are misallocated by the rule $r(\boldsymbol{x}; \boldsymbol{t})$. Therefore, this rate is obtained by applying the rule to the same data from which it has been learnt. As mentioned by several authors (McLachlan, 1992, Chapter 10), it provides an overly optimistic assessment of the true error rate and would need some bias correction. To avoid this bias, the rule should be tested on an independent test set or a hold out test set from which the rule has not been formed. We next present some estimation methods to avoid this bias.

2.5.2 Cross-validation

To almost eliminate the bias in the apparent error rate, one solution is to perform leave-one-out cross-validation (LOO-CV) or *V*-fold cross-validation (CV). Cross-validation consists in partitioning the data set into *V* subsets of roughly the same size, such that the learning of the rule $r(\boldsymbol{x}; \boldsymbol{t})$ is performed on the whole subsets minus the v^{th} subset, and tested on the v^{th} subset, $v = 1, \ldots, V$. This is performed *V* times, such that each sample is tested once and the *V* subsequent error rates are then averaged.

In the case of LOO-CV, $V = n$ and therefore, the rule is tested on only one sample point for each fold. LOO-CV may seem to require considerable amount of computing, as the rule has to be formed *n* times to estimate the error rate. Furthermore, this estimate may yield a too high a variance. A bootstrap approach was then proposed in an attempt to avoid these limitations.

2.5.3 Bootstrap approach

Efron (1979, 1983) showed that suitably defined bootstrap procedures can reduce the variability of the leave-one-out error in addition to providing a direct assessment of variability for estimated parameters in the discriminant rule. Furthermore, if the number of bootstrap replications is less than *n*, it will result in some saving in computation time relative to LOO-CV computation.

Let *E* denote the error computed on the cases that were not drawn in the bootstrap sample, Efron (1983) proposed the $B^{.632}$ estimator to correct some upward bias in the error *E* with the downwardly biased apparent error *A*:

$$B^{.632} = 0.368\,A + 0.632\,E.$$

Previously, McLachlan (1977) had derived an estimator similar to $B^{.632}$ in the special case of two classes with normal homocedastic distributions.

When the number of variables is much larger than the number of samples, the prediction rule $r(\boldsymbol{x};\boldsymbol{t})$ usually overfits, that is, A often equals 0. Efron & Tibshirani (1997) then proposed the $B^{.632+}$ estimate,

$$B^{.632+} = (1-w)A + wE,$$

where

$w = \frac{0.632}{1-0.368r}$, $r = \frac{E-A}{\min(E,\gamma)-A}$ and $\gamma = \sum_{k=1}^{g} p_k\,(1 - q_k)$.

r is an overfitting rate and γ is the no-information error rate, p_k is the proportion of samples from class C_k, q_k is the proportion of samples assigned to class C_k with the prediction rule $r(\boldsymbol{x};\ \boldsymbol{t})$ $(k = 1, \ldots, g)$.

3 Variable selection

The so-called "large *p* small *n* problem" poses analytic and computational challenges. It motivates the use of variable selection approaches, not only to infer reliable statistical results and to avoid the curse of dimensionality, but also to select relevant and potential gene signature related to the tissue characteristics.

In the machine learning literature, there exists three types of classification and feature selection methods (Kohavi & John, 1997; Guyon & Elisseeff, 2003): the *filter* methods, the *wrapper* methods and the *embedded* methods. We first describe the particularities of these approaches, before detailing some useful wrapper and embedded methods to perform gene selection: Recursive Feature Elimination (Guyon et al., 2002), Nearest Shrunken Centroids (Tibshirani et al., 2002) and Random Forests (Breiman, 2001), that will be applied on one well-known microarray data set from Golub et al. (1999).

3.1 Filter, wrapper and embedded approaches

The *filter methods* are often considered as a pre-processing step to select differentially expressed genes. The principle of this method is to independently test each gene and to order the genes according to a criterion, for example a *p*-value. The t- and F- tests are often used for microarray data. In one of the first comparative studies of

classification methods in the context of microarray data, Dudoit & Fridlyand (2002) proposed to pre-process the genes based on the ratio of their between-groups to within-groups sum of squares:

$$\frac{BBS(i)}{WSS(i)} = \frac{\sum_{j,k} I_{y_j=k}(\bar{x}_{ik} - \bar{x}_i)^2}{\sum_{j,k} I_{y_j=k}\,(x_{ij} - \bar{x}_{ik})^2}$$

where $\bar{x}_i$ is the average expression level of gene *i* across all samples and $\bar{x}_{ik}$ is the average expression level of the gene *i* across the samples that belong to class *k*.
They compared the performance of some classification methods, such as the *k* nearest neighbours (*k*-NN), CART and Linear Discriminant Analysis on a selection of 30 to 50 genes.
The main advantages of the filter methods are their computational efficiency and their robustness against overfitting. However, these methods do not take into account the interactions between genes, and they tend to select variables with redundant rather than complementary information (Guyon & Elisseeff, 2003). Furthermore, the gene selection that is performed in the first step of the analysis does not take into account the performance of the classification methods that are applied in the second step of the analysis (Kohavi & John, 1997).

