Metabolomic Data Analysis with MetaboAnalyst 5.0

Name: guest14450995440294215590

July 13, 2021

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. Table 1 summarizes the result of the data processing steps.

1.1.1 Reading Peak Intensity Table

The peak intensity table should be uploaded in comma separated values (.csv) format. Samples can be in rows or columns, with class labels immediately following the sample IDs.

Samples are in rows and features in columns The uploaded file is in comma separated values (.csv) format. The uploaded data file contains 40 (samples) by 19 (peaks(mz/rt)) data matrix.

1.1.2 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 should all be non-negative numbers. By default, all missing values, zeros and negative values will be replaced by the half of the minimum positive value found within the data (see next section)

1.1.3 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 K-Nearest Neighbours (KNN), 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.

[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

Zero or missing values were replaced by 1/5 of the min positive value for each variable.

1.1.4 Data Filtering

The purpose of the data filtering is to identify and remove variables that are unlikely to be of use when modeling the data. No phenotype information are used in the filtering process, so the result can be used with any downstream analysis. This step can usually improves the results. Data filter is strongly recommended for datasets with large number of variables (> 250) datasets contain much noise (i.e.chemometrics data). Filtering can usually improve your results[2].

For data with number of variables < 250, this step will reduce 5% of variables; For variable number between 250 and 500, 10% of variables will be removed; For variable number bwteen 500 and 1000, 25% of variables will be removed; And 40% of variabled will be removed for data with over 1000 variables. The None option is only for less than 5000 features. Over that, if you choose None, the IQR filter will still be applied. In addition, the maximum allowed number of variables is ***10000***

No data filtering was performed.

Table 1: Summary of data processing results

	Features (positive)	Missing/Zero	Features (processed)
NSCLC_A549_1	19	0	19
NSCLC_H1703_2	19	0	19
NSCLC_H1703_1	19	0	19
NSCLC_A549_2	19	0	19
NSCLC_H1437_1	19	0	19
NSCLC_H2228_1	19	0	19
NSCLC_H2228_2	19	0	19
NSCLC_H1437_2	19	0	19
NSCLC_H3122_1	19	0	19
NSCLC_H322_2	19	0	19
NSCLC_H322_1	19	0	19
NSCLC_H358_2	19	0	19
NSCLC_H3122_2	19	0	19
NSCLC_H522_1	19	0	19
NSCLC_H522_2	19	0	19
NSCLC_HCC4006_1	19	0	19
NSCLC_H358_1	19	0	19
NSCLC_PC9_1	19	0	19
NSCLC_PC9_2	19	0	19
NSCLC_HCC4006_2	19	0	19
SCLC_86M1_2	19	0	19
SCLC_86M1_1	19	0	19
SCLC_16HV_1	19	0	19
SCLC_16HV_2	19	0	19
SCLC_DMS79_1	19	0	19
SCLC_DMS79_2	19	0	19
SCLC_H187_2	19	0	19
SCLC_H187_1	19	0	19
SCLC_H209_1	19	0	19
SCLC_H524_1	19	0	19
SCLC_H209_2	19	0	19
SCLC_H524_2	19	0	19
SCLC_H69_1	19	0	19
SCLC_H82_1	19	0	19
SCLC_H82_2	19	0	19
SCLC_H69_2	19	0	19
SCLC_N417_2	19	0	19
SCLC_N417_1	19	0	19
SCLC_SW210-5_1	19	0	19
SCLC_SW210_5_2	19	0	19

[2]Hackstadt AJ, Hess AM.*Filtering for increased power for microarray data analysis*, BMC Bioinformatics. 2009; 10: 11.

1.2 Data Normalization

The data is stored as a table with one sample per row and one variable (bin/peak/metabolite) per column. The normalization procedures implemented below are grouped into four categories. Sample specific normalization allows users to manually adjust concentrations based on biological inputs (i.e. volume, mass); row-wise normalization allows general-purpose adjustment for differences among samples; data transformation and scaling are two different approaches to make features more comparable. You can use one or combine both to achieve better results.

The normalization consists of the following options:

1. Row-wise procedures:
   - Sample specific normalization (i.e. normalize by dry weight, volume)
   - Normalization by the sum
   - Normalization by the sample median
   - Normalization by a reference sample (probabilistic quotient normalization)[3]
   - Normalization by a pooled or average sample from a particular group
   - Normalization by a reference feature (i.e. creatinine, internal control)
   - Quantile normalization
2. Data transformation :
   - Generalized log transformation (glog 2)
   - Cube root transformation
3. Data scaling:
   - Mean centering (mean-centered only)
   - Auto 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)

Figure 1 shows the effects before and after normalization.

