Metabolomic Data Analysis with MetaboAnalyst

User ID: guest6522519400069885256

April 14, 2009

1 Data Processing and Normalization

1.1 Reading and Processing the Raw Data

MetaboAnalyst accepts a variety of data types generated in metabolomic studies, including compound concentration data, binned NMR/MS spectra data, NMR/MS peak list data, as well as MS spectra (NetCDF, mzXML, mzDATA). Users need to specify the data types when uploading their data in order for MetaboAnalyst to select the correct algorithm to process them.

The R scripts `datautils.R` and `processing.R` are required to read in and process the uploaded data.

1.1.1 Reading Binned Spectral Data

The binned spectra data should be uploaded in comma seperated values (.csv) format. Samples can be in rows or columns, with class labels immediately following the sample IDs.

The uploaded file is in comma separated values (.csv) format. Samples are in rows and features in columns The uploaded data file contains 50 (samples) by 200 (spectra bins) data matrix.

1.1.2 Filtering Baseline Noises

A significant proportion of bins contain close-to-zero values that come from baseline noises. These values are troublesome for some algorithms to work properly and should be excluded before further data analysis. MetaboAnalyst uses a simple linear filter based on the maximal values of each bin. The default cut-off threshold will remove 25% of the lowest spectra bins.

Please see Figure 1 for a summary graph. The selected cut-off threshold is 0.00186. A total of 51 bins were excluded based on this cut-off.

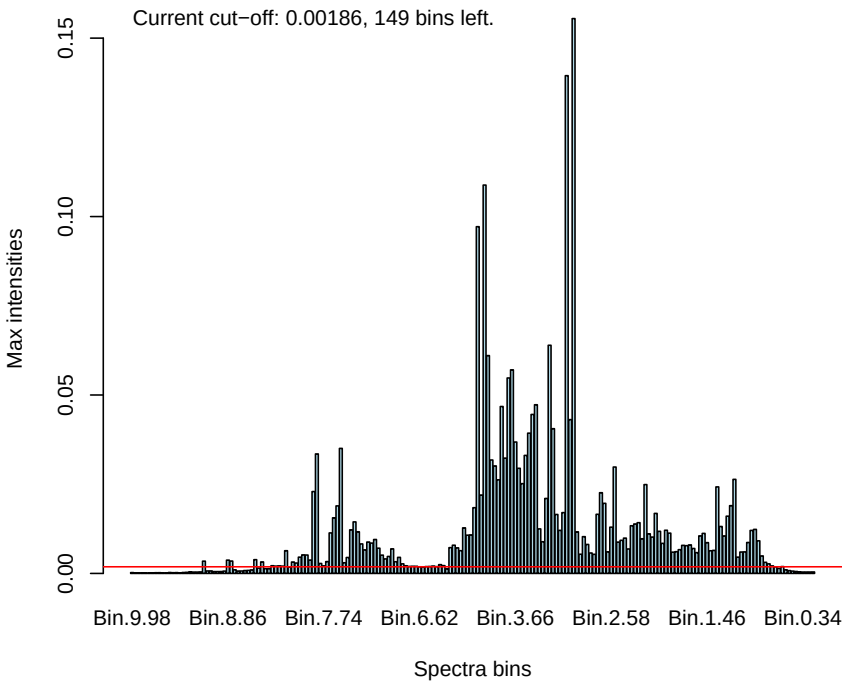

Figure 1: Filter baseline noises for binned spectra. The bars represent the maximum values of each corresponding bin.

1.1.3 Data Integrity Check

Before data analysis, a data integrity check is performed to make sure that all the necessary information has been collected. The class labels must be present and contain only two classes. If samples are paired, the class label must be from -n/2 to -1 for one group, and 1 to n/2 for the other group (n is the sample number and must be an even number). Class labels with same absolute value are assumed to be pairs. Compound concentration or peak intensity values must all be non-negative numbers.

1.1.4 Missing value imputations

Too many zeroes or missing values will cause difficulties for downstream analysis. MetaboAnalyst offers several different methods for this purpose. The default method replaces all the missing and zero values with a small values (the half of the minimum positive values in the original data) assuming to be the detection limit. The assumption of this approach is that most missing values are caused by low abundance metabolites (i.e.below the detection limit). In addition, since zero values may cause problem for data normalization (i.e. log), they are also replaced with this small value. User can also specify other methods, such as replace by mean/median, or use Probabilistic PCA (PPCA), Bayesian PCA (BPCA) method, Singular Value Decomposition (SVD) method to impute the missing values [1]. Please choose the one that is the most appropriate for your data. Table 1 summarizes the result of the data processing steps.

Missing variables were replaced with a small value: 0.00093

[1] Stacklies W, Redestig H, Scholz M, Walther D, Selbig J. *pcaMethods: a bioconductor package, providing PCA methods for incomplete data.*, Bioinformatics 2007 23(9):1164-1167

Table 1: Summary of data processing results

	Features (positive)	Missing/Zero	Features(baseline)	Features (processed)
P002	195	5	51	149
P012	187	13	51	149
P014	200	0	51	149
P027	200	0	51	149
P034	198	2	51	149
P037	187	13	51	149
P038	195	5	51	149
P041	178	22	51	149
P042	198	2	51	149
P049	189	11	51	149
P056	190	10	51	149
P058	179	21	51	149
P060	190	10	51	149
P064	200	0	51	149
P065	198	2	51	149
P070	190	10	51	149
P080	196	4	51	149
P085	200	0	51	149
P086	193	7	51	149
P089	199	1	51	149
P092	191	9	51	149
P099	190	10	51	149
P113	152	48	51	149
P013b	191	9	51	149
P100b	199	1	51	149
C002	194	6	51	149
C004	189	11	51	149
C005	191	9	51	149
C006	195	5	51	149
C007	200	0	51	149
C009	186	14	51	149
C010	196	4	51	149
C011	177	23	51	149
C012	189	11	51	149
C015	188	12	51	149
C016	188	12	51	149
C017	198	2	51	149
C019	181	19	51	149
C020	184	16	51	149
C021	187	13	51	149
C022	191	9	51	149
C024	190	10	51	149
C026	195	5	51	149
C028	196	4	51	149
C029	192	8	51	149
C030	182	18	51	149
C031	179	21	51	149
C032	191	9	51	149
C033	189	11	51	149
C034	199	1	51	149

1.2 Data Normalization

The data is stored as a table with one sample per row and one variable (bin/ peak/metabolite) per column. There are two types of normalization. Row-wise normalization aims to bring each sample (row) comparable to each other (i.e. urine samples with different dilution effects). Column-wise normalization aims to make each variable (column) comparable to each other within the same sample. The procedure is useful when variables are of very different orders of magnitude.

The normalization consists of the following options:

1. Row-wise normalization:
    - Normalization by the sum
    - Normalization by a reference sample (probabilistic quotient normalization) [2]
    - Normalization by a reference feature (i.e. creatinine, internal control)
    - Sample specific normalization (i.e. normalize by dry weight, volume)
2. Column-wise normalization :
    - Log transformation (log 2)
    - Unit scaling (mean-centered and divided by standard deviation of each variable)
    - Pareto scaling (mean-centered and divided by the square root of standard deviation of each variable)
    - Range scaling (mean-centered and divided by the value range of each variable)

The R script `normalization.R` is required. Figure 2 shows the effects before and after normalization.