The *wrapper* terminology was introduced by John et al. (1994). These methods involve successive evaluation of the performance of a gene subset and therefore, take the interactions between variables into account. The selection algorithm wraps the classification method, also named *classifier*, which evaluates the performance. The search for the optimal gene subset requires one to define 1) how to search the space of all possible variable subsets, 2) how to assess the prediction performance of a learning

machine to guide the search and 3) how to halt it. Of course, an exhaustive search is an NP-hard problem and when *p* is large, the problem is intractable and requires stochastic approximations. Furthermore, there is a risk of overfitting if the number of cases *n* is small. The number of variables to select must be fixed by the user, or chosen according to a criterion, such as the classification error rate. One of the main disadvantages of these methods is their computational cost that increases with *p*. Nonetheless, the wrapper strategy might be superior to the filter strategy in terms of classification performance, as was first shown by Aha & Bankert (1995) and John *et al.* (1994) in an empirical manner.

The *embedded methods* include variable selection during the learning process, without the validation step, to maximize the goodness-of-fit and minimize the number of variables which are used in the model. A well-known example is CART, where the selected variables split each node of the tree. Other approaches include greedy types of strategies, such as forward selection or backward elimination, that result in nested variable subsets. In a forward selection, variables are progressively included in larger and larger variable subsets, whereas the backward elimination strategy begins with all original variables and progressively discards the less relevant variables. According to the review of Guyon & Elisseeff (2003), these approaches are more advantageous in terms of computation time than wrapper methods, and should be robust against overfitting. The forward selection seems computationally more efficient than the backward elimination to generate nested variable subsets. However, the forward selection may select variable subsets that are not relevant, as the variable importance is not assessed with respect to the other variables, which are not included yet in the

model. As opposed to wrapper methods, the embedded methods define the size of the variable selection, which is often very small.

3.2 Recursive feature elimination

RFE (Guyon et al., 2002) is an embedded method which is based on a backward elimination and applies SVM to select an optimal non-redundant gene subset. The method relies on the fact that variables can be ranked on the basis of the magnitude of the coefficient β_i of each variable *i* when using a linear kernel SVM. In fact, each element β_i of the weight vector $\boldsymbol{\beta}$ is a linear combination of the cases, and most α_j are null, except for the *support* cases in the optimization problem (2.3). Consequently, the values in $\boldsymbol{\beta}$ can be directly interpreted as an importance measure of the variables in the SVM model. The variables *i* with the smallest weights $|\beta_i|$ will then be progressively discarded (recursive elimination) in the RFE procedure.

To speed up the computations, Guyon et al. (2002) proposed to discard several variables at a time in the algorithm, in spite of the fact that the classification performance may be altered. In this case, we obtain a ranking criterion on the variable subsets that are nested in each other, rather than a rank criterion on each variable. It is advised to discard variable subsets of various sizes, for example half of the variables in remaining set, to obtain sufficient density of information for the genes eliminated last. These latter will be ranked as first in the selection. Note that in any case, the user can actually choose the number of variables to select if desired.

RFE was first applied to microarray data (Ramaswamy et al., 2001) and was followed by numerous variants. SVM-RFE-annealing from Ding & Wilkins (2006) is based on a simulated annealing method and discards a large number of variables during the first iterations, and then reduces the number of discarded variables. This enables a

significant reduction in computation time. This method, which is very similar to RFE, requires a choice of the number of variables to select. Another example is SVM-RCE for Recursive Cluster Elimination (Yousef et al., 2007), to select correlated gene subsets and avoid missing important genes with small weights as they were correlated with some dominant genes. This stresses the issue of correlated genes that bring redundant information. Should they all be in the selection even if genes with complementary information may get a lower rank? Or should the selection be larger? Other variants were also proposed and the reader can refer to Tang et al. (2007); Mundra & Rajapakse (2007) or Zhou & Tuck (2007) for the MSVM-RFE for multiclass case. The abundant literature on this method shows the popularity of RFE for analysing microarray data.

3.3 Nearest shrunken centroids

Tibshirani et al. (2002) proposed a 'de-noised' version of the nearest-centroid rule defined in Section 2.3. The idea is to shrink the class centroids toward the overall centroids after standardizing by the within-class standard deviation for each gene. This gives higher weight to genes whose expression is stable within samples of the same class.

In the definition of the nearest centroid rule, the sample mean of the i^{th} gene in class k $\bar{\mathbf{x}}_{ik}$ is replaced by is shrunken estimate $\overline{\boldsymbol{x}}_{ik}^{*} = \overline{\boldsymbol{x}}_{ik} + m_k(s_i + s_0)d_{ik}^{*}$

With the following notations:

$$d_{ik}^{*} = \text{sign}(d_{ik})(|d_{ik}| - \Delta)_{+}, \tag{4.1}$$

where the + means positive part and zero otherwise, and $d_{ik} = \frac{\bar{x}_{ik} - \bar{x}_i}{m_k(s_i + s_0)}$; s_i is the pooled within-class standard deviation for gene *i* and s_0 is the median value of the s_i over the set of genes; $m_k = \sqrt{1/n_k - 1/n}$.