[3]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

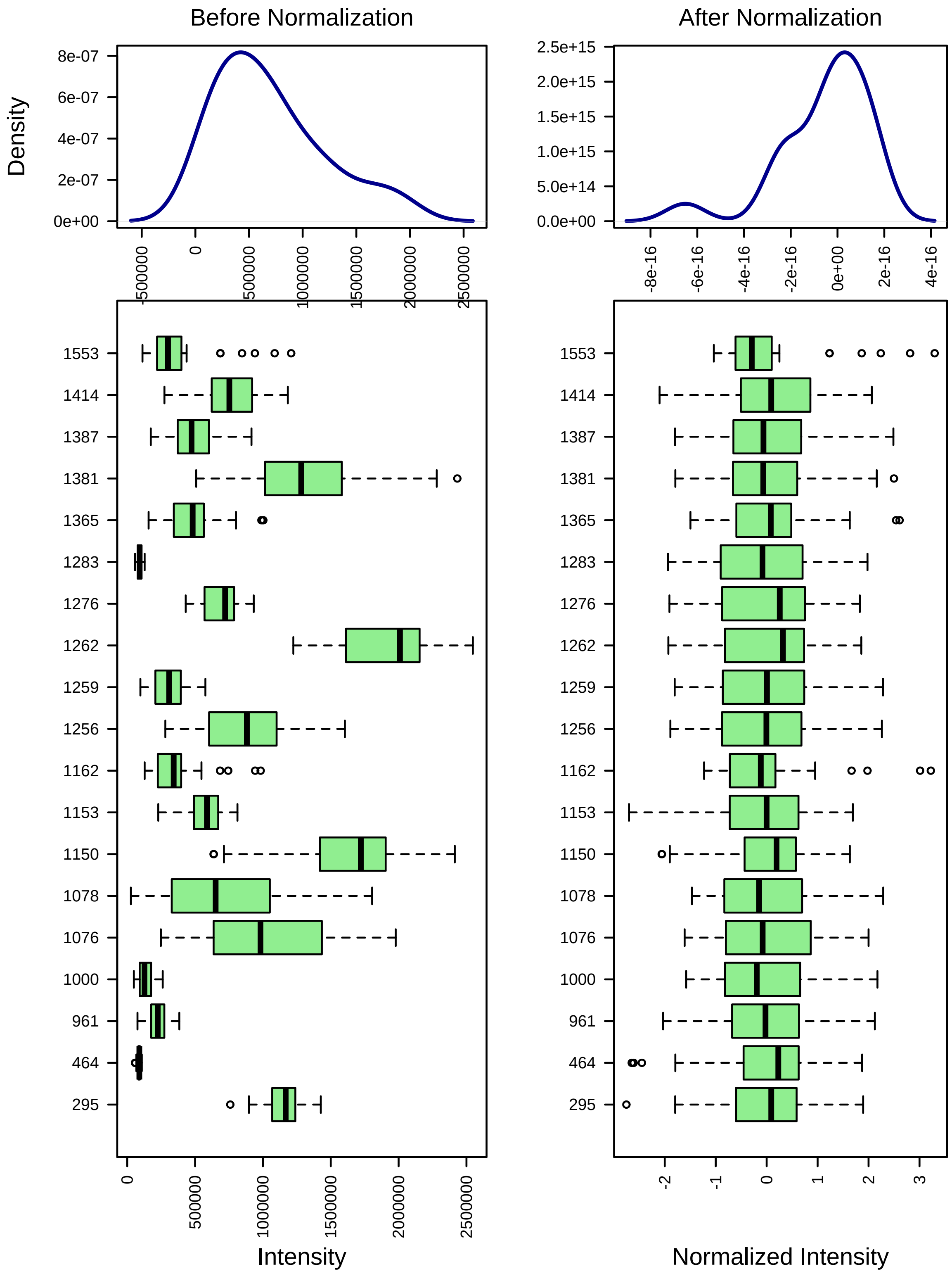

Figure 1: Box plots and kernel density plots before and after normalization. The boxplots show at most 50 features due to space limit. The density plots are based on all samples. Selected methods : Row-wise normalization: N/A; Data transformation: N/A; Data scaling: Autoscaling.

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
   - One-way ANOVA and post-hoc analysis
   - Correlation analysis
2. Multivariate analysis 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)

`Please note:  some advanced methods are available only for two-group sample analyais.`

2.1 Univariate Analysis

Univariate analysis methods are the most common methods used for exploratory data analysis. For two-group data, MetaboAnalyst provides Fold Change (FC) analysis, t-tests, and volcano plot which is a combination of the first two methods. All three these methods support both unpaired and paired analyses. For multi-group analysis, MetaboAnalyst provides two types of analysis - one-way analysis of variance (ANOVA) with associated post-hoc analyses, and correlation analysis to identify signficant compounds that follow a given pattern. The univariate analyses provide a preliminary overview about features that are potentially significant in discriminating the conditions under study.

For paired fold change analysis, the algorithm first counts the total number of pairs with fold changes that are consistently above/below the specified FC threshold for each variable. A variable will be reported as significant if this number is above a given count threshold (default > 75% of pairs/variable)

Figure 2 shows the important features identified by fold change analysis. Table 2 shows the details of these features; Figure 3 shows the important features identified by t-tests. Table 3 shows the details of these features; Figure 4 shows the important features identified by volcano plot. Table 4 shows the details of these features.

Please note, the purpose of fold change is to compare absolute value changes between two group means. Therefore, the data before column normalization will be used instead. Also note, the result is plotted in log2 scale, so that same fold change (up/down regulated) will have the same distance to the zero baseline.

Figure 2: Important features selected by fold-change analysis with threshold 2. The red circles represent features above the threshold. Note the values are on log scale, so that both up-regulated and down-regulated features can be plotted in a symmetrical way