[2] Dieterle F, Ross A, Schlotterbeck G, Senn H. *Probabilistic quotient normalization as robust method to account for dilution of complex biological mixtures. Application in 1H NMR metabonomics*, 2006, Anal Chem 78 (13);4281 - 4290

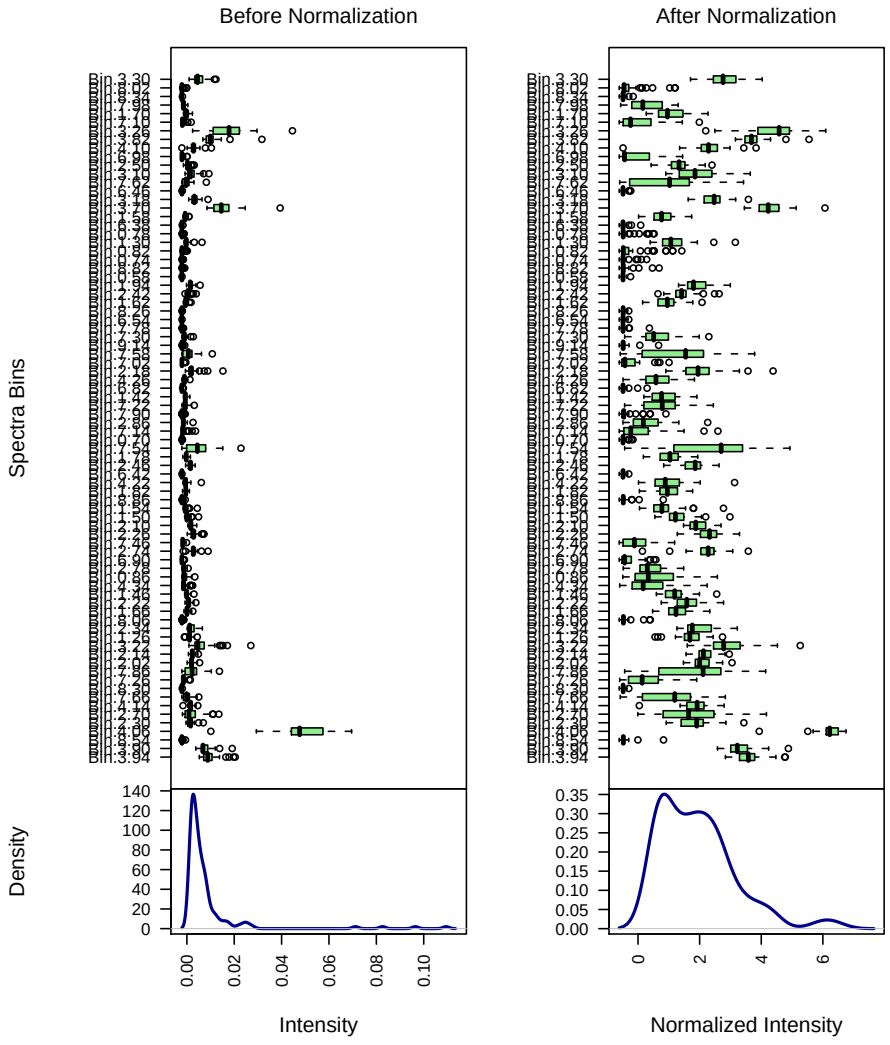

Figure 2: Box plots and kernel density plots before and after normalization. The boxplots show at most 80 features due to space limit. The density plots are based on all samples. Row-wise normalization: Normalization to constant sum Column-wise normalization: Log Normalization.

2 Statistical and Machine Learning Data Analysis

MetaboAnalyst offers a variety of methods commonly used in metabolomic data analyses. They include:

1. Univariate analysis methods:
    - Fold Change Analysis
    - T-tests
    - Volcano Plot
2. Dimensional Reduction methods:
    - Principal Component Analysis (PCA)
    - Partial Least Squares - Discriminant Analysis (PLS-DA)
3. Robust Feature Selection Methods in microarray studies
    - Significance Analysis of Microarray (SAM)
    - Empirical Bayesian Analysis of Microarray (EBAM)
4. Clustering Analysis
    - Hierarchical Clustering
        - Dendrogram
        - Heatmap
    - Partitional Clustering
        - K-means Clustering
        - Self-Organizing Map (SOM)
5. Supervised Classification and Feature Selection methods
    - Random Forest
    - Support Vector Machine (SVM)

2.1 Principal Component Analysis (PCA)

PCA is an unsupervised method aiming to find the directions that best explain the variance in a data set (X) without referring to class labels (Y). The data are summarized into much fewer variables called *scores* which are weighted average of the original variables. The weighting profiles are called *loadings*. The PCA analysis is performed using the `prcomp` package. The calculation is based on singular value decomposition.

The Rscript `chemometrics.R` is required. Figure 3 is pairwise score plots providing an overview of the various seperation patterns among the most significant PCs; Figure 4 is the scree plot showing the variances explained by the selected PCs; Figure 5 shows the 2-D score plot between selected PCs; Figure 6 shows the 3-D score plot between selected PCs; Figure 7 shows the loading plot between the selected PCs; Figure 8 shows the biplot between the selected PCs.

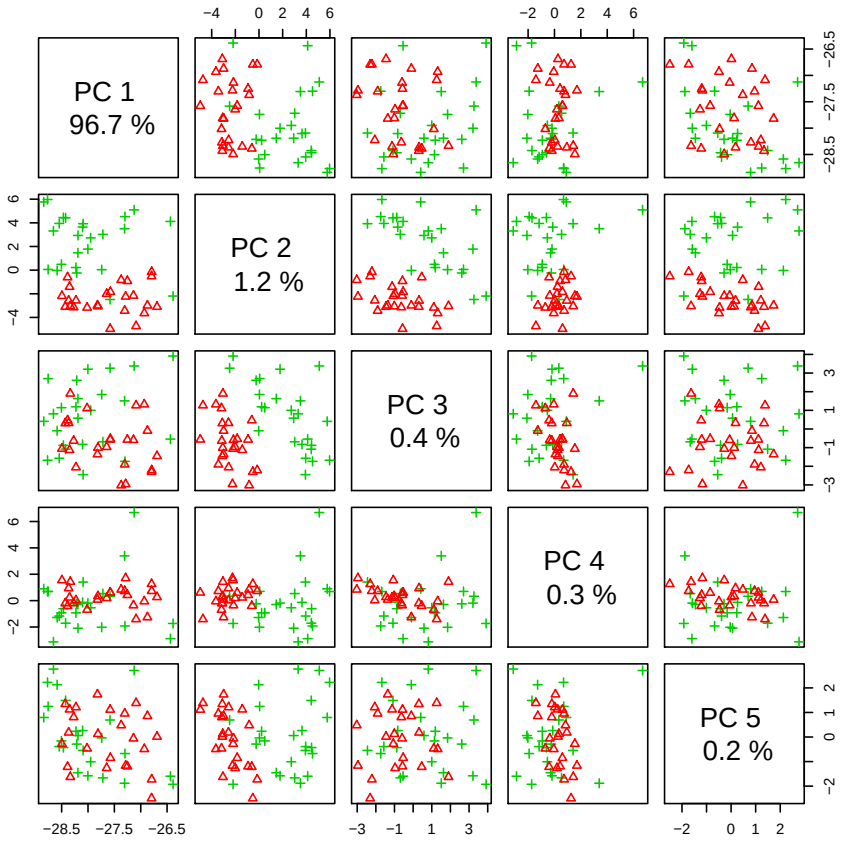

Figure 3: Pairwise score plots between the selected PCs. The explained variance of each PC is shown in the corresponding diagonal cell.