This shrinkage when computing d_{ik}^* is called soft thresholding, as the absolute value of d_{ik} is reduced by an amount of Δ and is set to zero if the result of (4.1) is less than zero. This allows variable selection as when Δ increases, many genes are eliminated from the class prediction and do not contribute to the nearest centroid computation. The shrinkage parameter Δ can be chosen by cross-validation.

Guo et al. (2007) then generalized the idea of Nearest Shrunken Centroids (NSC) with SCRDA (Shrunken Centroids Regularized Discriminant Analysis). Other variants of the nearest-shrunken centroids have been proposed in the literature for microarray data; see for example Dabney & Storey (2005) or Wang & Zhu (2007).

Nearest shrunken centroids are implemented in the *pamr* R package.

3.4 Random forests

Some classification methods are sensitive to small perturbations in the initial data set. For example, the construction of CART can dramatically vary if some values are modified in the data set. If a model does not generalize well, *i.e.* if its variance is large, a solution proposed by Breiman (1996) is to aggregate classifiers. The variance is consequently reduced, but the classifier interpretation becomes more difficult. This is why these techniques are sometimes called 'black box'. Breiman (1996) proposed to aggregate CART to reduce their variance by estimating each tree on a bootstrap sample. He introduced the bagging methodology for '*bootstrap aggregating*', by creating perturbed learning sets of the same size as the original sample. We describe a variant called Random Forests (Breiman, 2001) where an additional perturbation is

introduced when splitting the nodes of each tree to introduce more 'independence' between each tree.

Random forests is a wrapper method that became very popular for classification and feature selection in several contexts: Izmirlian (2004); Bureau et al. (2005); Diaz-Uriarte (2007) applied it with biological data, Svetnik et al. (2003) with QSAR data, Prasad et al. (2006) with ecological data etc. This approach, which at first seemed empirical is now theoretically studied, for example Biau et al. (2008) established some results regarding the consistency of aggregated trees.

The surprising paradox of random forests is that it benefits from the great instability of the classifiers CART by aggregating them. This approach combines two sources of randomness that largely improve the prediction accuracy: the bagging and the random feature selection to split each node of the tree. This results in low bias and low variance in the model.

Each tree (classification or regression) is constructed as follows:

1. B bootstrap samples $\{B_1, \dots, B_B\}$ are drawn from the original data.
2. Each sample $B_b (b = 1, \dots, B)$ is used as a training set to construct an unpruned tree T_b. Let p be the number of input variables of the tree, for each node of T_b, m variables are randomly selected ($m << p$) to determine the decision at the node, where m is constant during the forest growing. Then, the best split among these m predictors is chosen to split the node.

The predictions of the B trees are then aggregated to predict new data either by majority vote for classification or by average for regression.

Random forests also generate an internal estimation of the generalisation error by computing the out-of-bag error rate for each bootstrap sample. However, this error rate, which seems accurate and unbiased, cannot be used to evaluate the performance

of the variable selection. Indeed, Svetnik et al. (2003) showed that the OOB error estimate tends to overfit since the evaluation is not performed on an external test set. Instead, the variable selection should be evaluated on a test set sample. We will come back to the bias on the variable selection evaluation in Section 3.6.

The choice of the m randomly selected variables to split each node can be fixed by default to $\sqrt{p}$ (Liaw & Wiener, 2003). However, the number of trees *B* must be chosen by the user. To obtain stable results, in particular when the number of cases is small, we strongly advise to set a large number of trees to be large, *i.e.* ≥ 5000..

Two internal measures of variable importance are proposed in random forests, which allow for feature selection. These are called '*Mean Decrease Accuracy*' and '*Mean Decrease Gini*'. Both importance measures are described in Liaw & Wiener (2003) and in the *RandomForests* R package. Note that these measures can lead to different results if the data set contains a very small number of cases, or if some of the classes share similar (biological) characteristics.

3.5 Extension to multiclass

3.5.1 Division into binary problems

Multiclass problems could make feature selection easier than binary problems, as the more classes, the better the gene subset for a perfect classification task (Guyon & Elisseeff, 2003). But in practice, the multiclass case is difficult to deal with. Indeed, in the context of high dimensionality, the number of cases per class is usually smaller than in the binary problem due to experimental costs. This degrades the prediction accuracy when there are numerous classes. Furthermore, some authors noticed that

most of the classification errors were due to cases belonging to very similar classes, rather than being outliers (Yeang et al., 2001).

Some binary classification methods are naturally adapted to multiclass problems. This is the case for example for Linear Discriminant Analysis, CART or Nearest Centroid. Other methods require the decomposition of the multiclass problem into several binary problems, such as one class against the other (*1 vs. 1*) or one class against the rest (*1 vs. rest*). Another solution is to define multiclass objective functions. This solution was often addressed with SVMs. For example, Weston & Watkins (1999) and Lee & Lee (2003) proposed to solve the multiclass optimization quadratic problem directly into the SVM, rather than aggregating binary SVMs. The authors concluded that there was a smaller number of support vectors by directly solving the multiclass case than by aggregating binary SVMs. However, it is still less costly to solve several small binary problems rather than a big complex multiclass problem.