Table 2: Important features identified by fold change analysis

	Peaks(mz/rt)	Fold Change	log2(FC)
1	1076	2.2015	1.1385
2	1553	2.0073	1.0053

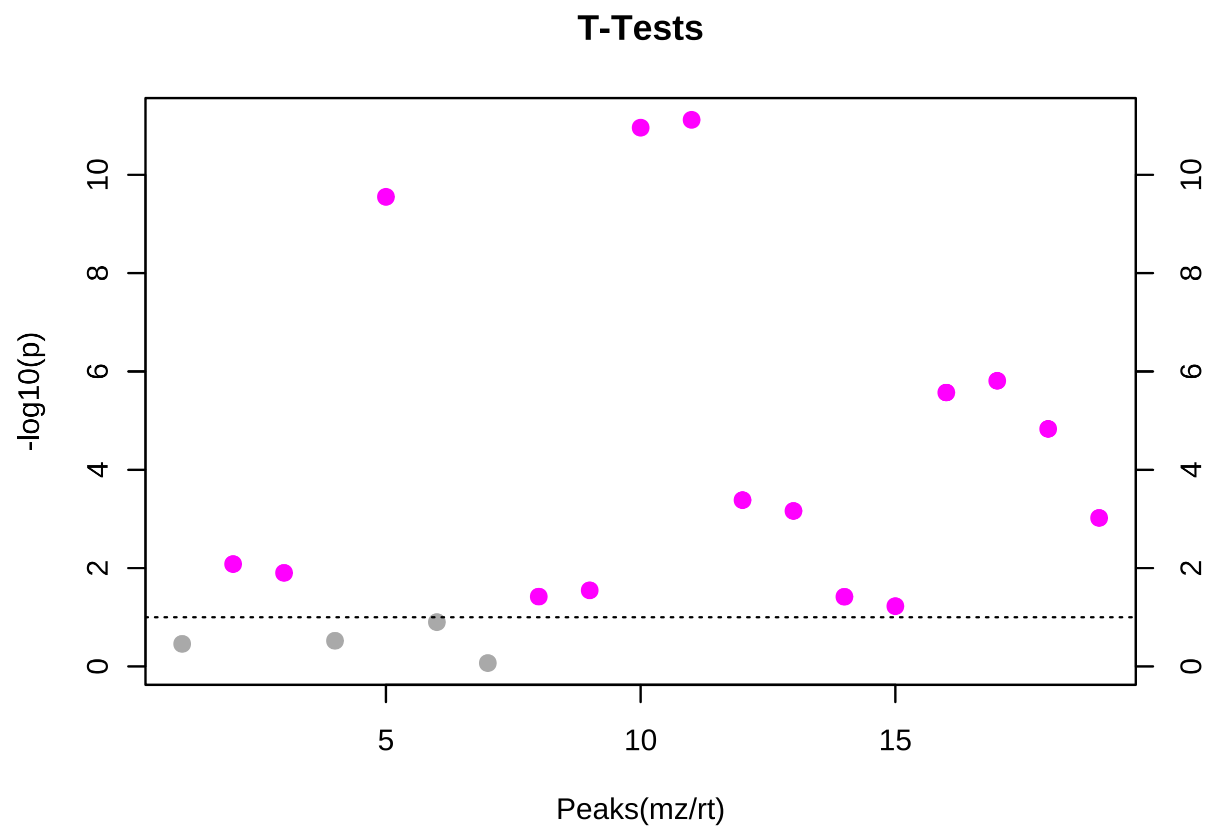

Figure 3: Important features selected by t-tests with threshold 0.1. The red circles represent features above the threshold. Note the p values are transformed by -log10 so that the more significant features (with smaller p values) will be plotted higher on the graph.

Table 3: Important features identified by t-tests

	Peaks(mz/rt)	t.stat	p.value	-log10(p)	FDR
1	1259	9.7134	7.613e-12	11.118	1.0455e-10
2	1256	9.5829	1.1006e-11	10.958	1.0455e-10
3	1076	8.4672	2.8077e-10	9.5516	1.7782e-09
4	1387	5.6832	1.5488e-06	5.81	7.3569e-06
5	1381	5.5088	2.6858e-06	5.5709	1.0206e-05
6	1414	-4.9666	1.4734e-05	4.8317	4.6659e-05
7	1262	-3.8706	0.00041375	3.3833	0.001123
8	1276	-3.6961	0.00068776	3.1626	0.0016334
9	1553	3.5825	0.00095318	3.0208	0.0020123
10	464	-2.7863	0.008274	2.0823	0.015721
11	961	-2.6221	0.012502	1.903	0.021595
12	1162	-2.2801	0.028302	1.5482	0.044812
13	1153	2.15	0.037987	1.4204	0.051921
14	1283	-2.1468	0.038258	1.4173	0.051921
15	1365	1.9434	0.059403	1.2262	0.075244

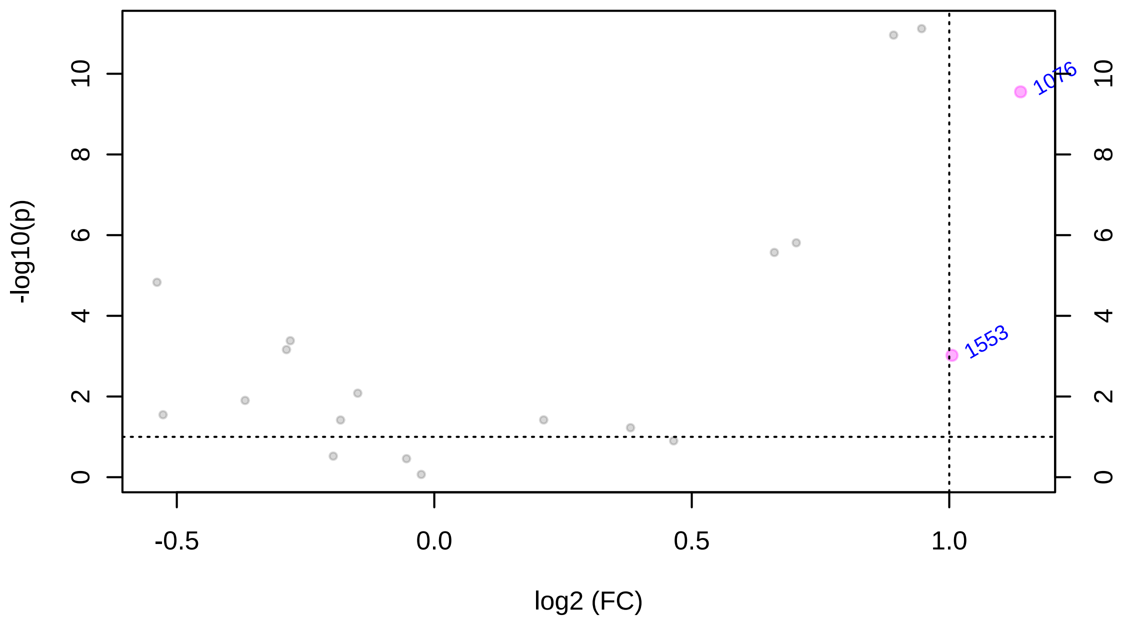

Figure 4: Important features selected by volcano plot with fold change threshold (x) 2 and t-tests threshold (y) 0.1. The red circles represent features above the threshold. Note both fold changes and p values are log transformed. The further its position away from the (0,0), the more significant the feature is.

Table 4: Important features identified by volcano plot

	Peaks(mz/rt)	FC	log2(FC)	raw.pval	-log10(p)
1	1076	2.2015	1.1385	2.8077e-10	9.5516
2	1553	2.0073	1.0053	0.00095318	3.0208

2.2 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 5 is pairwise score plots providing an overview of the various seperation patterns among the most significant PCs; Figure 6 is the scree plot showing the variances explained by the selected PCs; Figure 7 shows the 2-D scores plot between selected PCs; Figure 8 shows the 3-D scores plot between selected PCs; Figure 9 shows the loadings plot between the selected PCs; Figure 10 shows the biplot between the selected PCs.