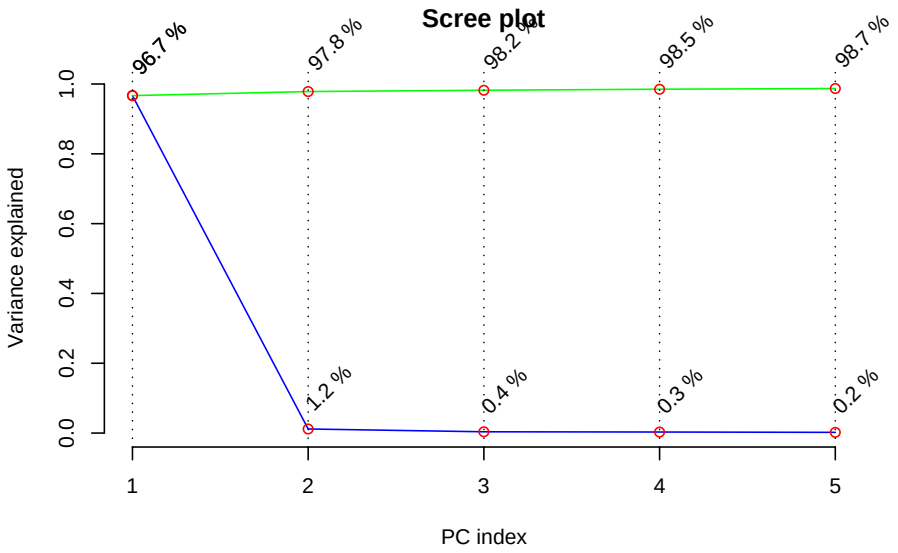

Figure 4: Scree plot shows the variance explained by PCs. The green line on top shows the accumulated variance explained; the blue line underneath shows the variance explained by individual PC.

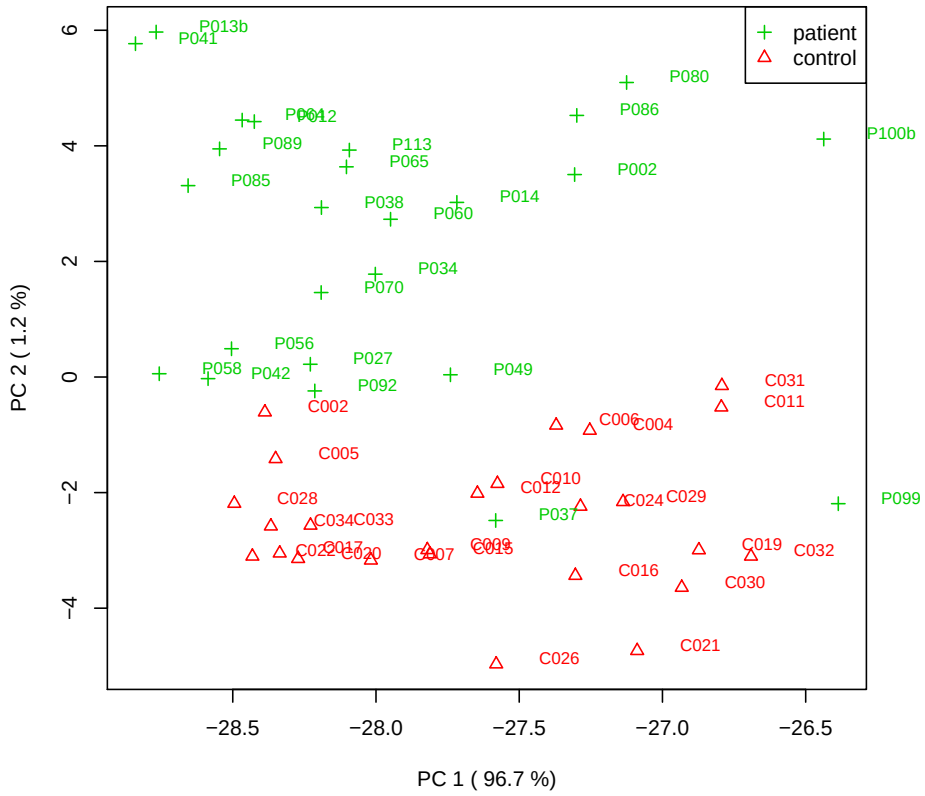

Figure 5: Score plot between the selected PCs. The explained variances are shown in brackets.

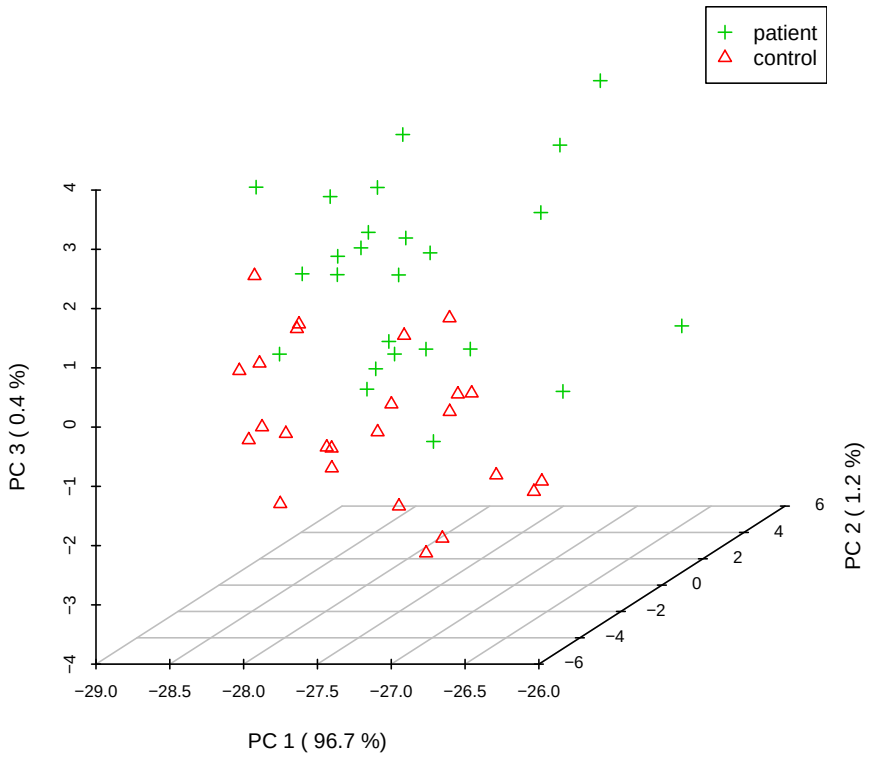

Figure 6: 3D score plot between the selected PCs. The explained variances are shown in brackets.

