A Cross-Validation Study to Select a Classification Procedure for Clinical Diagnosis Based on Proteomic Mass Spectrometry

A Cross-validation Study to Select a Classification Procedure for Clinical Diagnosis Based on Proteomic Mass Spectrometry Data

Dirk Valkenborg*, Suzy Van Sanden*, Dan Lin, Adetayo Kasim, Qi Zhu, Philippe Haldermans, Ivy Jansen, Ziv Shkedy, Tomasz Burzykowski

Center for Statistics, Hasselt University, Agoralaan 1 building D, 3590 Diepenbeek, Belgium.

* Both authors contribute equally.

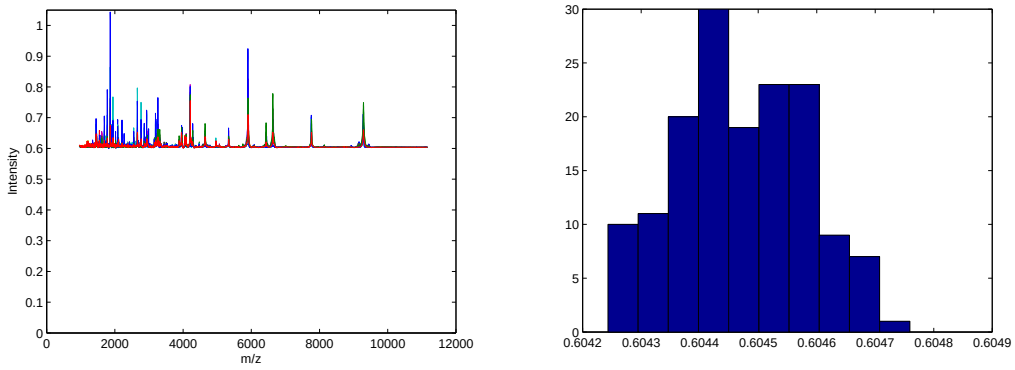
1 Introduction

Analysis of protein content of samples can play an important role in disease diagnostics. For instance, Petricoin *et al.* (2002) used SELDI TOF mass spectra to discriminate between ovarian cancer and normal samples. They reported a construction of a proteomic pattern that provided 100% sensitivity and 95% specificity. The estimated values of sensitivity and specificity were impressive and the results deservedly attracted a lot of attention. In 2004 the same team published results of an additional analysis of the data, using a higher resolution technique called hybrid quadrupole time-of-flight (QqTOF) mass spectrometry (Conrads *et al.*, 2004). Using the same biological samples as Petricoin *et al.* (2002), they constructed a pattern capable of achieving a 100% sensitivity and 100% specificity for identifying cancer from normal.

Reports as those just mentioned increase the interest in the use of protein mass spectrometry for classification and diagnostic purposes. However, there are potential pitfalls. Baggerly *et al.* (2004), (2005) re-examined the data of Petricoin *et al.* (2002) and Conrads *et al.* (2004) and encountered problems with the reproducibility of the results. One of the issues was that the classification rule constructed by Petricoin *et al.* (2002) used features (intensity measurements at particular locations) of the mass spectra found in the regions likely to be strongly affected by random noise. Moreover, problems with baseline correction and calibration of the spectra were discovered, that might have influenced the construction of the rule and the reproducibility of its findings.

This example clearly illustrates the need for a careful development of methods that would allow use of mass spectrometry data for classification purposes. An exercise like that proposed by the organizers of the "Classification Competition on Clinical Mass Spectrometry Proteomic Diagnosis Data" is an interesting step in this direction.

Analyses performed by Baggerly *et al.* (2004), (2005) also clearly underline the importance of pre-processing of mass spectra, aimed at the removal of systematic effects, before the use of the data for classification. Following this logic, we attempted to pre-process the training mass spectrometry data, provided by the organizers of the competition, before constructing a classification rule. Our aim was to select features of the spectra that are likely due to true biological signals (i.e., peptides). As a result, we selected a set of 92 features. Next, to construct the classification rule, we considered using 8 methods of choosing a subset of the features, combined with 7 classification methods. We assessed the performance of the $7 \times 8 = 56$ combinations by using a cross-validation procedure. The best result, as indicated by the lowest overall misclassification rate, was obtained for the use of the whole set of 92 features as the input for a support-vector machine (SVM) with



(a) (b)

Figure 1: Baseline correction. Panel a) shows an arbitrary spectrum of the provided data. Panel b) shows a histogram which indicate a low baseline fluctuation.

a linear kernel. This method was therefore used to construct the classification.

Our report is organized as follows. In Section 2 we describe in more detail the pre-processing strategy and the approach used to select the optimal classification procedure. Section 3 presents the results of the cross-validation study undertaken to select the classification procedure, and the results of the application of the chosen procedure to the calibration dataset provided by the organizers of the competition. Section 4 closes the report with some concluding remarks.

2 Methodology

In this section we describe the pre-processing strategy and the approach used to select the optimal feature selection/classification strategy.

2.1 Pre-processing

The importance of pre-processing was already motivated in the introduction. We used baseline correction, dimensionality reduction, clustering for feature selection, and intensity normalization.

2.1.1 Baseline correction

The intensity values measured in a spectrum are used as a measure for the relative abundance of a peptide in a sample. However, before the intensity measurements can be used, the baseline shift should be removed, so that it does not influence the analyte intensity. A rigid baseline correction was already performed on the data by the organizers of the competition. Figure **??** shows an example of how the provided data look like. The 0.6 offset is an artefact from a log-transform of a preprocessing step and should not be confused with the baseline. However, we could still detect small baseline fluctuation around this offset as indicated in Figure 1(b). Regardless the magnitude of the baseline variability, we choose to remove this effect from the data before attempting any further analysis. In this case, the baseline was found by calculating (and subtracting from the intensity measurements) the median value of the observed local minima in a spectrum. All negative values were truncated to zero. If the small baseline fluctuations affected the classification rule was not investigated, however we argue that removing baseline effects is good practise.