Dividing a multiclass problem into several binary problems requires one to choose the appropriate aggregation method. For example with SVM, one could choose majority vote, least square estimation based on weighting that involves weighting each SVM, or double layer hierarchical combining that aggregates SVMs outputs into another SVM (Kim et al., 2003). The type of binary classifier must also been chosen. Lee and Lee (2003) showed that the *1 vs. rest* SVM can give bad results if several classes are similar, and that the *1 vs. 1* SVM may contain a high variance, as each binary classifier is computed on a very small subset of cases with only one misclassifying cost for all classes. This latter problem is partly due to unbalanced classes.

A comparative study of several multiclass SVM approaches such as the Weston & Watkins (1999) or Lee & Lee (2003) approaches, *1 vs. rest*, and *1 vs. 1* was presented

in Statnikov et al. (2005) for microarray data, with first an initial pre-processing step with a filter method.

3.5.2 Unbalanced multiclass problems

In addition to numerous classes in microarray data, one often faces unbalanced classes. The main reason is that the class of interest is the rare one where data are difficult to obtain. There has been little attention given to the problem of unbalanced multiclasses in the context of microarray data, although of Eitrich & Lang (2006) and Qiao & Liu (2008) recently address this issue for general classification purposes. The main concern when performing feature selection in a classification context is that a classifier aims at minimizing the overall classification error rate. It thus minimizes the classification error rate of the majority classes, to the detriment of the minority classes. This type of approach has a serious drawback when performing feature selection, as the selected genes will mainly discriminate against the majority classes which are not the most biologically relevant.

In the case of random forests, Chen et al. (2004) proposed two approaches to balance the classes and to introduce a higher penalty when a minority class is misclassified. The first approach, called *Balanced Random Forests* (BRF) is based on a re-sampling technique. Each tree is constructed on the same number of cases in the majority and minority classes (sampling with replacement). The second approach, called *Weighted Random Forests* (WRF, currently implemented in the *RandomForests* R package) is based on cost sensitive learning. Weights are introduced in the RF algorithm, first during the tree construction, where class weights are used when splitting the nodes with the Gini criterion, and second when assigning the class of the terminal node. However, BRF risks overfitting the data if the number of cases in the minoritary class is very low, as this down sampling approach does not use many cases in the

majoritary classes. The inclusion of weights into the feature selection algorithm seems a better approach and was proposed in Lê Cao et al. (2008) in a stochastic wrapper algorithm.

For the SVM case, Qiao & Liu (2008) recently proposed an adaptive weighted learning procedure in the multiclass quadratic formulation of Lee & Lee (2003) to optimally weight each class.

3.6 Selection bias and performance assessment

In the classification context, the performance assessment of a variable selection remains difficult due to the small number of samples. As Dudoit & Fridlyand (2002) underlined, more cases would be needed to compute an accurate classification error rate. It is often unfeasible to obtain an external test set and the performance evaluation must often be computed on the learning set. Furthermore, several authors warn of the selection bias problem (Ambroise & McLachlan, 2002; Reunanen, 2003). Indeed, some articles presented extremely optimistic results as the classification error rate estimation *and* the variable selection were both performed on the learning set. Therefore, to correct for this selection bias, it is essential that cross-validation or the bootstrap be used external to the gene selection process.

In the present context where feature selection is used in training the prediction rule $r(\boldsymbol{x};\boldsymbol{t})$ from the full training set, the same feature selection method must be implemented in training the rule on the *V-1* subsets combined at each stage of an cross-validation of $r(\boldsymbol{x};\boldsymbol{t})$ for the selected subset of genes. Of course, there is no guarantee that the same subset of genes will be obtained as during the original training of the rule (on all the training observations). Indeed, with the huge number of genes

available, it generally will yield a subset of genes that has at most only a few genes in common with the subset selected during the original training of the rule.

In the case where the final version of the discriminant rule is based on a small subset selected in some optimal way from a much larger set of variables (genes), it is important that cross-validation is undertaken as described above, as otherwise a large selection bias can result; see Ambroise & McLachlan (2002); McLachlan *et al.* (2008); Wood *et al.* (2007); Zhu *et al.* (2008).

3.7 Optimal size of the selection

Choosing the optimal size of the selection is a difficult question as the small number of samples does not allow for an accurate estimate of the classification error. A naïve choice would be to select a number of genes that gives the lowest error rate. However, McLachlan et al., 2004, Chapter 7) showed on the van 't Veer et al. (2002) study that the estimated error rate needed to be corrected for bias. The authors showed that the minimum error rate was attained for approximately 256 genes when evaluating the gene selection with bias correction on the whole data set, instead of only 70 genes as originally proposed in this study.

A solution to choose the optimal set of genes would be to select the genes which give a stabilized error rate and, therefore, consistent predictive results.