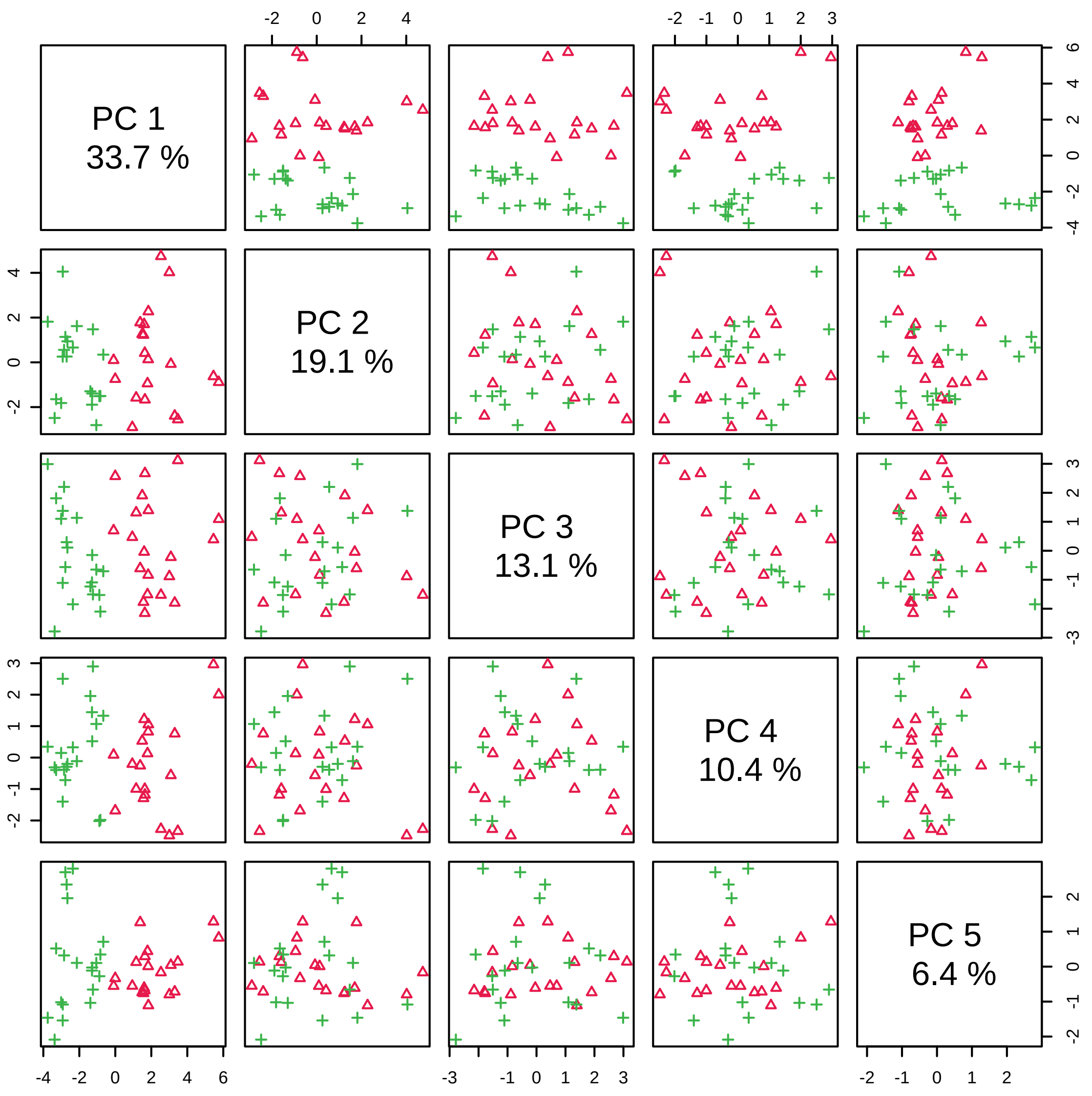

Figure 5: Pairwise score plots between the selected PCs. The explained variance of each PC is shown in the corresponding diagonal cell.

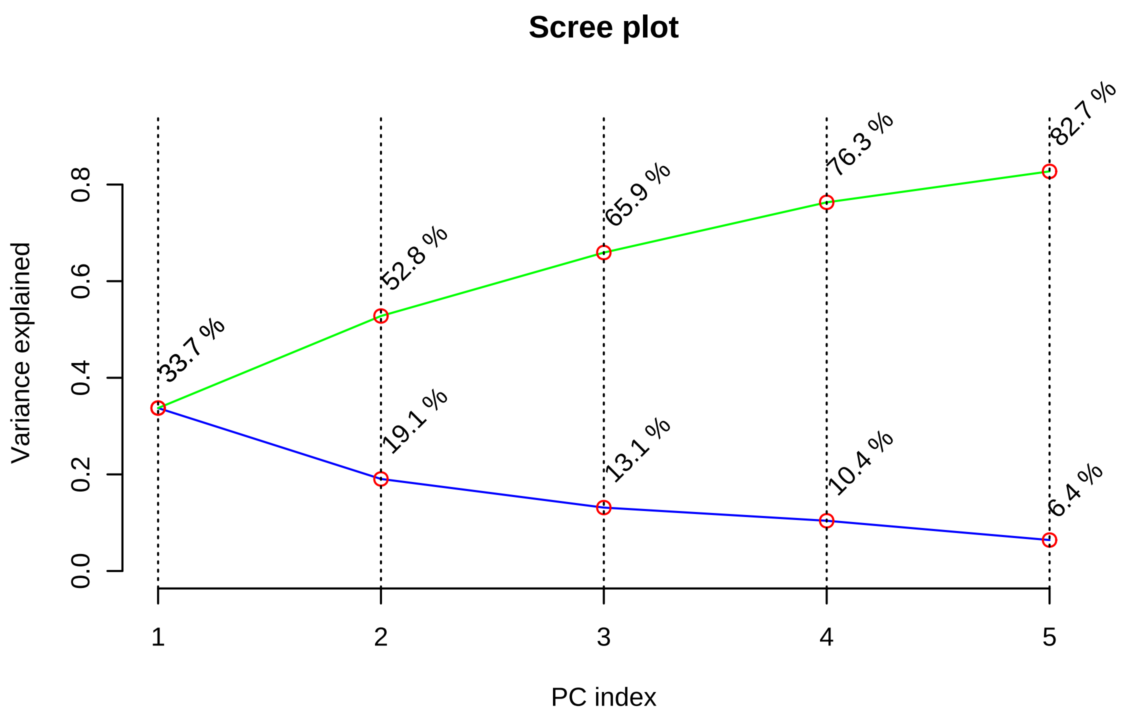

Figure 6: Scree plot shows the variance explained by PCs. The green line on top shows the accumulated variance explained; the blue line underneath shows the variance explained by individual PC.