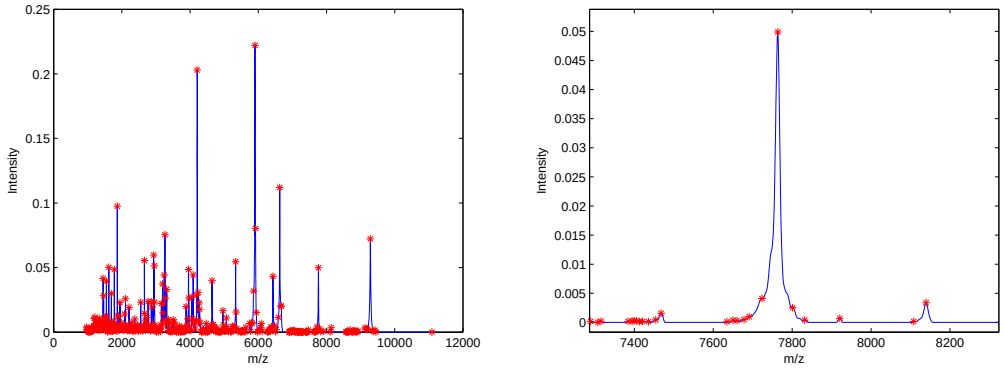
2.1.2 Dimensionality reduction and noise filtering

The provided spectra contained $11,205$ intensity measurements obtained by using a variable binning window on a grid of roughly $30,000$ bins. This is still a large number of potential variables that could be used in a classification procedure. Some of the measurements are likely to be noise-generated, though. To reduce the dimensionality of the problem, and in an attempt to filter out noise, we first selected all the local maxima in a spectrum (indicated in Figure 2 by red stars). Figure 2(b) shows that there were many low intense local maxima. These local maxima were assumed to be most likely due to noise and were removed from the data by using a threshold of 0.005.

A disadvantage of this method is that it only captures information about the height of peaks in the mass spectrum. Information about the shape of the peaks is removed during this process and thereby we can possibly miss peptides which might be hiding in the shoulders of larger peaks.

It would be desirable to improve the resolution of the MALDI-experiments, such that information about the isotopic variants becomes available. Then, a more meaningful peptide selection algorithm could have been applied (Breen *et al.*, 2000; Valkenborg *et al.*,2007; Valkenborg *et al.*,2008). The algorithm uses the fact that the height of peaks depends on the proportional distribution of atomic isotopes composing a peptide. Prior chemical knowledge about the distribution, and hence about the expected height of the peaks, can be used to reduce the dimensionality of the data and discriminate between a valid peptide peak and peaks originating from noise.

Unfortunately, as already mentioned, this procedure could not be applied to the data at hand because for this form of MALDI-experiments, the grids were chosen fairly rough due to the poor resolution. Thus, an additional noise-filtering step was implemented, as described in the next section.



(a) (b)

Figure 2: Baseline corrected spectrum with the local maxima indicated by a red star. Panel a) shows an arbitrary baseline corrected spectrum over its entire range. Panel b) shows a close-up of the spectrum from panel a) in the mass region 7400 Da to 8200 Da.

2.1.3 Feature extraction

To additionally distinguish noise-generated peaks from those that might be due to valid peptides, we assumed that the latter would manifest themselves as peaks consistently appearing around the same mass-to-charge m/z-value, for (almost) all spectra from a specific group. Figure 3 shows a heatmap of intensity measurements, with mass-to-charge on the horizontal axis and an arbitrary spectrum number on the vertical axis. Many apparent long stretches of high intensity measurements across the ordinate can be observed. We assumed that these stretches were likely due to peptides. In order to define the m/z location of the peptides, a bi-dimensional clustering algorithm was used. It consisted of two independent steps:

- First, all points in the heatmap as in Figure 3, were projected on the m/z-axis.The resulting projections were well separated over the m/z-axis because of the selection of local maxima, as described in Section 2.1.2. Hence, on the m/z-axis, clustering was performed using a window of 2 Da. That is, the

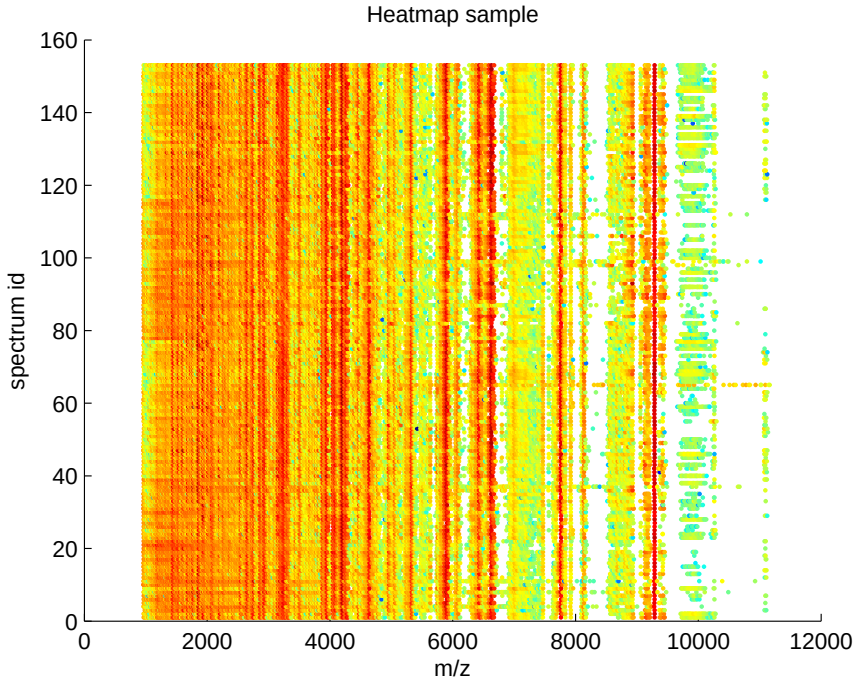
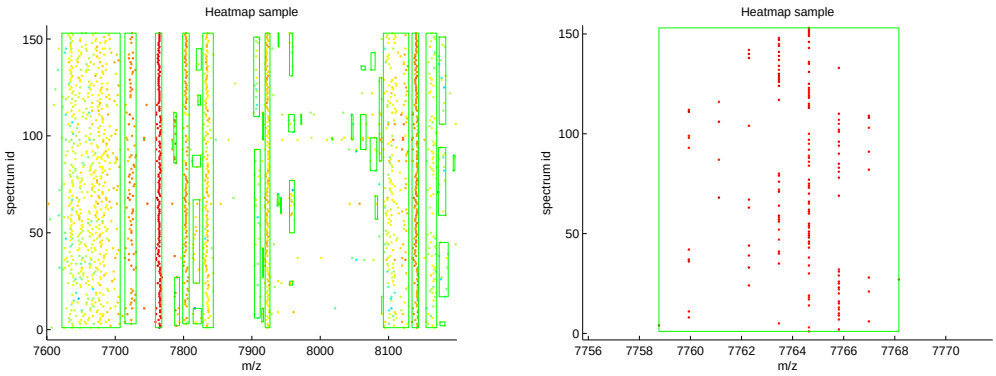