4 Illustrative example with the Golub data set

4.1 Performance of the three feature selection methods

As an illustrative example, we considered the well known leukaemia data set (Golub et al., 1999), where Affymetrix oligonucleotide arrays were used to measure gene

expressions in two types of acute leukaemias: acute lymphoblastic leukaemia (ALL), and acute myeloid leukaemia (AML). The entire data set consists in 72 tissue samples, among which 47 are ALL cases and 25 are AML, and the measurement of 7,129 genes. The data set was pre-processed as in Dudoit & Fridlyand (2002) by filtering and log transforming the data. The final data set comprises 3,731 genes.

We performed external 10-fold cross validation $A^{(CV10E)}$ as used by Ambroise & McLachlan (2002) for different sizes of selected subsets of genes to evaluate the performance of Recursive Feature Elimination (RFE), Nearest Shrunken Centroids (NCS), and Random Forests (RF). For the 10-folds, we divided the 72 tissues into balanced training and test sets such that approximately 42 ALL and 22 AML were used for training, 5 ALL, and 3 AML were used for testing in the binary problem. We calculated the 10-CV estimated error rates over 50 random splits.
The averaged values of these estimates are plotted in Figure 1. It can be seen from this figure that all three wrapper methods perform similarly, except for NCS that requires a larger selection of genes to be competitive with the other approaches. $A^{(CV10E)}$ was found to have little bias when estimating the error rate (Ambroise & McLachlan, 2002). However, the conclusion about this graph should be taken with caution, as the error rate should be corrected for bias.

As an illustrative example, this data set is of interest as the ALL cases can be divided into 2 subclasses, called ALL-B cells (38 samples) and ALL-T cells (9 samples). We are here in the typical case of unbalanced multiclass data set, where the ALL-T class is the minority class. Therefore, when performing external balanced 10-fold cross-validation, 34 ALL-B, 8 ALL-T, and 22 AML were used for training and

approximately 4 ALL-B, 1 ALL-T, and 3 AML were used for testing. The fact that there is only one ALL-T sample in the test set may severely affect the estimation of a too optimistic error rate. Indeed, as mentioned in Section 3, when computing $A^{(CV10E)}$, we tend to neglect misclassified cases from of the minority class.

The averaged values of the $A^{(CV10E)}$ estimates are plotted in Figure 2 over 50 random splits. In this multiclass problem, the estimated error rate is higher than in the binary case presented above where the gene selection is small. Therefore, a larger selection of genes might be advisable for further biological validation. Interestingly, the stabilized error rate seems to be similar to the one obtained in the binary case (around 5%). This may be due to the fact that there is only one ALL-T sample in the test set that can be misclassified. This may result in a too optimistic estimation of the error rate. Indeed (not shown), the error rate for RFE was between 0.1 and 0.2 for the ALL-T minority class, and between 0.01 and 0.02 for the two other classes. Since in this last example the classes are strongly unbalanced, a better way to take into account the minority class would be to weight the error rate estimation according to the proportion of samples in each class, as was proposed in Lê Cao et al. (2008).

4.2 Comparison of the gene selections

We arbitrarily chose a selection size of 50 genes and compared the overlap between the selected genes resulting with each approach, for the binary and the 3-class cases (Figure 3). Note that the same trend was observed when the selection size was increased.

It is interesting to see that although each approach uses a different classifier, a fair amount of genes are commonly selected by the three methods (20 and 15 genes for the binary and the multiclass problems). Therefore, these approaches have the ability to

select (the same) discriminative genes and these discriminative genes may be of potential relevance for the biological experiment.

Half of the genes selected with RFE differed from those selected with RF and NSC. This difference might be due to the fact that as a backward technique, RFE tends to select non-redundant and non-correlated genes (Yousef et al., 2007) whereas NSC and RF can highlight correlated genes in their selections.

As expected, when the number of classes increased from $g = 2$ to $g = 3$, the overlap between all three methods became smaller. This can be explained by the increasing complexity of the data set, where numerous subsets can lead to a good classification of the samples.

As an illustrative example, Figure 4 displays the heat map of the 50 genes selected with NSC for the multiclass case, with Euclidian distance and Ward aggregation method. This type of unsupervised clustering enables a global overview of the genes that were selected with respect to each tissue sample.

For the binary problem (not shown), it was surprising to see that although RFE selected genes with a poor contrast (mostly under-expressed genes), it allowed for a perfect classification of the tissue samples, whereas RF and NSC seemed to select interesting and contrasted gene clusters, but with one misclassified sample.

The same trend could be observed for the 3-class case, where contrasted gene clusters were obtained with RF and NSC (Figure 4). One would expect the ALL-T and ALL-B to share the same dendogram as they are a subclass of ALL. In fact it is the AML samples that seem to share similarities with ALL-B with these gene selections.