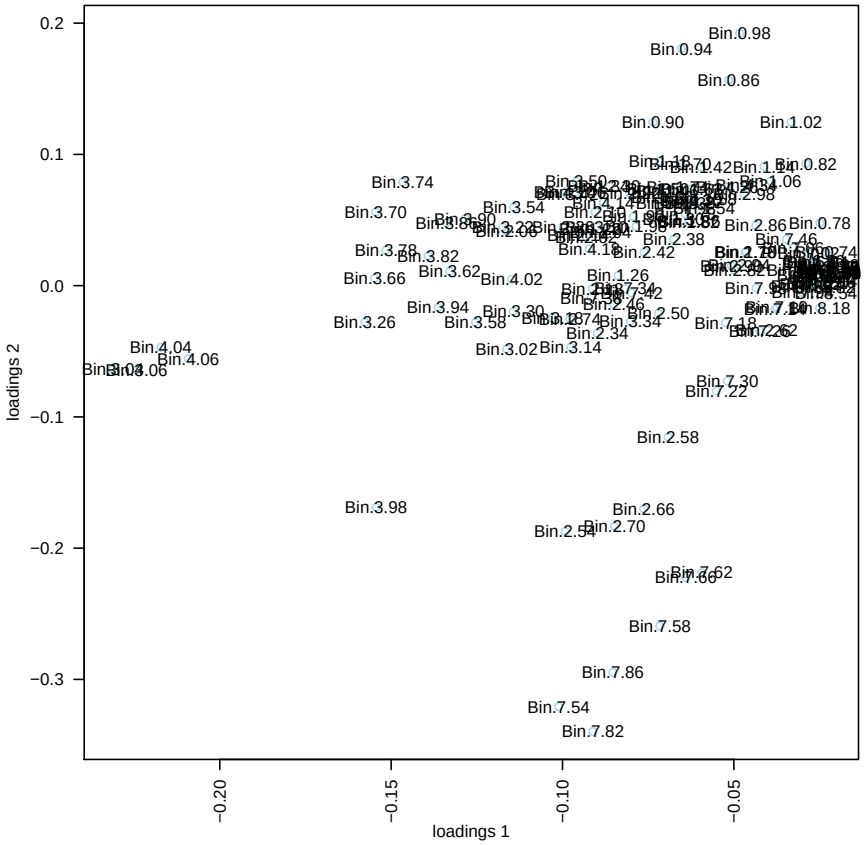

Figure 7: Loading plot for the selected PCs.

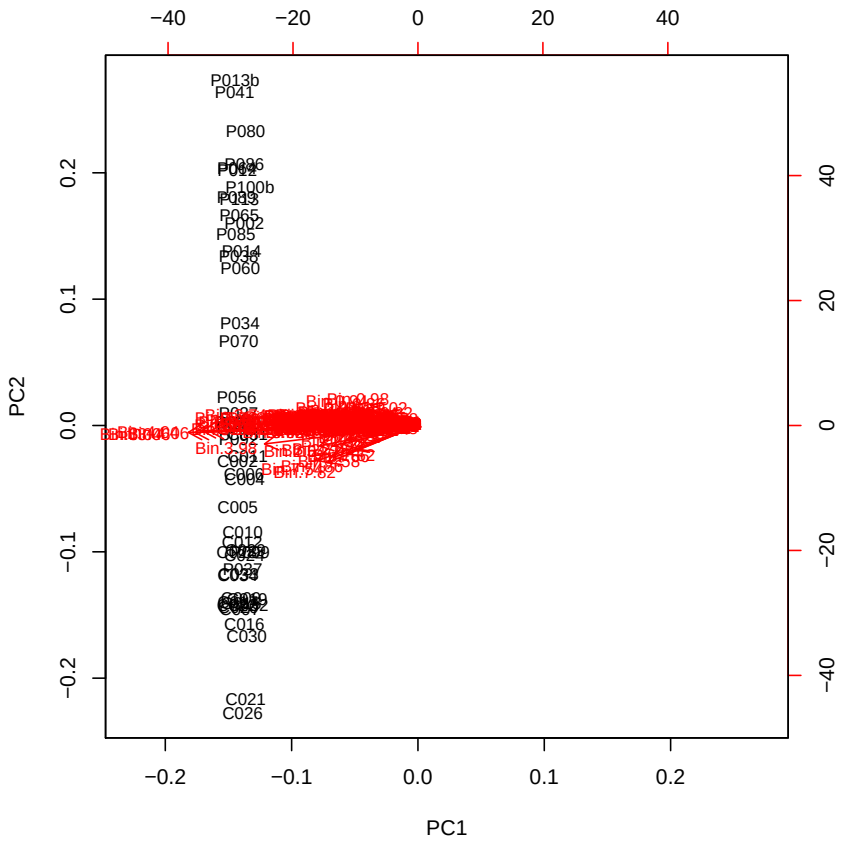

Figure 8: PCA biplot between the selected PCs. Note, you may want to test different centering and scaling normalization methods for the biplot to be displayed properly.

2.2 Partial Least Squares - Discriminant Analysis (PLS-DA)

PLS is a supervised method that uses multivariate regression techniques to extract via linear combination of original variables (X) the information that can predict the class membership (Y). The PLS regression is performed using the `plsr` function provided by R `pls` package[3]. The classification and cross-validation are performed using the corresponding wrapper function offered by the `caret` package[4].

To assess the significance of class discrimination, a permutation test was performed. In each permutation, a PLS-DA model was built between the data (X) and the permuted class labels (Y) using the optimal number of components determined by cross validation for the model based on the original class assignment. The ratio of the between sum of the squares and the within sum of squares (B/W-ratio) for the class assignment prediction of each model was calculated. If the B/W ratio of the original class assignment is a part of the distribution based on the permuted class assignments The contrast between the two class assignment cannot be considered significant from a statistical point of view.

There are two variable importance measures in PLS-DA. The first, Variable Importance in Projection (VIP) is a weighted sum of squares of the PLS loadings taking into account the amount of explained Y-variation in each dimension. The other importance measure is based on the weighted sum of PLS-regression The weights are a function of the reduction of the sums of squares across the number of PLS components. coefficients[5].

The R script `chemometrics.R` is required. Figure 9 shows the overview of score plots; Figure 10 shows the 2-D score plot between selected components; Figure 11 shows the 3-D score plot between selected components; Figure 12 shows the loading plot between the selected components; Figure 13 shows the classification performance with different number of components. Figure 14 shows the important features identified by PLS-DA. Figure 15 shows the permutation test results for model validation.

[3]Ron Wehrens and Bjorn-Helge Mevik.*pls: Partial Least Squares Regression (PLSR) and Principal Component Regression (PCR)*, 2007, R package version 2.1-0

[4]Max Kuhn. Contributions from Jed Wing and Steve Weston and Andre Williams.*caret: Classification and Regression Training*, 2008, R package version 3.45

[5]Bijlsma et al.*Large-Scale Human Metabolomics Studies: A Strategy for Data (Pre-) Processing and Validation*, Anal Chem. 2006, 78 567 - 574