Figure 3: Heatmap of the local maxima. The ordinate represents the spectrum number and the abscis indicates the mass. The intensity of the local maximum is indicates by a color (red = high intense, blue = low intense)

maximum distance from a projection point to a cluster was assumed to be equal to 2 Da. The threshold was chosen empirically; lower values resulted in too many small clusters in a heatmap, while larger thresholds were yielding stretches with too much m/z variability. In other words, the outcome of the first step is a set of clusters with the locations of the spectral local maxima across the samples on the m/z axis.

- Second, each cluster of spectral local maxima obtained in the first clustering step are now projected on the vertical "heatmap" axis. The maximum distance of a projection point to a cluster was assumed to be 5 units (spectra). The threshold was again chosen empirically to arrive at a set of not too small clusters.

Only clusters (stretches) that contained less then 170 points and more than 70 point were included in the feature list. This choice was made based on the idea that peptide-related peaks should be seen in many spectra. If a peptide were present



(a) (b)

Figure 4: Clusters found in the heatmap of local maxima. Panel a) shows the clusters between a mass range of 7600 and 8200 Da. Panel b) shows a close-up of a cluster at 7765 Da.

in all spectra, the cluster (stretch in a heatmap) should contain about 153 points (some duplicates were allowed by taking the threshold of 170). On the other hand, it should be present at least in about half (70) of the spectra. Note that, if a peptide were present in almost all spectra from only one group of samples (cases or controls), the cluster (stretch) would contain about 70 points.

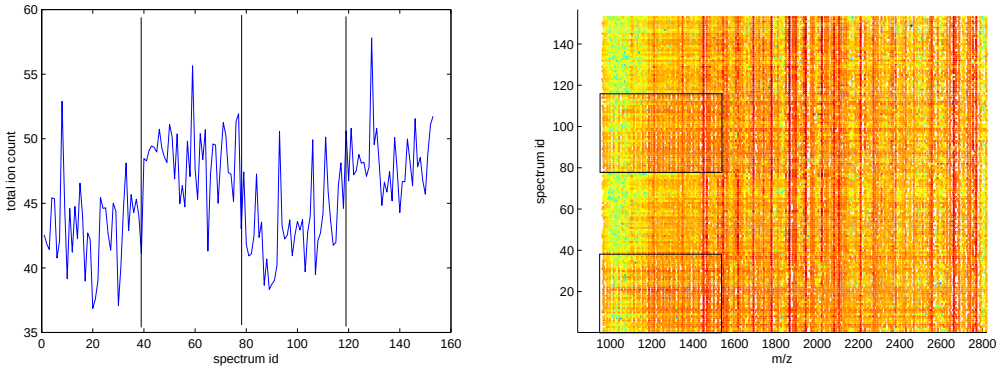
The result of this clustering step is illustrated in Figure 4(a). Note that the clusters correspond to the spectral peaks observed in Figure 2(b). It is worth mentioning that even the small local maxima on the shoulder of the peak at 7765 Da are detected.

When the clustering procedure was applied to the calibration set of 153 spectra, 92 clusters (stretches) were selected. The outcome of the feature extraction procedure were the m/z intervals corresponding to the resulting clusters. These intervals defined a region along the m/z-axis, wherein a spectral local maxima, possibly related to a peptide, can be found. The intensity values of the spectral local maximum found in the defined interval are kept across the samples and were to be used in a classification method.

Note that some of the 92 clusters could miss a point for a spectrum, because a peak was not present in the spectrum in the cluster-defining m/z interval. In such cases, the missing intensity value for the spectrum was imputed directly as the largest baseline-corrected intensity measure within the m/z interval. On the other hand, if two or more points for a particular spectrum were included in a cluster, the

intensity value of the peak with the m/z coordinate closest to the mean value of m/z coordinates of all points in the cluster was selected. The intervals specified by the calibration set are also used to classify a new sample.

2.1.4 Peptide quantification and normalization



(a) (b)

Figure 5: Possible plate or pin tip effect. Note, that 4 distinct regions with fluctuating intensities can be identified (indicated by black lines). Panel a) shows the fluctuation of the total ion count across the 153 mass spectra. Panel b) shows a heatmap of the local maxima in the lower mass region.

Intensity measurements in a MALDI-TOF mass spectrum can be influenced by many factors, like sample degradation, plate effect, laser intensity, matrix crystallization, ionization efficiency of a peptide, and the absolute abundance of the peptide in a sample. The latter is the main effect of interest: if there is a higher abundance of a peptide then we want it to be reflected by the height of a peptide peak. However, even for a constant peptide abundance, fluctuations in the other factors can influence the value of the intensity measurement. This is illustrated in Figure 5 as fluctuation in the total ion count (TIC, the sum of all intensity values) of the baseline corrected spectra. The higher TIC for some spectra might be due to, e.g., a higher amount of biological material spotted on a plate. This in turn might bias the comparison of peptide abundance across spectra. Heatmap 5(b) displays the effect of these fluctuations as intensity differences for the selected candidate peptide peaks.