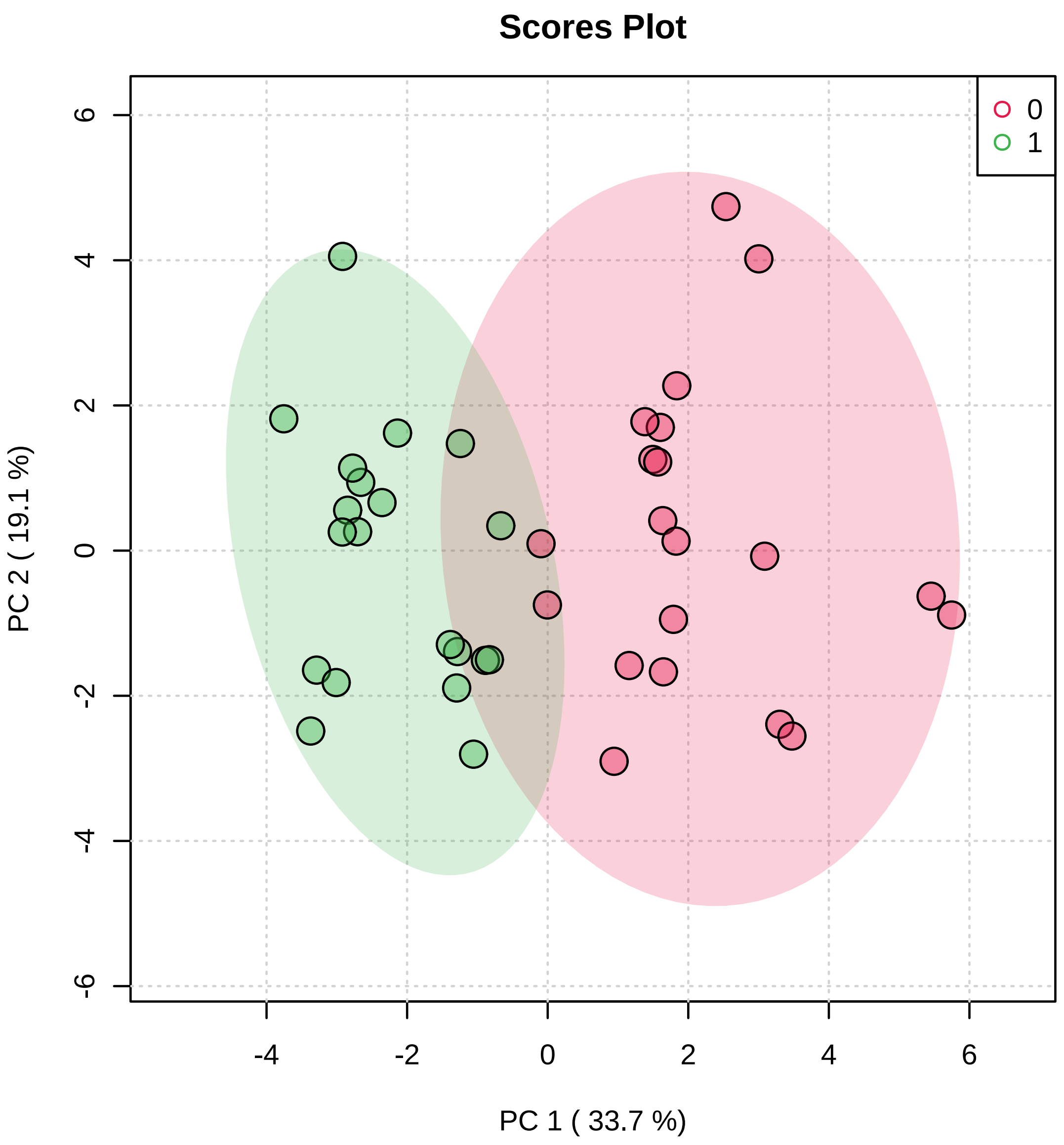

Figure 7: Scores plot between the selected PCs. The explained variances are shown in brackets.

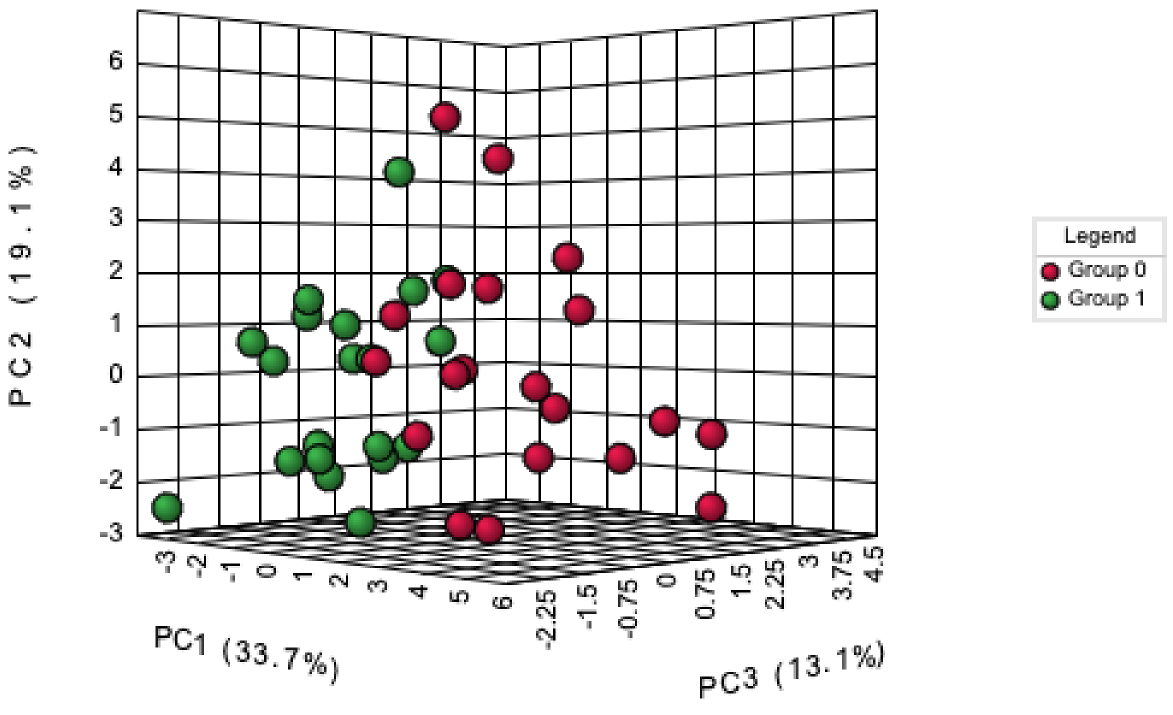

Figure 8: 3D score plot between the selected PCs. The explained variances are shown in brackets.