4.3 Choice of method

The large difference between the three feature selection methods, and therefore the three gene selections did not really appear when estimating the classification error rate

(Figures 1 and 2). It is highly probable that different gene subsets can lead to the same classification performance of a given classifier. Nevertheless, some genes were commonly selected by all 3 approaches, despite the fact that these statistical approaches differ in their construction and the classifiers they use. It is therefore difficult to choose the appropriate statistical method to perform variable selection and we cannot have a definitive answer for this question. Microarray data are very complex and the statistical outcome highly depends on the biological experiment, design and the quality of the data. Furthermore, some statistical approaches might be appropriate in one study, but not in another. Therefore, one has to take into account different criteria as proposed in this illustrative section, compare several statistical approaches, as well as to investigate the biological relevance of the selected genes related to the biological experiment.

5 Validation

Validation of the results has been often discussed in the literature. Once the gene signatures have been selected, their clinical utility must be established. For example, they must prove to reliably identify patients with poor or good prognosis. The first step consists in validating the microarray experiment, while the second part consists in an independent validation using these gene signatures.

5.1 Biological interpretation

Once the gene signatures have been selected using a statistical approach, it is of tradition to validate the results and look for false positive by using the same samples,

but on different mRNA measurement procedure, such as reverse-transcriptase PCR. This may highlight erroneous inferences due to poor measurement quality. However, repeating measurement on the same biological samples but with a different measurement technique is a highly debatable practice to validate the microarray experiment.

Post hoc analysis is then required to assess the biological relevance of the gene list. For example, pathway analysis, using softwares such as DAVID (Dennis et al., 2003), Panther (Mi et al., 2005), FatiGO (Al-Shahrour et al., 2004)-to cite a few, can be performed on the gene selection to identify biological functions and networks; see also Lê Cao et al. (2007) for an example of such analysis. This type of analysis also enables one to highlight other genes that are strongly correlated to the selected genes and interact with these genes in biological pathways, but might not be spotted on the microarray, or were discarded during the pre-processing step because of poor quality spots.

5.2 Independent test set

The gene signatures then need to be proven that they provide additional information to the clinicopathologic risk criteria that are currently used in the clinic. The validation must hence be performed completely independently, not only on a new batch of patients, but also by external institutions to the study. In addition, it should also be applied to a prospective study, rather than using retrospective data of patient that may not be representative of the nowadays breast cancer population.

Buyse et al. (2006) performed this type of analysis, using independent statisticians and multinational collaborations to assess the usefulness of the 70-gene signature in breast cancer on a retrospective study. They showed that this set of gene had

reproducible prognostic value across different patient populations, laboratories and biostatistical facilities. However, many questions remain, such as the lack of gene overlap among different studies (Michiels et al., 2005). Some authors argue that these different gene selection that predict the same outcome might be the result of differences among microarray platforms, but also the differences among the genes spotted on the array or the different experimental conditions. Others state that the resulting lists of genes are highly unstable as it depends on the patients on the training set.

6 Conclusion

Microarray technology is a promising and a powerful high-throughput tool for researchers in many fields of biology and medicine. Microarray analysis has the potential to refine cancer prognosis, well beyond the currently used clinical parameters to predict disease outcome. Diagnostic assays developed on gene expression profiling studies will therefore benefit to oncology and other areas of medicine.

Many studies showed that supervised classification methods appear to be one of the best approaches to identify prognostic and predictive profiles (Golub et al., 1999; van 't Veer et al., 2002; Nuyten & van de Vijver, 2008). Further studies are required to check the consistency of the results obtained with these sophisticated statistical approaches before they can replace the current clinical and pathological indicators and be made available to patients.

It would be interesting to further investigate the integration of clinical data and microarray data to improve the prediction performance of the classification methods. Gevaert et al. (2006) and McLachlan & Ng (2008) have shown that a significant improvement could be achieved by using Bayesian networks or expert networks to integrate both discrete and continuous data on the van ’t Veer breast data set. Clinical variables are often under-used when analysing microarray data. Combined with often noisy gene expression data, they would allow for a better cancer prognosis as they have a very low noise level (Gevaert et al., 2006).

References

Aha D., and Bankert R. (1995). A comparative evaluation of sequential feature selection algorithms. *In* "Learning from Data: Artificial Intelligence and Statistics V", pp. 199–206, Springer.

Al-Shahrour F., Diaz-Uriarte R., and Dopazo J. (2004). FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. *Bioinformatics* **20:** 578-580.

Ambroise C., and McLachlan G. (2002). Selection bias in gene extraction on the basis of microarray gene-expression data. *Proceedings of the National Academy of Sciences* **99:** 6562-6566.

Biau G., Devroye L., and Lugosi G. (2008). Consistency of random forests and other averaging classifiers. *The Journal of Machine Learning Research* **9:** 2015-2033.

Breiman L. (1996). Bagging predictors. *Machine Learning* **24:** 123-140.

Breiman L. (2001). Random forests. *Machine Learning* **45:** 5-32.

Breiman L., Friedman J., Olshen R., and Stone C. (1984). "*Classification and regression trees*," The Wadsworth statistics/probability series, Belmont, CA.

Bureau A., Dupuis J., Falls K., Lunetta K., Hayward B., Keith T., and Van Eerdewegh P. (2005). Identifying SNPs predictive of phenotype using random forests. *Genetic Epidemiology* **28:** 171-182.