To correct for the TIC fluctuations, the intensity values obtained for the 92 features found by the clustering algorithm described in the previous section were standardized by using the total ion count of a spectrum. More specifically, feature intensity F_{ij} of the jth measurement ($j = 1, \ldots, 92$) in baseline-corrected spectrum i ($i = 1, \ldots, 153$) was standardized by re-weighting it as follows:

$$F'_{ij} = F_{ij} \left(\frac{\sum_{l=1}^{L} I_{il}}{\sum_{k=1}^{K} \sum_{l=1}^{L} I_{kl}} \right)^{-1} = I_{ij} \left(\frac{\text{TIC}_i}{\sum_{k=1}^{K} \text{TIC}_k} \right)^{-1}, \qquad (1)$$

where $L = 11205$ is the number of intensity measurements I in a baseline-corrected spectrum, and $K = 153$ is the number of spectra in the calibration (training) dataset.

The same procedure is used for the classification of a new sample. The factor $\sum_{k=1}^{K} \text{TIC}_k$ is unchanged and is reused for the standardization of the new feature intensities.

2.2 Selection of the classification procedure

In this section we describe the approach that was applied to choose the classification procedure that would perform best for the type of data at hand.

We considered using a two-stage classification procedure. At the first stage, a subset (or all) of the 92 available features, found by the clustering algorithm described in Section 2.1.3, would be selected. At the second stage, a classification rule would be constructed using the selected subset of features.

We considered using one of 8 different selection criteria (statistics) at the first stage of the procedure. Additionally, we allowed for the selection of a subset of 10, 20, 30, 50, or all features. For the second stage, we considered using one of 7 classification methods. This resulted into $8 \times 5 \times 7 = 280$ possible approaches. All the approaches were applied to 1000 re-sampled data sets, and their misclassification error rate was estimated.

The choice of methods for the use at the first and second stage was partially motivated by the results of simulations performed in a microarray setting (Van Sanden *et al.*, 2007).

In most cases existing R functions were used for the implementation of the different methods. In what follows the necessary packages, as well as the parameter settings of the particular functions, are indicated.

2.2.1 Feature-subset selection criteria

We considered 8 statistics for selecting a subset of features for the purpose of the construction of a classification rule. For each statistic, p features with the highest

values are selected ($p = 10, 20, 30, 50$). Several methods were proposed for the purpose of analysing microarrays. For these methods we keep the description in terms of genes and gene-expression.

Wilcoxon rank sum (Wilc)

We considered the basic non-parametric test statistic in the form of the Wilcoxon rank sum test (Wilc). It was applied with the help of the R-function *wilcox.test* from the `stats` package.

Significance Analysis of Microarrays (SAM)

SAM is a method for analysing microarray experiments and detecting significant genes. It was proposed by Tusher *et al.* (2001). A score is assigned to each gene based on change in gene expression relative to the standard deviation augmented by a small positive constant. This constant ensures that the variance of the score is independent of gene expression. Its value is chosen to minimize the coefficient of variation of the test statistic. The t-statistic for the case of two unpaired classes was calculated by the *samr* function from the `SAMR` package.

Prediction Analysis for Microarrays (PAM)

PAM fits a nearest shrunken centroid classifier to microarray data. The method, also referred to as soft-thresholding, was introduced by Tibshirani *et al.* (2002). It provides a list of significant genes whose expression best characterizes each class. The functions *pamr.train* and *pamr.listgenes* from the `PAMR` package implement the method.

Extreme-value-distribution-based gene/feature selection (Extval)

Li *et al.* (2004) introduced gene selection based on the comparison of the maximum likelihood of a logistic regression model applied to the original data and permutation datasets. To avoid using computational intensive procedures they propose to take advantage of the extreme-value distribution for the log likelihood ratios. From there a ranking of the genes follows, that can be used to select a predefined number of genes with the highest ranks (extval). Alternatively, Li *et al.* (2004) also suggest two criteria to determine the number of genes to be selected from the ranking list. One is based on the expected values (**E**-criterion) and the other one is based on p-values (**P**-criterion). We applied both criteria. A self-written code was used to implement the method in R.

Between-within ratio (BW)

The between-within (BW) ratio was used for ranking and selection of genes in a microarray context by, e.g., Dudoit *et al.* (2002). The BW ratio is the ratio of the between-treatment sum of squares and the within-treatment sum of squares of gene-expression values. In a two group setting it reduces to the same statistic as the

t-test. A self-written code was used to implement the method in R.

Prediction strength (PS)

The prediction strength (PS) (Xiong *et al.*, 2001) of a certain gene is defined as the ratio of the difference in mean log expression level between the two groups and the sum of the variances of the two classes. A self-written code was used to implement the method in R.

Normal mixture (Mix)

The distribution of intensity measures for an individual feature (peptide) is assumed to come from a mixture of two normal populations with a common variance. Assuming the true class of the spectrum is unknown, z_{ij} is an indicator variable that equals 1 or 0 if spectrum i is obtained for a case or control sample, respectively, given feature j. Let F'_{ij} denote the standardized intensity of feature j in spectrum i. The normal mixture model can be formulated as

$$F'_{ij} \sim z_{ij} N(\mu_{1j}, \sigma_j^2) + (1 - z_{ij}) N(\mu_{0j}, \sigma_j^2). \tag{2}$$

z_{ij} is a latent classification variable assumed to be Bernoulli-distributed with mixing probability π_j (Congdon 2003).

The model was fitted using a Bayesian approach with the following priors:

$$\sigma_j^2 \sim gamma(0.0001, 0.0001),\ \mu_{kj} \sim N(0, \sigma_\mu^2),$$

$$\sigma_\mu^2 \sim gamma(0.0001, 0.0001),\ \pi_j \sim U(0, 1).$$

A spectrum is assigned to the class more frequently represented in the posterior distribution of z_{ij}. This can be seen as corresponding to the choice of the class according to whether the posterior mean of the probability π_j is larger or smaller than 50%.

The ranking of the features is based on misclassification error obtained from comparing, for each feature, the true and predicted classes of the training spectra. The model was fitted in R and WinBugs by using the package `R2WinBugs`.

Statistical impurity measures (Gini)

In contrast to determining a test statistic, we can attempt to find a feature-specific threshold in the intensity range. If a measured value for a particular feature is larger (resp. smaller) than this threshold, the spectrum is assigned to, for instance, class one (resp. two). Statistical impurity measures quantify the effectiveness of this method. There are several ways this can be done, leading to multiple impurity measures. We focus on the Gini index (Gini). A full description can be found in Murthy *et al.* (1994) and Su *et al.* (2003). A self-written code was used to implement the method in R.