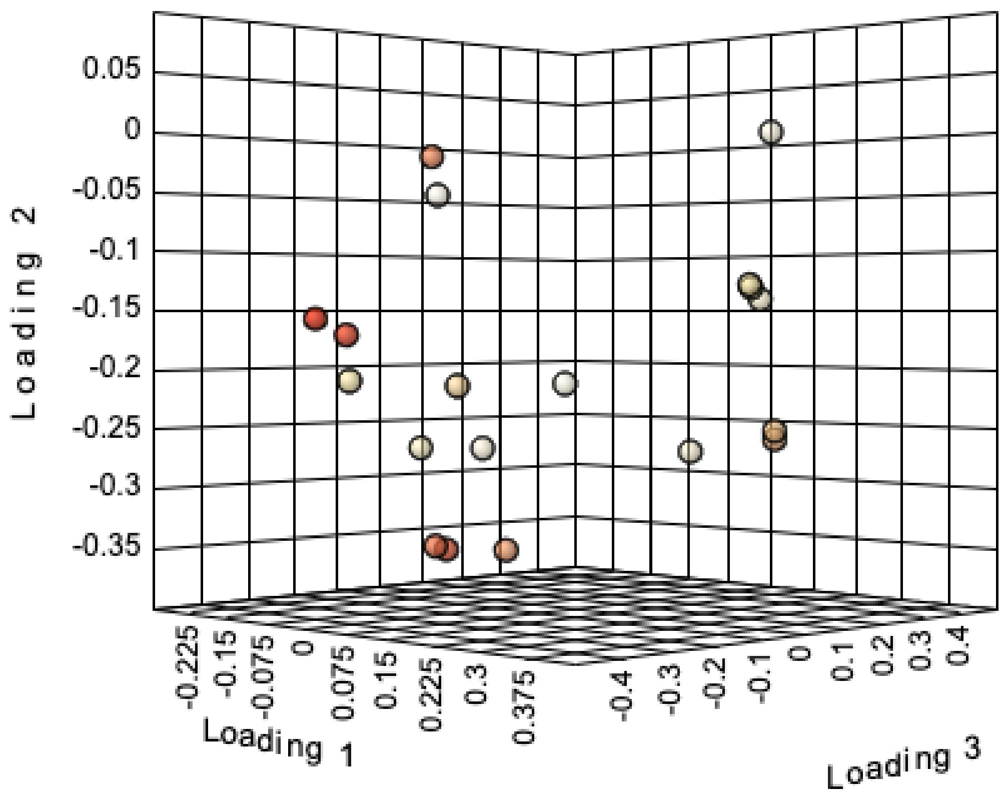

Figure 9: Loadings plot for the selected PCs.

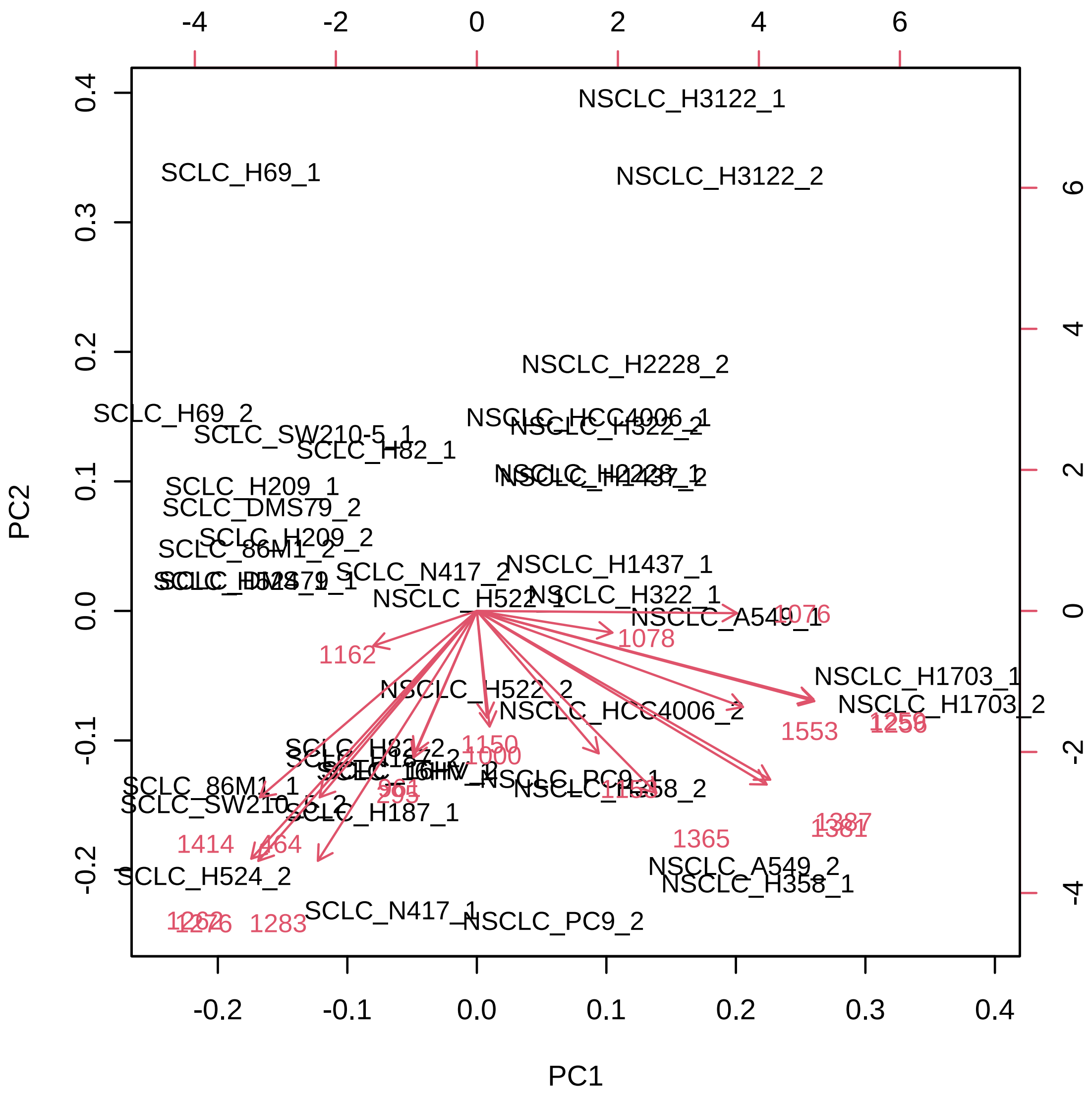

Figure 10: PCA biplot between the selected PCs. Note, you may want to test different centering and scaling normalization methods for the biplot to be displayed properly.