Burges C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* **2:** 121-167.

Buyse M., Loi S., van 't Veer L., Viale G., Delorenzi M., Glas A., Saghatchian d'Assignies M., Bergh J., Lidereau R., and Ellis P. (2006). Validation and clinical utility of a 70-gene prognostic signature for women with node-negative breast cancer. *Journal of the National Cancer Institute* **98:** 1183-1192.

Chen C., Liaw A., and Breiman L. (2004). Using random forests to learn unbalanced data, Dpt of Statistics, University of Berkeley.

Cristianini N., and Shawe-Taylor J. (1999). "*An introduction to support vector machines: and other kernel-based learning methods*", Cambridge University Press New York, NY, USA.

Dabney A., and Storey J. (2005). Optimal feature selection for nearest centroid classifiers, with applications to gene expression microarrays. *UW Biostatistics Working Paper Series*, article 267.

Dennis G. Jr, Sherman B., Hosack D., Yang J., Gao W., Lane H., and Lempicki R. (2003). DAVID: database for annotation, visualization, and integrated discovery. *Genome Biology* **4**: article R60.

Diaz-Uriarte R. (2007). GeneSrF and varSelRF: a web-based tool and R package for gene selection and classification using random forest. *BMC Bioinformatics* **7:** article 328.

Ding Y., and Wilkins D. (2006). Improving the performance of SVM-RFE to select genes in microarray data. *BMC Bioinformatics* **7:** article S12.

Dudoit S., and Fridlyand J. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association* **97:** 77-87.

Efron B. (1979). Bootstrapping methods: another look at the jackknife. *Annals of Statistics* **7:** 1-26.

Efron B. (1983). Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American Statistical Association* **78:** 316-331.

Efron B., and Tibshirani R. (1997). Improvements on cross-validation: the. 632+ bootstrap method. *Journal of the American Statistical Association* **92:** 548-560.

Eitrich T., and Lang B. (2006). Efficient optimization of support vector machine learning parameters for unbalanced datasets. *Journal of Computational and Applied Mathematics* **196:** 425-436.

Gevaert O., Smet F., Timmerman D., Moreau Y., and Moor B. (2006). Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks. *Bioinformatics* **22:** 184-190.

Golub T., Slonim D., Tamayo P., Huard C., Gaasenbeek M., Mesirov J., Coller H., Loh M., Downing J., and Caligiuri M. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* **286:** 531-537.

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

Guyon I., and Elisseeff A. (2003). An introduction to variable and feature selection. *The Journal of Machine Learning Research* **3:** 1157-1182.

Guyon I., Weston J., Barnhill S., and Vapnik V. (2002). Support vector machine with recursive feature selection. *Machine Learning* **46:** 389–422.

Izmirlian G. (2004). Application of the random forest classification algorithm to a SELDI-TOF proteomics study in the setting of a cancer prevention trial. *Annals of the New York Academy of Sciences* **1020:** 154-174.

John G., Kohavi R., and Pfleger K. (1994). Irrelevant features and the subset selection problem. *In* "Proceedings of the Eleventh International Conference on Machine Learning", New Brunswick, NJ, USA, Morgan Kaufmann.

Kim H., Pang S., Je H., Kim D., and Yang Bang S. (2003). Constructing support vector machine ensemble. *Pattern Recognition* **36:** 2757-2767.

Kohavi R., and John G. (1997). Wrappers for feature subset selection. *Artificial Intelligence* **97:** 273-324.

Lê Cao K.-A., Bonnet A., and Gadat S. (2008). Multiclass classification and gene selection with a stochastic algorithm. *Computational Statistics and Data Analysis (in press)*.

Lê Cao K.-A., Goncalves O., Besse P., and Gadat S. (2007). Selection of biologically relevant genes with a wrapper stochastic algorithm. *Statistical Applications in Genetics and Molecular Biology* **6:** article 29.

Lee Y., and Lee C. (2003). Classification of multiple cancer types by multicategory support vector machines using gene expression data. *Bioinformatics* **19:** 1132-1139.

Li C., Tseng G., and Wong W. (2003). Model-based analysis of oligonucleotide arrays and issues in cDNA microarray analysis. *In* "Statistical Analysis of Gene Expression Microarray Data. Chapman & Hall, NY" (T. Speed, Ed.), pp. 1–34.

Liaw A., and Wiener M. (2003). Classification and regression by randomForest. *R news* **2/3:** 18-22.

McLachlan G. (1977). A note on the choice of a weighting function to give an efficient method for estimating the probability of misclassification. *Pattern Recognition* **9:** 147-149.

McLachlan G. (1992). "*Discriminant analysis and statistical pattern recognition*", Wiley New York.

McLachlan G., Chevelu J., and Zhu J. (2008). Correcting for selection bias via cross-validation in the classification of microarray data. *In* "Beyond parametrics in Interdisciplinary Research: Festschrift in Honor of Professor Paranab K. Sen" (N. Balakrishnan, E. Pena, and M.J. Silvapulle, Eds.) Hayward, California: IMS Collections, Vol. 1, pp. 364-376.