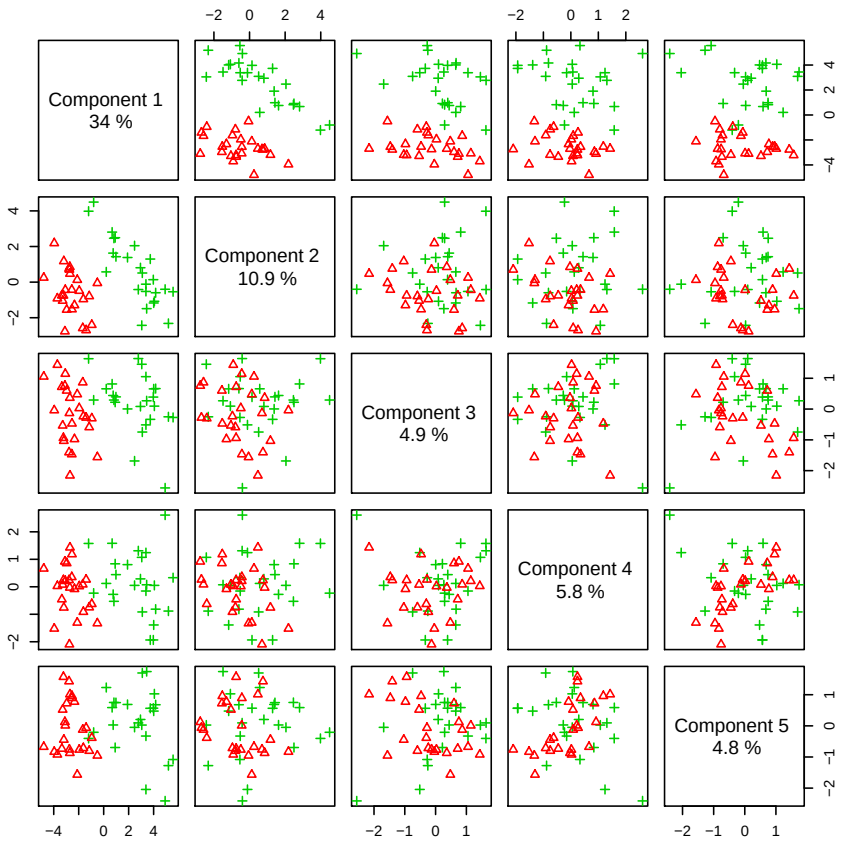

Figure 9: Pairwise score plots between the selected components. The explained variance of each component is shown in the corresponding diagonal cell.

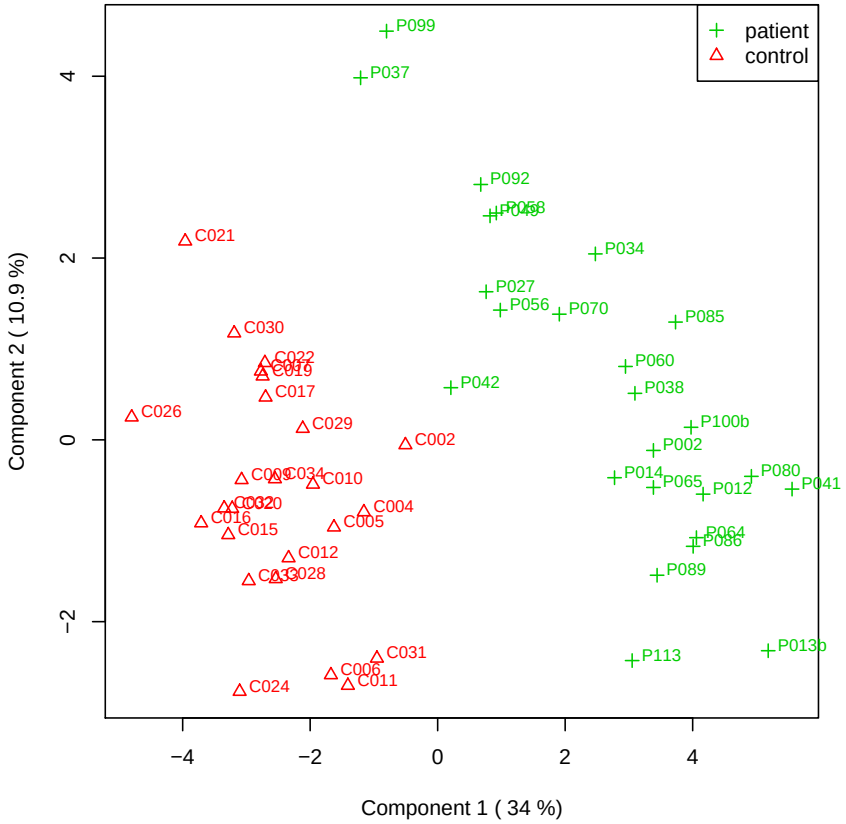

Figure 10: Score plot between the selected PCs. The explained variances are shown in brackets.

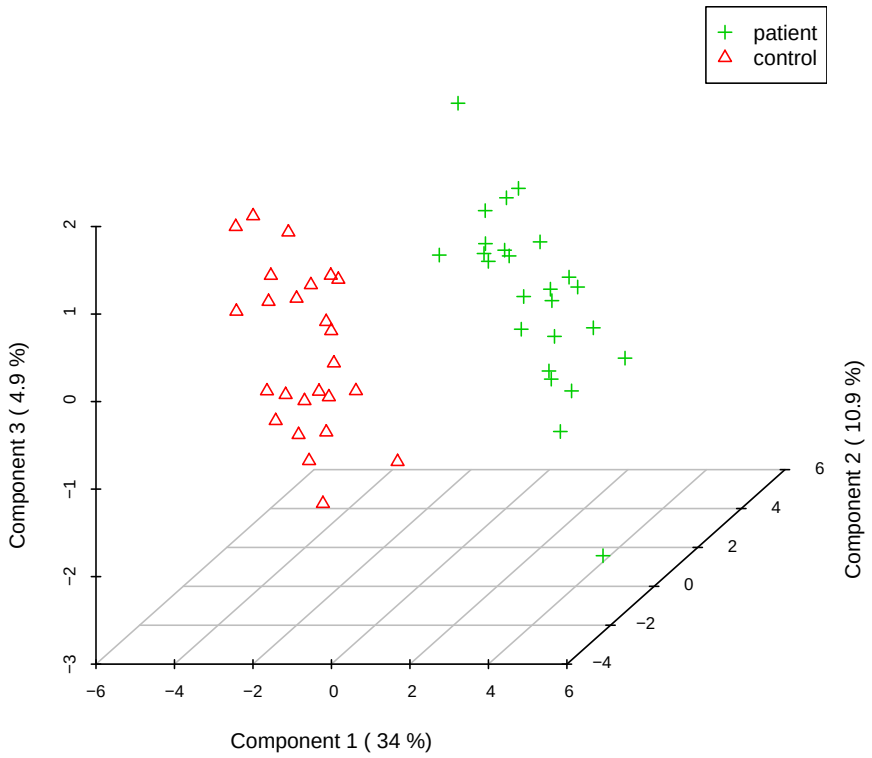

Figure 11: 3D score plot between the selected PCs. The explained variances are shown in brackets.

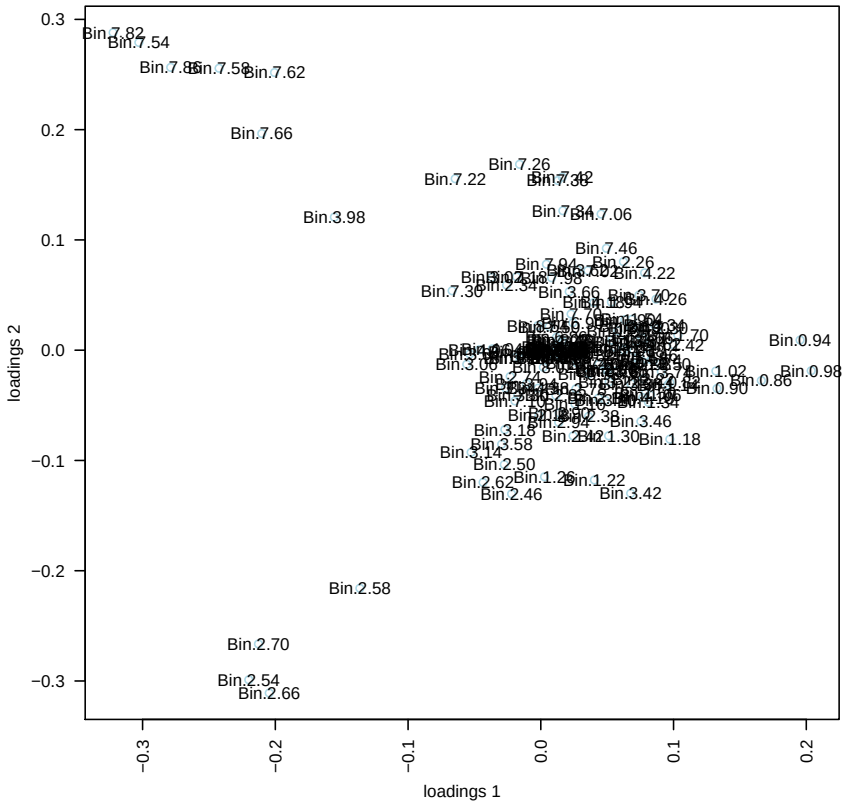

Figure 12: Loading plot between the selected PCs.