2.3 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[4]. The classification and cross-validation are performed using the corresponding wrapper function offered by the `caret` package[5].

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. MetaboAnalyst supports two types of test statistics for measuring the class discrimination. The first one is based on prediction accuracy during training. The second one is separation distance based on the ratio of the between group sum of the squares and the within group sum of squares (B/W-ratio). If the observed test statistic is part of the distribution based on the permuted class assignments, the class discrimination cannot be considered significant from a statistical point of view.[6].

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. Please note, VIP scores are calculated for each components. When more than components are used to calculate the feature importance, the average of the VIP scores are used. 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. Please note, for multiple-group (more than two) analysis, the same number of predictors will be built for each group. Therefore, the coefficient of each feature will be different depending on which group you want to predict. The average of the feature coefficients are used to indicate the overall coefficient-based importance.

Figure 11 shows the overview of scores plots; Figure 12 shows the 2-D scores plot between selected components; Figure 13 shows the 3-D scores plot between selected components; Figure 14 shows the loading plot between the selected components;Figure 15 shows the classification performance with different number of components; Figure 16 shows the results of permutation test for model validation; Figure 17 shows important features identified by PLS-DA.

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

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

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

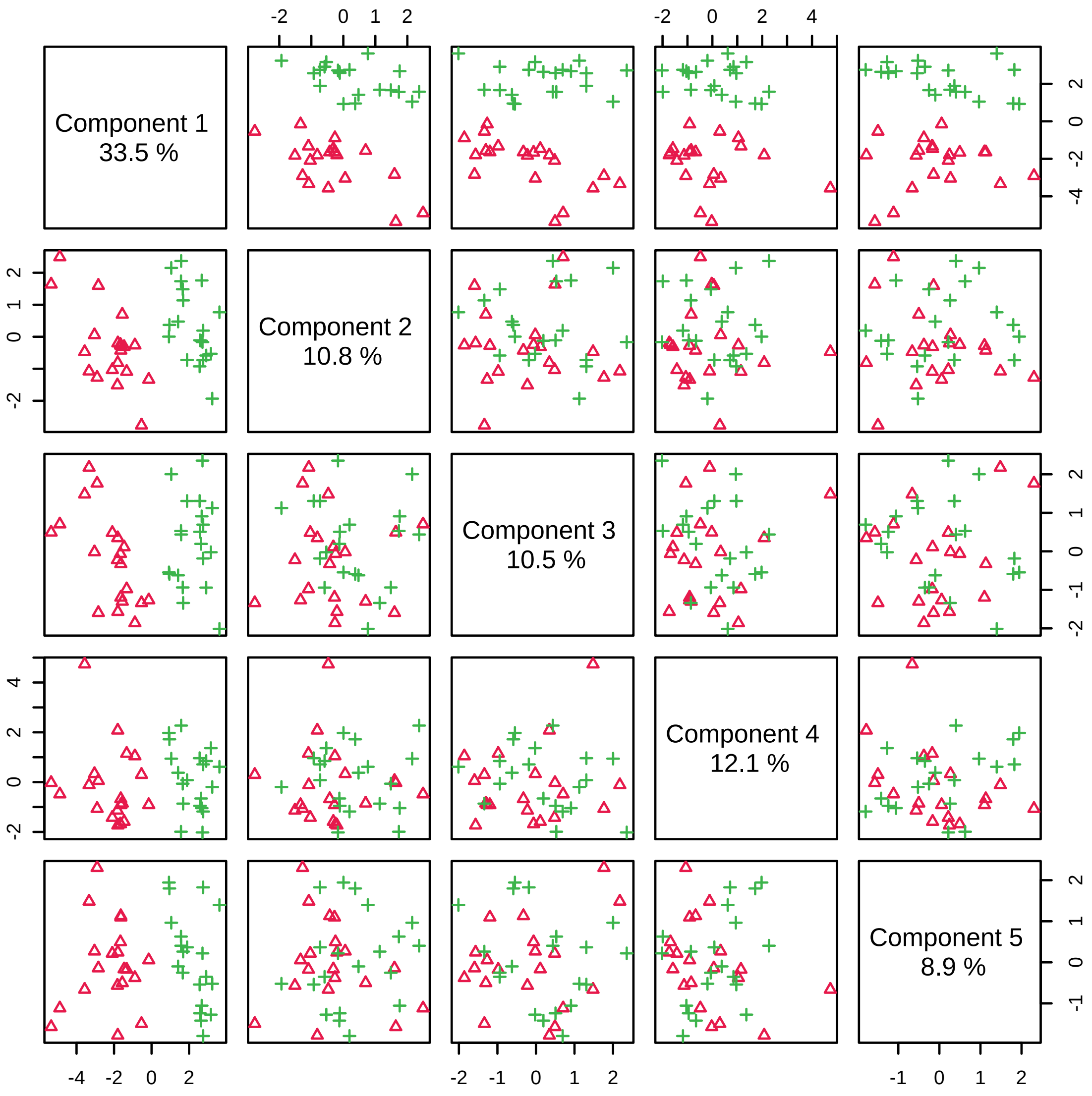

Figure 11: Pairwise scores plots between the selected components. The explained variance of each component is shown in the corresponding diagonal cell.