2.2.2 Classification methods

The class prediction procedures investigated in our study included classical discriminant analysis techniques, tree methods, and machine learning methods.

Discriminant analysis

Linear discriminant analysis (LDA), a classical discriminant method, estimates linear discriminant functions for decision boundaries based on assumptions of Gaussian distribution and equal covariance matrices for the grouped data. Diagonal linear discriminant analysis is a variant of the LDA method. It assumes a diagonal structure for the covariance matrix. If the matrices are assumed equal for the considered classes, a linear discriminant rule is obtained (DLDA). Otherwise, one obtains a quadratic discriminant rule (DQDA). In a sense, DLDA and DQDA ignore the correlation structure between variables (features). In our study we included LDA and DLDA based on their performance in a microarray context (Van Sanden *et al.* 2007). LDA is implemented as the function *lda* in the `MASS` package, while *stat.diag.da* from the `sma` package is used for DLDA.

Classification tree

A classification tree is a binary recursive partitioning method developed by Breiman *et al.* (1984). In each step a subset of training samples is split in two, based on the intensity value of one particular feature. The value is chosen to obtain an as homogeneous set of labels as possible in each partitioning. The subsets remaining at the final stage are assigned to a certain class, the one which is most frequently represented in the subset. In a way, the method has its own feature selection procedure. It determines which feature to use (from the given set) at each splitting node in order to get the best classification. This feature makes them quite robust to the presence of classification noise.

Aggregated classifiers combine tree classifiers to improve the accuracy of the class prediction. One such method is called *bagging* (Breiman, 1996). Bootstrap replicates (in our case 100) are taken from the training dataset. A tree is constructed for each replicate and the final classification is determined by majority vote. That is, the sample is assumed to belong to the class to which it is most frequently assigned by the different trees. Bagging is said to be a variance reduction technique, designed to stabilize trees. It is implemented by the function *ipredbagg.factor* from the `ipred` package.

Boosting, proposed by Schapire and Freund (1999) is another form of aggregating classifiers. A series of classification trees is produced for the training dataset, each time with different weights assigned to the samples. The idea is to give samples misclassified in the previous step more weight in the current one. The final outcome is a weighted majority vote of all created trees. It is believed that bag-

ging is much better than boosting in situations with substantial classification noise. Boosting is however expected to reduce both the variance and bias of unstable trees. It is implemented as the functions *gbm* and *gbm.more* in the `gbm` package. The *gbm* function is first applied to the data in order to create 100 trees in the manner described above. When applying the function, a shrinkage parameter of 0.001, the fraction of randomly selected observations for building a tree of 0.5, and Bernoulli distribution were used. The *gbm.more* function is used to create 1000 additional trees.

Random forests (Breiman, 2001) are formed by a combination of tree predictors. Subsets of spectra and peptides are obtained by independently drawing samples with replacement from the training dataset and by selecting a number of features at random. A classification tree is estimated for each of the newly formed datasets. A new spectrum is allocated to the class with the most votes over all the trees in the forest. The method is implemented as the function *randomForest* from the `randomForest` package. The number of samples drawn at random is set at 63% of the total number. This is the default value in R. For the features it is determined by a function specified in the help file of *randomForest*. The method was applied with the number of trees equal to 500 and 1000. Results obtained for 1000 trees were very close to those obtained for 500. Therefore, only the latter are reported.

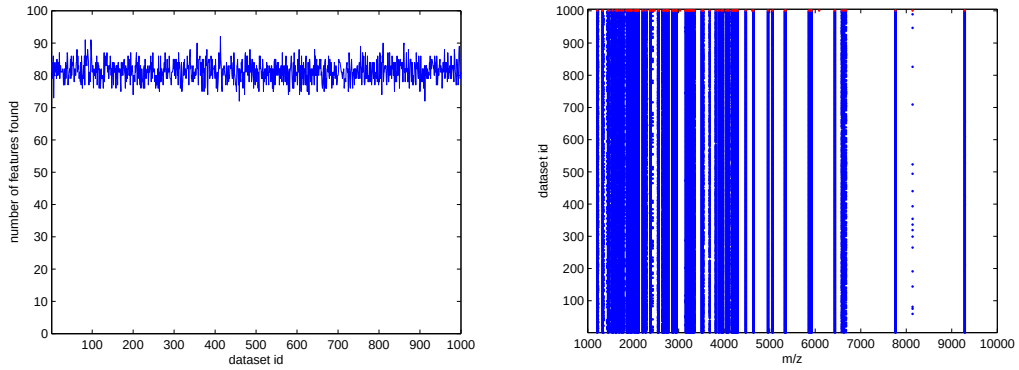
Machine learning

Support vector machines (SVM), first introduced by Cortes and Vapnik (1995) in the machine learning theory, are used to solve two-group classification problems. The idea behind them is the following: the samples from the calibration data are non-linearly mapped to a very high-dimensional feature space. In this space a hyperplane is designed that provides an optimal separation between the two groups. The support vectors are the samples which lie closest to the separating hyperplane. In the input space this hyperplane corresponds to a non-linear decision boundary.

To classify a sample a decision value is calculated. This value quantifies the distance between a sample and the decision boundary. The sign of the decision value determines the class label. Furey *et al.* (2000) give an overview of the calculations involved. Note that SVM does not provide probabilities for assigning individual observations to classes.

SVM are characterized by the regularization parameter and the use of linear, polynomial, radial, splines, and other kernels to solve the optimization problem. For our analysis only the linear and radial kernel were considered. The regularization parameter was set equal to one. The other parameters were set at the default value of the R-function (shrinking was allowed, epsilon=0.1, tolerance=0.001). The method is implemented as the *SVM* function in the package `e1071`.

2.2.3 Cross-validation study



(a) (b)

Figure 6: Features found in re-sampled datasets. Panel a) shows the number of features (clusters) found for the 1000 resampling exercises. Panel b) indicates the location of the features found from the resampling exercise by blue dots. The red dots are the features found for the whole dataset (153 spectra)

In order to choose the best combination of a feature-selection method and a classification procedure, we applied all the combinations to 1000 re-sampled data sets, and we evaluated the misclassification error rate.

