Supplementary information

A comparative investigation of modern feature selection and classification approaches for the analysis of mass spectrometry data

Piotr S. Gromski,[a] Yun Xu,[a] Elon Correa,[a] David I. Ellis,[a] Michael L. Turner[b] and Royston Goodacre[a]

[a] School of Chemistry, Manchester Institute of Biotechnology, The University of Manchester, 131 Princess Street , Manchester, M1 7DN, UK.

[b] School of Chemistry, Brunswick Street, The University of Manchester, Manchester, M13 9PL, UK.

Correspondence to Prof Roy Goodacre: roy.goodacre@manchester.ac.uk

Tel: +44 (0) 161 306-4480

Additional information on the algorithms employed in this study

1. Bootstrapping with replacement

Bootstrapping is a statistical approach based on building a sampling distribution model by multiple resampling from the experimental data set. In this approach observations are selected with the same probability (i.e., randomly) from a *data set* and these are assigned to the *training set*. The selected sample is then replaced into the original *data set*. The observations that have not been assigned to the *training set* are used to build the *test set* as shown in Fig. S1. The procedure is repeated *n* times (hundreds or thousands) depending on the dimension of the data that have to be analysed, and thus many training and test sets are constructed. This method allows for the objective assessment of the prediction accuracy accurately without the necessity to perform additional data collection [1, 2]. All metrics of prediction accuracy are conducted on the *test sets only*.