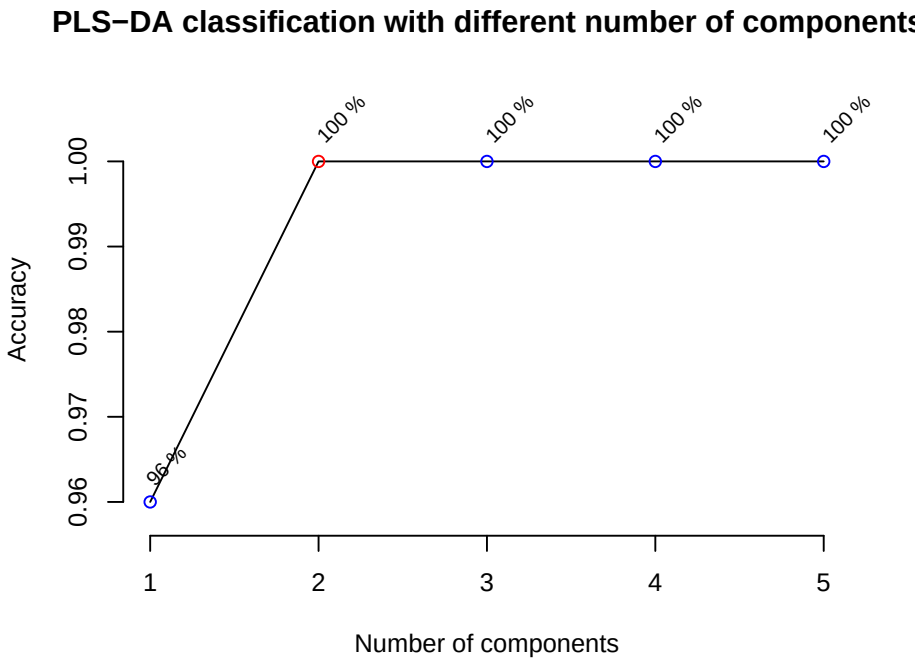

Figure 13: PLS-DA classification using different number of components. The red circle indicates the best classifier.

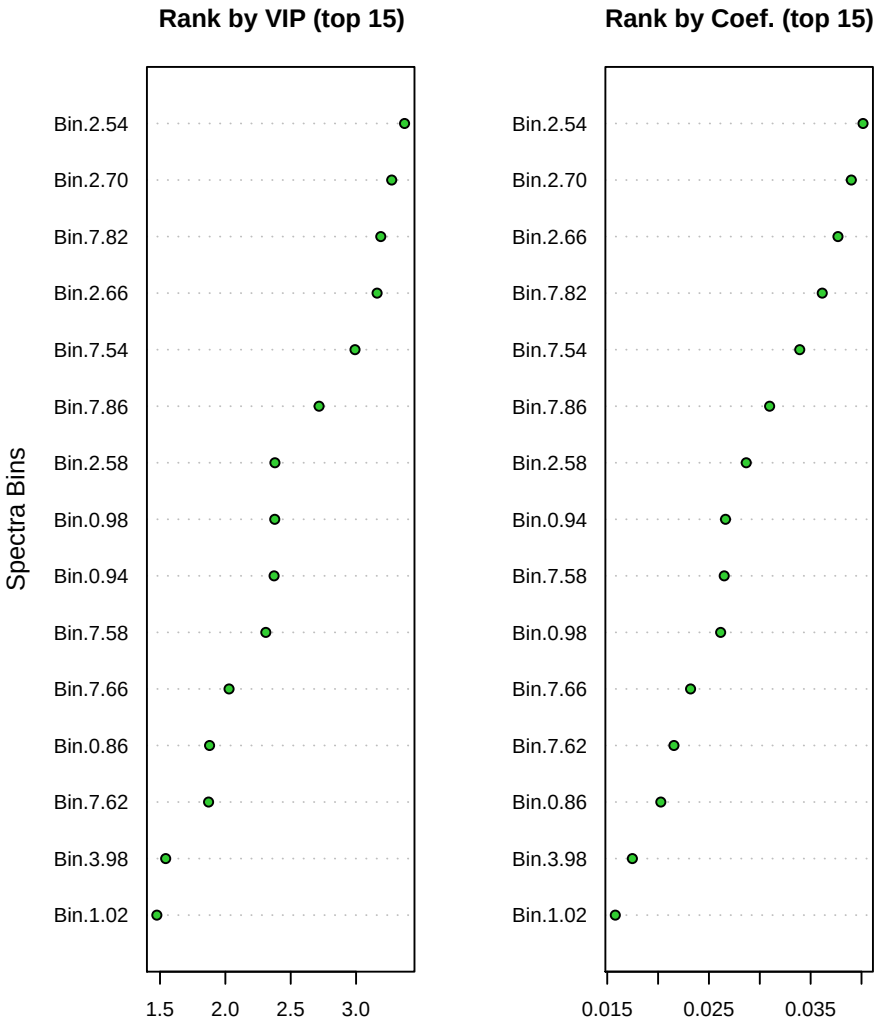

Figure 14: Important features identified by PLS-DA. The left panel shows the features ranked by VIP score. The right panel shows the features ranked based on their regression coefficients.