Each re-sampled dataset contained 50 spectra from the case-group and 50 spectra from the control-group, randomly selected (without replacement) from the complete set of 153 spectra. For these 100 spectra, treated as a training set, the preprocessing steps described in Section 2.1 were applied.

Figure 6 shows the number and location of clusters found in each of the re-sampled datasets. On average, 82 features were selected (Figure 6, panel (a)). The heatmap in panel (b) shows that in the majority of cases, the same clusters (blue dots) were consistently found in the 1000 re-sampled datasets. Note that, for the whole set of 153 spectra, 92 features were selected. Their mean m/z location is indicated by red dots at the top of panel (b) of Figure 6.

Once the features were selected for the set of 100 spectra by the clustering algorithm, corresponding intensity values were obtained for the 53 remaining spectra, that were treated as the test set. (Note that the testing spectra were also preprocessed; for the intensity standardization, the term in the numerator of (1) was based only on $K = 100$ training spectra.) The misclassification error for each

classification procedure was computed using the test set.

3 Results

In this section we present the results of the cross-validation study undertaken to select the classification procedure, and the results of the application of the chosen procedure to the calibration dataset provided by the organizers of the competition.

3.1 Cross-validation study

The estimates of total misclassification error obtained for the 280 combinations of feature-subset selection and classification methods by applying them to the 1000 re-sampled datasets are displayed in Table 1. For each method the mean misclassification rate and its standard deviation (on parentheses) were calculated.

Instead of using the pre-determined number p of "best" scoring feature, one could use the E- or P-criterion to select it automatically (see Section 2.2.1). Table 2 presents the results obtained using the two criteria for all classification procedures listed in Table 1.

It is worth noting that for the vast majority of the considered approaches the misclassification error decreases with the increasing number of selected features, and it achieves its minimum (except for LDA and DLDA) if all $p = 92$ features are selected for building a classification rule. Based on a simulation study in a microarray context, Van Sanden *et al.* (2007) reported that, for the methods considered in our study, there was a clear dependence of the average misclassification rate on p, the number of selected features. Using too few or too many features increases the misclassification. This is most likely due to the fact that using too few features does not allow to discriminate between the classes, even if the features are truly differentiating. On the other hand, as p increases, more and more non-differentiating features enter the subset selected for building the classification rule and make the classification more difficult. Hence, there might be an optimal value of p that might lead to the best performance of a classification procedure.

From this point of view, Tables 1 and 2 suggest that the optimum choice is the selection of all the 92 features. In this case, the choice of the method of feature-subset selection is irrelevant. It is worth noting, though, that in general, different methods of feature-subset selection give similar results for each classification procedure.

Furthermore, when all $p = 92$ features are considered for building a classification rule, the minimum misclassification error is obtained for the linear kernel

SVM. This approach was therefore selected to be applied to the test data provided by the organizers of the competition.

In the case of the linear kernel SVM, the use of PAM for a subset of 50 features might have been also considered. However, given the aforementioned argumentation for choosing all the features, confirmed by a slightly lower misclassification error, we decided not to use PAM on 50 features.

Table 1: Mean misclassification error rate (standard deviation in parentheses) for the method selection. Abbreviations: RF – random forest; LDA – linear discriminant analysis; DLDA – diagonal LDA; SVMlin – support vector machine with a linear kernel; SVMrad – SVM with a radial kernel.