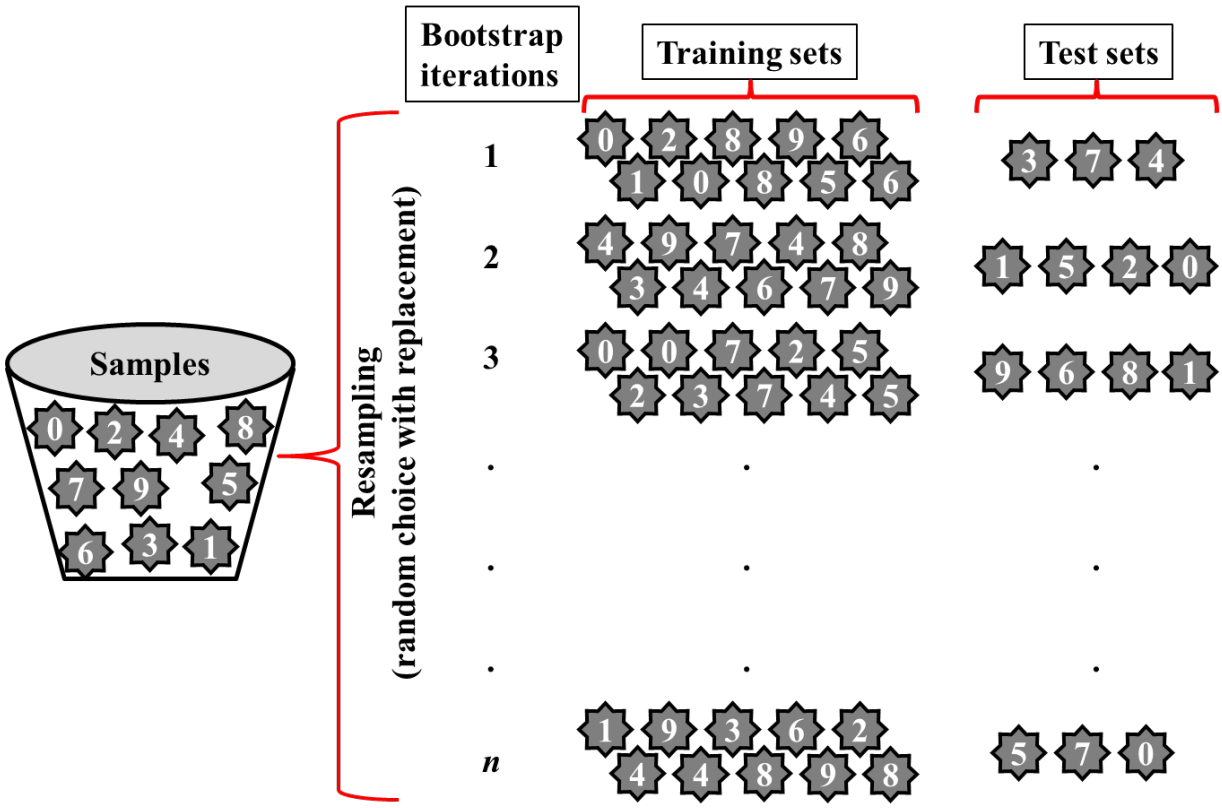

Fig. S1 **Bootstrapping with replacement.** This figure mimics the process of resampling (random selection with replacement) for *n* iterations, which in this example uses a sample data set of size 10. This process randomly selected 10 samples from the original data set (the 'bucket' on the LHS) and these are allocated to the training set, after each selection the sample is replaced back into the 'bucket'. Any unallocated samples are used in the test set. On average the training sets comprise 63.2% of all samples and the test set the remaining 36.8% (on average) of all samples from original data set [1, 2].

2. Random forests

Random forests (RF) is a technique that uses many parse trees (in our study this is set to 500) that represent mathematical operations to transform the input data (Py-MS spectra) into a desired output (e.g., spore coded as '1' versus vegetative cell coded as '0'). Multiple outputs (≥ 2) can be predicted with RF. Unlike CART (classification and regression tree) RF does not pruned these parse trees as the whole population of trees are used for prediction [3]. A particular advantage of RF is that the predicted output depends on only two parameters, the number of variables to be selected for the generation of each tree (*mtry*; in this study $\sqrt{150}$ =12) and the number of trees to be grown within each forest (*ntree*; 500) [4]. In general this provides good predictive ability when the input data are noisy and is useful when the data are short and fat (i.e., more variables than samples). Moreover, this algorithm returns variables ranked in order of importance for classification and thus can be used for feature reduction.

Finally, as a large number of trees are used, this reduces the chances of overfitting the input data and thus help generalisation for 'unseen' data [5, 6].

With R the "randomForest" package was used in this study and produced interactive permutation importance plots. We limited our selection to the top 30 variables selected as this corresponds to 20% of the inputs. For this process both the mean decrease in accuracy and mean decrease in Gini were used; where Gini is a criterion that measures statistical dispersion [4]. These values were averaged over all trees. The resultant plots are displayed in Figure S2 where the variables are ranked in order of importance vertically and the mean decrease in accuracy (S2A) or Gini (S2B) shown on the abscissa [5].