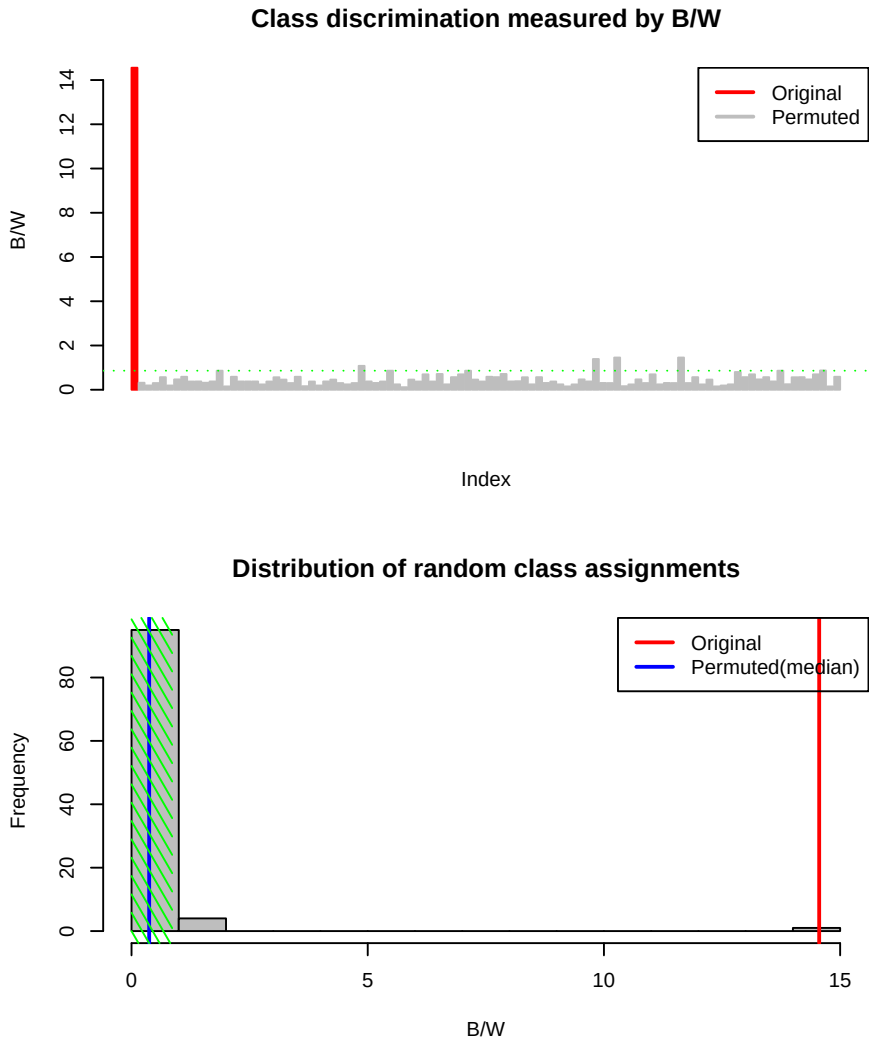

Figure 15: PLS-DA model validation by permutation tests. The top panel shows B/W ratio calculated for both original and permuted PLS-DA models. The bottom panel shows the distribution of random class assignments based on the frequencies of permuted B/W ratios. The green line (top) and green area (bottom) mark the 95% confidence regions of B/W for the permuted data.

2.3 Hierarchical Clustering

In (agglomerative) hierarchical cluster analysis, each sample begins as a separate cluster and the algorithm proceeds to combine them until all samples belong to one cluster. Two parameters need to be considered when performing hierarchical clustering. The first one is similarity measure - Euclidean distance, Pearson's correlation, Spearman's rank correlation. The other parameter is clustering algorithms, including average linkage (clustering uses the centroids of the observations), complete linkage (clustering uses the farthest pair of observations between the two groups), single linkage (clustering uses the closest pair of observations) and Ward's linkage (clustering to minimize the sum of squares of any two clusters). Heatmap is often presented as a visual aid in addition to the dendrogram.

Hierachical clustering is performed with the `hclust` function in package `stat`. The R script `clustering.R` is required. Figure 16 shows the clustering result in the form of a dendrogram. Figure 17 shows the clustering result in the form of a heatmap.

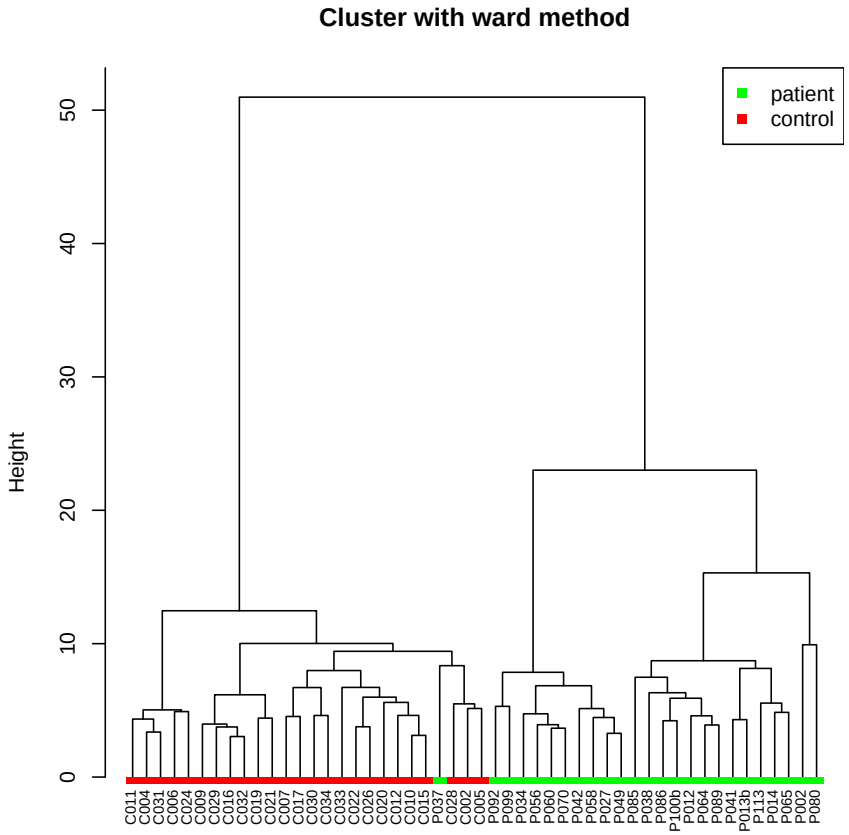

Figure 16: Clustering result shown as dendrogram (distance measure using `euclidean`, and clustering algorithm using `ward`).

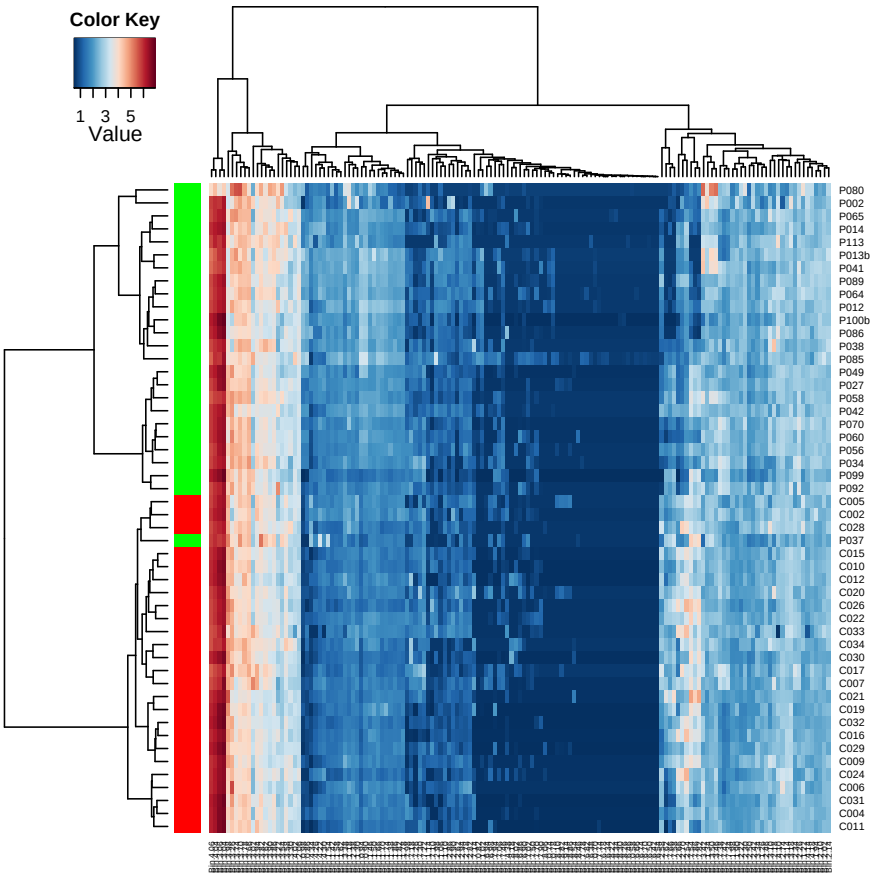

Figure 17: Clustering result shown as heatmap (distance measure using `euclidean`, and clustering algorithm using `ward`).