Classification procedure	Selection criterion	Number of selected features p				
		10	20	30	50	all
Bagging	wilc	.2357 (.0511)	.2147 (.0512)	.2116 (.0510)	.2041 (.0508)	.1952 (.0528)
Bagging	sam	.2385 (.0511)	.2135 (.0517)	.2049 (.0510)	.1925 (.0524)	.1952 (.0528)
Bagging	pam	.2352 (.0541)	.2127 (.0523)	.2022 (.0516)	.1933 (.0517)	.1952 (.0528)
Bagging	ps	.2383 (.0510)	.2122 (.0538)	.2073 (.0516)	.1960 (.0525)	.1952 (.0528)
Bagging	bw	.2389 (.0513)	.2165 (.0543)	.2087 (.0522)	.1952 (.0518)	.1952 (.0528)
Bagging	gini	.2346 (.0535)	.2085 (.0523)	.1975 (.0531)	.1930 (.0532)	.1952 (.0528)
Bagging	extval	.2364 (.0510)	.2138 (.0536)	.2087 (.0524)	.1960 (.0521)	.1952 (.0528)
Bagging	mix	.2204 (.0553)	.2088 (.0550)	.2049 (.0558)	.1970 (.0538)	.1952 (.0528)
Boosting	wilc	.2104 (.0490)	.2062 (.0499)	.2037 (.0497)	.2005 (.0508)	.1994 (.0512)
Boosting	sam	.2152 (.0494)	.2049 (.0507)	.2013 (.0507)	.1996 (.0514)	.1994 (.0512)
Boosting	pam	.2323 (.0535)	.2232 (.0552)	.2034 (.0516)	.1991 (.0510)	.1994 (.0512)
Boosting	ps	.2109 (.0488)	.2050 (.0504)	.2012 (.0513)	.1991 (.0509)	.1994 (.0512)
Boosting	bw	.2118 (.0494)	.2064 (.0506)	.2016 (.0504)	.1991 (.0513)	.1994 (.0512)
Boosting	gini	.2113 (.0497)	.2018 (.0510)	.1993 (.0516)	.1996 (.0515)	.1994 (.0512)
Boosting	extval	.2106 (.0487)	.2052 (.0505)	.2015 (.0511)	.1995 (.0515)	.1994 (.0512)
Boosting	mix	.2321 (.0586)	.2277 (.0595)	.2206 (.0595)	.1980 (.0508)	.1994 (.0512)
RF	wilc	.2211 (.0468)	.1981 (.0476)	.1940 (.0471)	.1877 (.0483)	.1800 (.0477)
RF	sam	.2232 (.0485)	.1990 (.0485)	.1902 (.0468)	.1824 (.0466)	.1800 (.0477)
RF	pam	.2196 (.0521)	.1997 (.0503)	.1899 (.0453)	.1830 (.0464)	.1800 (.0477)
RF	ps	.2219 (.0471)	.1993 (.0490)	.1911 (.0483)	.1830 (.0473)	.1800 (.0477)
RF	bw	.2229 (.0485)	.2025 (.0498)	.1927 (.0491)	.1835 (.0468)	.1800 (.0477)
RF	gini	.2207 (.0508)	.1914 (.0488)	.1833 (.0484)	.1813 (.0473)	.1800 (.0477)
RF	extval	.2204 (.0478)	.1989 (.0495)	.1918 (.0487)	.1830 (.0476)	.1800 (.0477)
RF	mix	.2071 (.0528)	.1938 (.0512)	.1896 (.0523)	.1781 (.0469)	.1800 (.0477)
LDA	wilc	.2068 (.0491)	.2132 (.0528)	.2104 (.0603)	.2074 (.0590)	.3111 (.0709)
LDA	sam	.2067 (.0502)	.1980 (.0519)	.1893 (.0551)	.1957 (.0572)	.3111 (.0709)
LDA	pam	.1879 (.0468)	.1886 (.0531)	.1856 (.0521)	.1948 (.0573)	.3111 (.0709)
LDA	ps	.2071 (.0497)	.1992 (.0534)	.1964 (.0555)	.2020 (.0578)	.3111 (.0709)
LDA	bw	.2107 (.0499)	.2020 (.0537)	.1973 (.0555)	.2019 (.0573)	.3111 (.0709)
LDA	gini	.2068 (.0508)	.1966 (.0537)	.1783 (.0568)	.1953 (.0562)	.3111 (.0709)
LDA	extval	.2065 (.0492)	.2024 (.0538)	.1959 (.0563)	.2029 (.0578)	.3111 (.0709)
LDA	mix	.2083 (.0573)	.2022 (.0565)	.1968 (.0550)	.2121 (.0590)	.3111 (.0709)
DLDA	wilc	.1967 (.0512)	.2016 (.0532)	.2083 (.0543)	.2033 (.0507)	.2046 (.0512)
DLDA	sam	.2045 (.0508)	.2008 (.0520)	.1964 (.0516)	.2005 (.0500)	.2046 (.0512)
DLDA	pam	.2134 (.0513)	.1976 (.0529)	.1945 (.0511)	.2000 (.0498)	.2046 (.0512)
DLDA	ps	.1942 (.0505)	.2011 (.0539)	.2035 (.0537)	.2009 (.0509)	.2046 (.0512)
DLDA	bw	.1995 (.0520)	.2044 (.0546)	.2049 (.0537)	.2010 (.0509)	.2046 (.0512)
DLDA	gini	.2055 (.0536)	.1927 (.0539)	.1971 (.0542)	.2007 (.0514)	.2046 (.0512)
DLDA	extval	.1922 (.0502)	.2013 (.0536)	.2040 (.0536)	.2012 (.0511)	.2046 (.0512)
DLDA	mix	.2308 (.0587)	.2253 (.0575)	.2149 (.0588)	.2123 (.0548)	.2046 (.0512)
SVMlin	wilc	.2048 (.0470)	.1921 (.0521)	.1776 (.0517)	.1748 (.0518)	.1540 (.0492)
SVMlin	sam	.2055 (.0480)	.1810 (.0509)	.1693 (.0502)	.1583 (.0488)	.1540 (.0492)
SVMlin	pam	.1977 (.0476)	.1728 (.0496)	.1698 (.0494)	**.1561 (.0484)**	**.1540 (.0492)**
SVMlin	ps	.2041 (.0474)	.1862 (.0501)	.1772 (.0495)	.1615 (.0499)	.1540 (.0492)
SVMlin	bw	.2052 (.0477)	.1893 (.0514)	.1780 (.0508)	.1626 (.0508)	.1540 (.0492)
SVMlin	gini	.2065 (.0485)	.1831 (.0525)	.1581 (.0506)	.1599 (.0494)	.1540 (.0492)
SVMlin	extval	.2041 (.0476)	.1859 (.0503)	.1762 (.0494)	.1609 (.0506)	.1540 (.0492)
SVMlin	mix	.2040 (.0550)	.1880 (.0551)	.1786 (.0542)	.1788 (.0525)	.1540 (.0492)
SVMrad	wilc	.2078 (.0495)	.1976 (.0500)	.1895 (.0511)	.1795 (.0491)	.1709 (.0464)
SVMrad	sam	.2124 (.0508)	.1871 (.0500)	.1767 (.0485)	.1715 (.0456)	.1709 (.0464)
SVMrad	pam	.2114 (.0515)	.1862 (.0492)	.1725 (.0453)	.1699 (.0458)	.1709 (.0464)
SVMrad	ps	.2076 (.0492)	.1894 (.0520)	.1819 (.0497)	.1721 (.0467)	.1709 (.0464)
SVMrad	bw	.2096 (.0502)	.1937 (.0530)	.1841 (.0506)	.1726 (.0465)	.1709 (.0464)
SVMrad	gini	.2130 (.0524)	.1801 (.0526)	.1680 (.0494)	.1695 (.0451)	.1709 (.0464)
SVMrad	extval	.2054 (.0489)	.1903 (.0521)	.1834 (.0511)	.1727 (.0467)	.1709 (.0464)
SVMrad	mix	.2115 (.0552)	.1933 (.0520)	.1767 (.0498)	.1678 (.0459)	.1709 (.0464)

Table 2: Mean misclassification error rate (standard deviation in parentheses) for the **E**- and **P**-criteria of the extreme-value-distribution-based gene/feature selection. Abbreviations as in Table 1.

Classification procedure	E-criterion	P-criterion
Bagging	0.2016 (0.0506)	0.2224 (0.0498)
Boosting	0.2008 (0.0506)	0.2097 (0.0485)
RF	0.1896 (0.0468)	0.2102 (0.0458)
LDA	0.1933 (0.0567)	0.2037 (0.0511)
DLDA	0.2039 (0.0529)	0.2000 (0.0536)
SVMlin	0.1720 (0.0498)	0.1960 (0.0478)
SVMrad	0.1780 (0.0475)	0.1995 (0.0495)

3.2 Training data and leave-one-out cross-validation

The linear kernel SVM was applied to the pre-processed training dataset of 153 spectra. Forty-eight spectra (24 of each class) where selected as support vectors. When the classifier was then applied to the whole set of 153 spectra of the calibration data, a perfect classification was reached. The total error rate was therefore 0 and both the sensitivity and the specificity of the classifier were estimated at 100%.