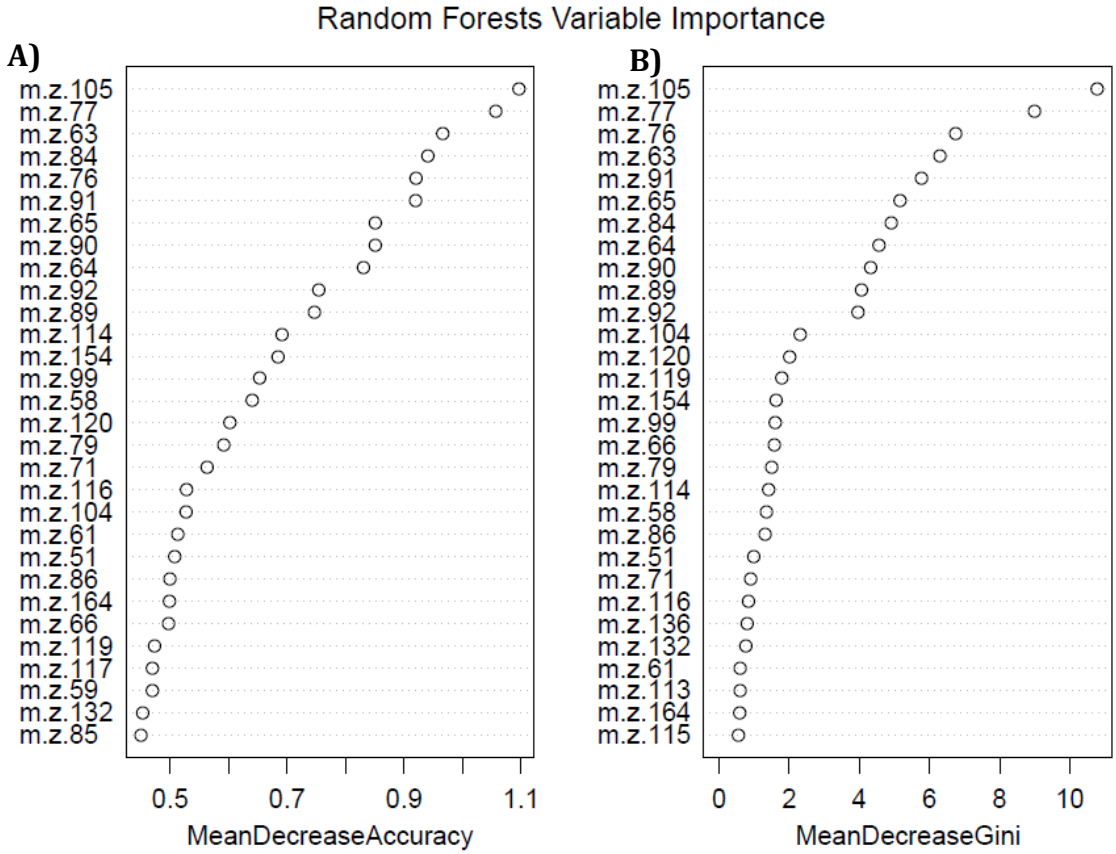

Fig. S2 **RF generated variable importance plots**. This displays the ranked variables according to (A) mean decrease in accuracy and (B) mean decrease in Gini for one case study out of 100 for the training set. Note as this is a typical example of the ranking results this will not be the same as the sum frequency from all 100 iterations which is shown in Fig. 3.

3. Linear discriminant analysis (LDA)

A cartoon of the LDA boundaries is provided in Figure S3. In this plot just two features are plotted and the location of samples from two groups (classes) are shown. A linear decision boundary is computed between them according to Fisher statistics [7, 8]. In this representation the LD scores would separate the two groups in a single dimensional space and the green arrow indicates the projection of the data onto this LD score.

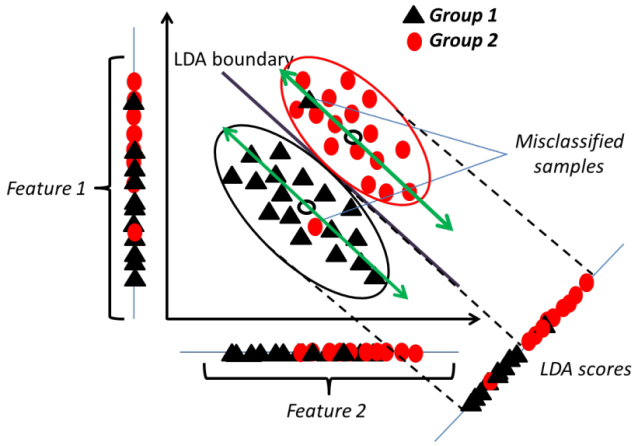

Fig. S3 **Cartoon of LDA.** In this graphic two groups are separated using a single latent variable.

4. Support vector machines (SVM)

A cartoon of the support vectors used in SVM is provided in Figure S4. In this representation two classes (groups) are separated by a single SVM hyperplane (black line). Light blue dash lines represent the optimal margin either side of this hyperplane and the position of this margin is dictated by so called "support vectors". Test set samples are then projected into this space and predicted as belonging to class 1 or class 2 [9, 10].

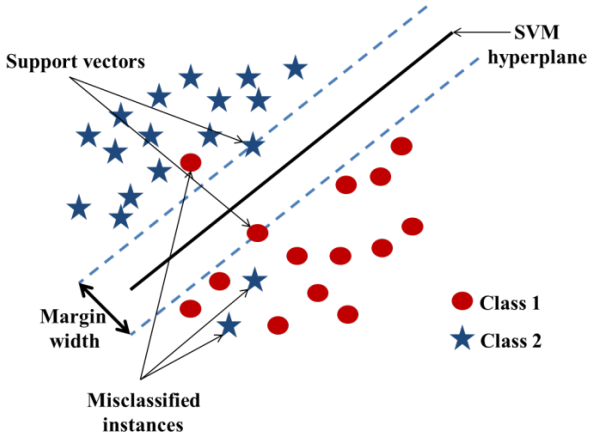

Fig. S4 **Cartoon of SVM with "linear" kernel.** In this graphic two groups are separated using a single latent variable.