2.4 Random Forest (RF)

Random Forest is a supervised learning algorithm suitable for high dimensional data analysis. It uses an ensemble of classification trees, each of which is grown by random feature selection from a bootstrap sample at each branch. Class prediction is based on the majority vote of the ensemble. RF also provides other useful information such as OOB (out-of-bag) error and variable importance measure. During tree construction, about one-third of the instances are left out of the bootstrap sample. This OOB data is then used as test sample to obtain an unbiased estimate of the classification error (OOB error). Variable importance is evaluated by measuring the increase of the OOB error when it is permuted.

RF analysis is performed using the `randomForest` package[6]. The R script `classification.R` is required. Table 2 shows the confusion matrix of random forest. Figure 18 shows the cumulative error rates of random forest analysis for given parameters. Figure 19 shows the important features ranked by random forest. The OOB error is 0.04

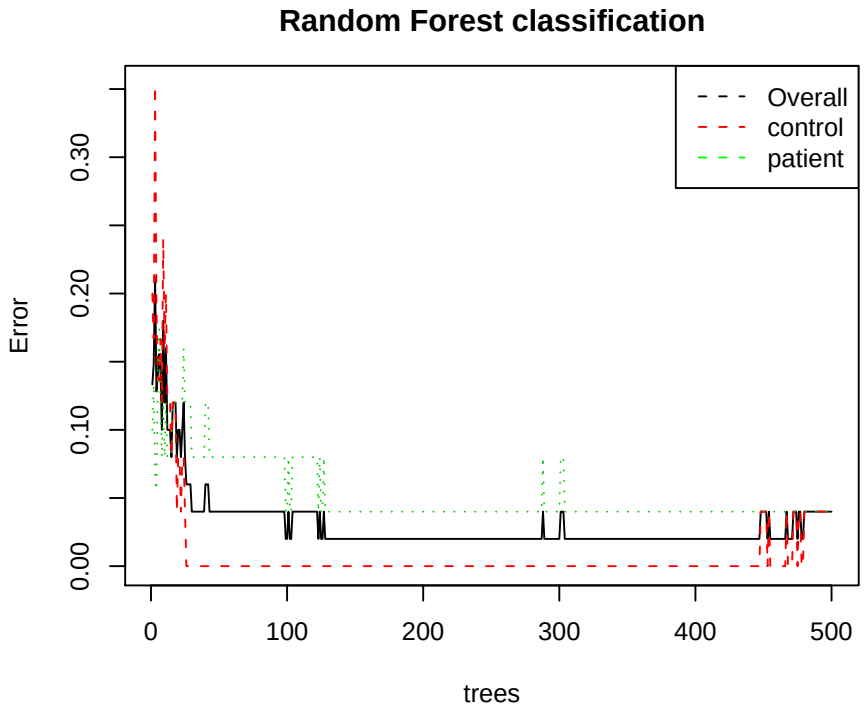

Figure 18: Cumulative error rates by Random Forest classification. The overall error rate is shown as the black line; the red and green lines represent the error rates for each class.

[6]Andy Liaw and Matthew Wiener. *Classification and Regression by randomForest*, 2002, R News

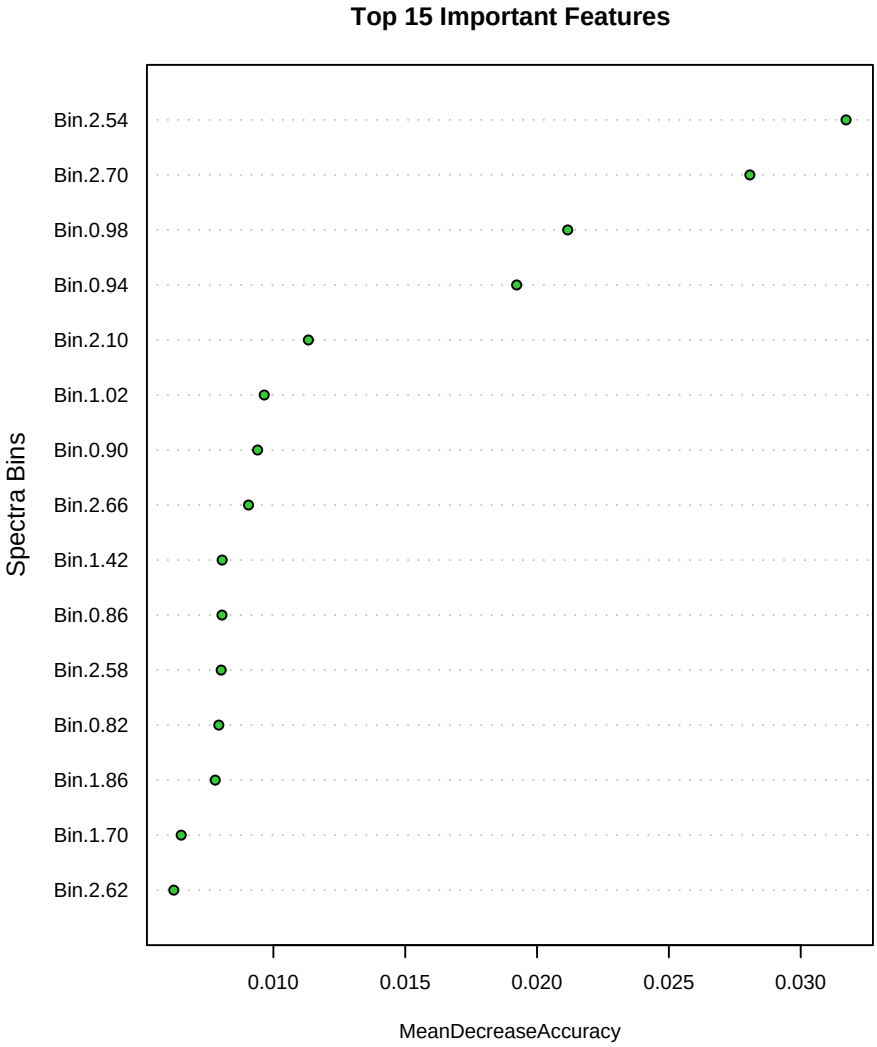

Figure 19: Significant features identified by Random Forest. The features are ranked by the mean decrease in classification accuracy when they are permuted.

	control	patient	class.error
control	24	1	0.04
patient	1	24	0.04

Table 2: Random Forest Classification Performance

3 Data Annotation

Please be advised that MetaboAnalyst also supports metabolomic data annotation. For NMR, MS, or GC-MS peak list data, users can perform peak identification by searching the corresponding libraries. For compound concentration data, users can perform pathway mapping. These tasks require a lot of manual efforts and are not performed by default.

The report was generated on Tue Apr 14 21:30:04 2009 with R version 2.8.1 (2008-12-22) on a i386-redhat-linux-gnu platform. Thank you for using MetaboAnalyst! For suggestions and feedback please contact Jeff Xia (*jianguox@ualberta.ca*).