The error rate was also estimated using leave-one-out cross-validation. That is, each spectrum was removed from the dataset, the classification method was applied to the remaining spectra, and the class prediction was obtained for the removed spectrum. Results are provided in Table 3. The total error rate was estimated to equal $24/153 = 0.1569$, with the sensitivity and specificity equal to $67/77 = 0.8701$ and $62/76 = 0.8158$, respectively. Note that the estimated error rate is very close to the estimate reported in Table 1.

Table 3: Classification results for leave-one-out cross-validation.

Predicted class	True class	
	0	1
0	67	14
1	10	62

4 Discussion

As mentioned in the Introduction, classification of samples using mass spectra requires a careful consideration of various sources of nuisance and error. For instance, a removal of systematic effects like, e.g., a varying baseline on the intensity scale, or miscalibration of mass-to-charge coordinates, needs to be performed. An important step is also a selection of features of the spectra that are likely due to true biological signal (peptides). To this aim, chemical-knowledge-based peak finding methods might be used (Valkenborg *et al.*, 2007; Valkenborg *et al.*, 2008). In this way, the selection of noise-generated spectrum features for building a classification rule might be avoided.

We believe that a careful pre-processing of mass spectra is a key to developing a successful classification procedure. From this point of view, one should start from raw data and apply the methods of choice aimed at removal of the various nuisance effects and additional error processes present within the spectral data. Note that

some methods (e.g., non-linear removal of baseline) make it impossible to retrieve raw data from the pre-processed ones (Baggerly *et al.*, 2004). This was the case for the data made available to the participants of the competition. It would be of interest to investigate whether using the raw data might improve the reported results.

Another important issue is the choice of the classification procedure. This issue also applies to other complex experimental technique as, e.g., microarrays. Given the complexity of data produced by such techniques, one probably should not expect that a single classification method will always outperform all the others. It is therefore paramount to investigate relative merits of different procedures to see which methods, and in which settings, might be expected to work reasonably well. In the current study we attempted to achieve it by estimating the misclassification error rate for the considered classification approaches by applying the approaches to 1000 re-sampled datasets. The results were pointing favorably towards linear kernel SVM, with radial kernel SVM and RF as the next best alternatives. Interestingly, for a microarray context, Van Sanden *et al.* (2007) reported DLDA, RF, and radial SVM as the best performing approaches. Good performance of DLDA and RF in the microarray context was also reported by, e.g., Dudoit *et al.* (2002) and Lee *et al.* (2005). It would be of interest to check whether these slightly different conclusions can be confirmed and related to the different nature of the mass spectrometry and microarray data.

Acknowledgment

Financial support from the IAP research network nr P6/03 of the Belgian government (Belgian Science Policy) is gratefully acknowledged by all the authors. The first author also gratefully acknowledges support from Bijzonder Onderzoeksfonds Universiteit Hasselt (grant BOF04G01).

We are grateful to the reviewers for their insightful comments, which resulted in an improved manuscript.

References

K.A. Baggerly, J.S. Morris, and K.R. Coombes. Reproducibility of seldi-tof protein patterns in serum: comparing data sets from different experiments. *Bioinformatics*, 20:777785, 2004.

K.A. Baggerly, J.S. Morris, S.R. Edmonson, and K.R. Coombes. Signal in noise: evaluating reported reproducibility of serum proteomic tests for ovarian cancer. *Journal of the National Cancer Institute*, 97(4):307309, 2005.

E.J. Breen, F.G. Hopwood, K.L. Williams, and M.R. Wilkins. Automatic poisson peak harvesting for high throughput protein identification. *Electrophoresis*, 21:2243–2251, 2000.

L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, 1996.

L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. New York: Chapman & Hall, 1984.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 10:273–297, 1995.

S. Dudoit, J. Fridlyand, and T.P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 98:77–87, 2002.

Y. Freund and R.E. Schapire. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14:771–780, 1999.

T. S. Furey, N. Christianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Hauessler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

Lee J.B. Lee, J.W. and, M. Park, and S.H. Song. Extensive comparison of recent classification tools applied to microarray data. *Computational Statistics and Data Analysis*, 48:869–885, 2005.

W. Li, F. Sun, and I. Grosse. Extreme value distribution based gene selection criteria for discriminant microarray data analysis using logistic regression. *Journal of Computational Biology*, 11(2/3):215–226, 2004.

S.K. Murthy, S. Kasif, and S. Salzberg. A system for iinduction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–33, 1994.

E.F. Petricoin III, A.I. Ardekani, M. Ali, and B.A. Hitt. Use of proteomic patterns in serum to identify ovarian cancer. *The Lancet*, 359:572–577, 2002.

Y. Su, T.M. Murali, V. Pavlovic, M. Schaffer, and S. Kasif. Rankgene: a program to rank genes from expression data. *Bioinformatics*, 19:1578–1579, 2003.

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

V. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, 98:5116–5121, 2001.

D. Valkenborg, P. Assam, L. Krols, G. Thomas, K. Kas, and T. Burzykowski. Using a poisson approximation to predict the isotopic distribution of sulphur-containing peptides in a peptide-centric proteomic approach. *Rapid Communications in Mass Spectrometry*, 21:3387–3391, 2007.

D. Valkenborg, I. Jansen, and T. Burzykowski. A model-based method for the prediction of the isotopic distribution of peptides. *Journal of the American Society for Mass Spectrometry*, 2008.

S. Van Sanden, D. Lin, and T. Burzykowski. Performance of classification methods in a microarray setting: a simulation study. accepted for publication, 2007.

M. Xiong, W. Li, J. Zhao, L. Jin, and E. Boerwinkle. Feature (gene) selection in gene expression-based tumor classification. *Molecular Genetics and Metabolism*, 73(3):239–247, 2001.