McLachlan G., Do K., and Ambroise C. (2004). "*Analyzing microarray gene expression data*", Wiley-Interscience.

McLachlan G., and Ng S.-K. (2008). Expert networks with mixed continuous and categorical feature variables: a location modeling approach. *In* "Machine Learning Research Progress" (H. Peters, and M. Vogel, Eds.), pp. 1-14, Hauppauge, New York.

Mi H., Lazareva-Ulitsky B., Loo R., Kejariwal A., Vandergriff J., Rabkin S., Guo N., Muruganujan A., Doremieux O., and Campbell M. (2005). The PANTHER database of protein families, subfamilies, functions and pathways. *Nucleic Acids Research* **33:** 284-288.

Michiels S., Koscielny S., and Hill C. (2005). Prediction of cancer outcome with microarrays: a multiple random validation strategy. *The Lancet* **365:** 488-492.

Mundra P., and Rajapakse J. (2007). SVM-RFE with relevancy and redundancy criteria for gene selection. *Lecture Notes in Computer Science* **4774:** 242-252.

Nuyten D., and van de Vijver M. (2008). Using microarray analysis as a prognostic and predictive tool in oncology: focus on breast cancer and normal tissue toxicity. *In* "Seminars in radiation oncology", pp. 105-114.

Prasad A., Iverson L., and Liaw A. (2006). Newer classification and regression tree techniques: bagging and random forests for ecological prediction. *Ecosystems* **9:** 181-199.

Qiao X., and Liu Y. (2008). Adaptive weighted learning for unbalanced multicategory classification. *Biometrics (in press).*

Ramaswamy S., Tamayo P., Rifkin R., Mukherjee S., Yeang C., Angelo M., Ladd C., Reich M., Latulippe E., and Mesirov J. (2001). Multiclass cancer diagnosis

using tumor gene expression signatures. *Proceedings of the National Academy of Sciences* **98:** 15149-15154.

Reunanen J. (2003). Overfitting in making comparisons between variable selection methods. *The Journal of Machine Learning Research* **3:** 1371-1382.

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

Svetnik V., Liaw A., Tong C., Culberson J., Sheridan R., and Feuston B. (2003). Random forest: a classification and regression tool for compound classification and QSAR modeling. *Journal of chemical information and computer sciences* **43:** 1947-1958.

Tang Y., Zhang Y., and Huang Z. (2007). Development of two-stage SVM-RFE gene selection strategy for microarray expression data analysis. *IEEE ACM Transactions on computational biology and bioinformatics* **4:** 365-389.

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

van 't Veer L., Dai H., van de Vijver M., He Y., Hart A., Mao M., Peterse H., van der Kooy K., Marton M., and Witteveen A. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature* **415:** 530-536.

Vapnik V. (2000). "*The Nature of Statistical Learning Theory*", Springer.

Wang S., and Zhu J. (2007). Improved centroids estimation for the nearest shrunken centroid classifier. *Bioinformatics* **23:** 972-979.

Weston J., and Watkins C. (1999). Multi-class support vector machines. *In* "Proceedings ESANN, Brussels".

Wood I., Visscher P., and Mengersen K. (2007). Classification based upon gene expression data: bias and precision of error rates. *Bioinformatics* **23:** 1363-1370.

Yeang C., Ramaswamy S., Tamayo P., Mukherjee S., Rifkin R., Angelo M., Reich M., Lander E., Mesirov J., and Golub T. (2001). Molecular classification of multiple tumor types. *Bioinformatics* **17:** 316-322.

Yousef M., Jung S., Showe L., and Showe M. (2007). Recursive cluster elimination (RCE) for classification and feature selection from gene expression data. *BMC Bioinformatics* **8:** 144.

Zhou X., and Tuck D. (2007). MSVM-RFE: extensions of SVM-RFE for multiclass gene selection on DNA microarray data. *Bioinformatics* **23:** 1106-1114.

Zhu J., McLachlan G., Ben-Tovim Jones L., and Wood I. (2008). On selection biases with prediction rules formed from gene expression data. *Journal of Statistical Planning and Inference* **138:** 374-386.

Figure Captions

Figure 1. Estimation of the classification error rate for each method with external 10-fold cross-validation (repeated 50 times) with respect to the number of genes selected, for the binary problem.

Figure 2. Estimation of the classification error rate for each method with external 10 fold cross-validation (repeated 50 times) with respect to the number of genes selected, for the unbalanced multiclass problem.

Figure 3. Venn diagrams. Overlap between the gene lists selected with Random Forests, Recursive Feature Elimination and Nearest Shrunken Centroids (selection of 50 genes for each method).

Figure 4. Heat map for the 50 genes selected with Nearest Shrunken Centroids for the unbalanced multiclass problem. Rows (genes) and columns (tissues) are arranged according to a hierarchical clustering method. Tissue classes are indicated by color bars on the upper dendogram (black: ALL-T, grey: AML, and light grey: ALL-B)