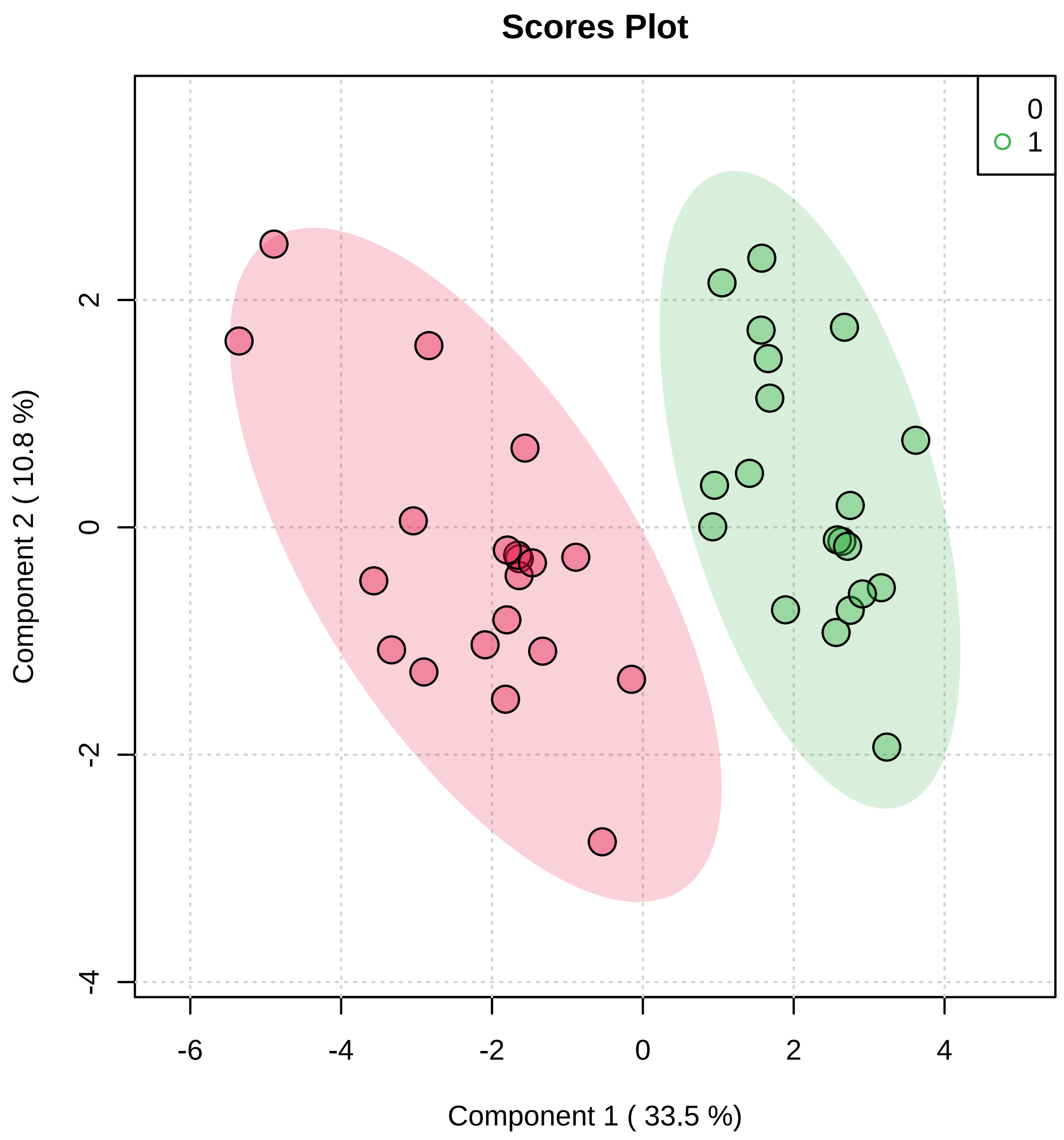

Figure 12: Scores plot between the selected PCs. The explained variances are shown in brackets.

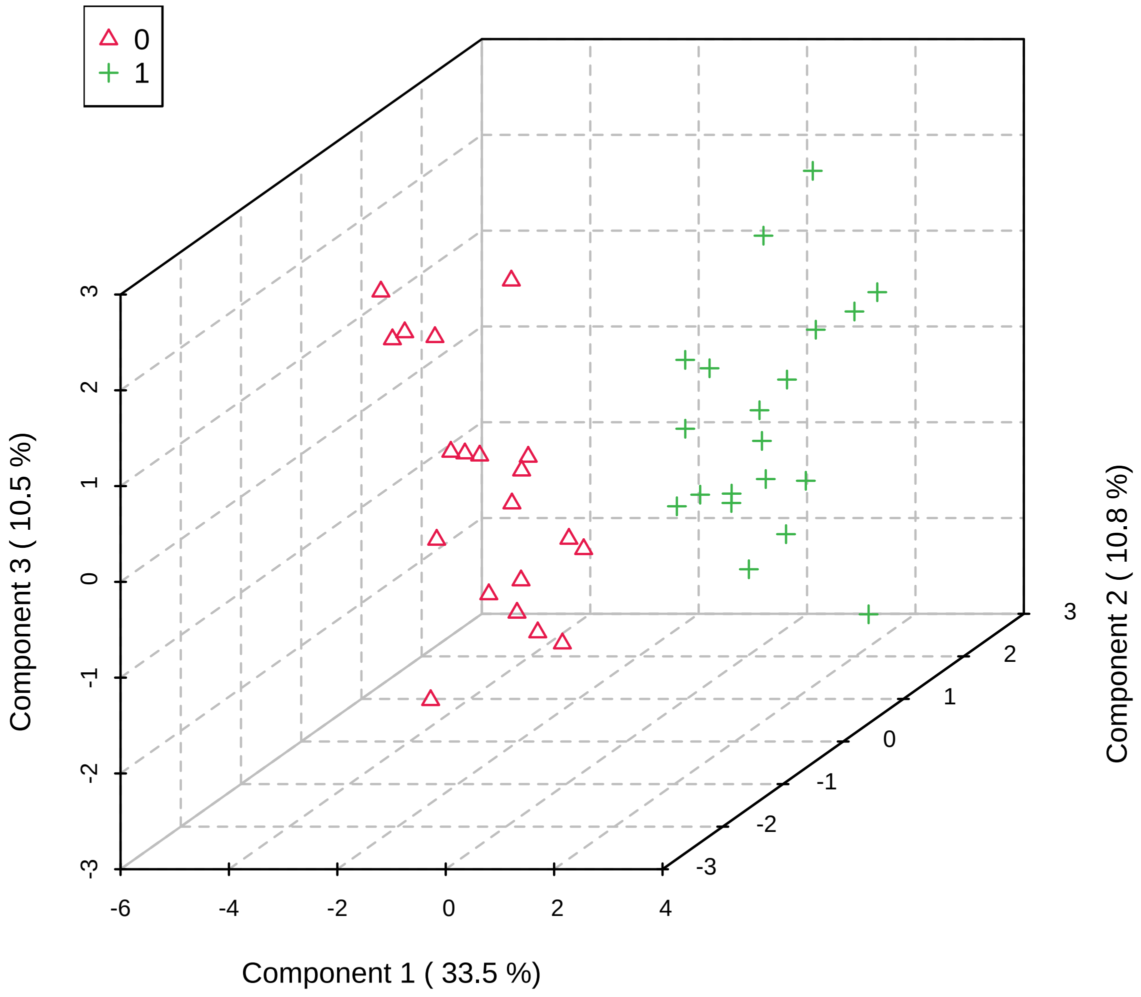

Figure 13: 3D scores plot between the selected PCs. The explained variances are shown in brackets.

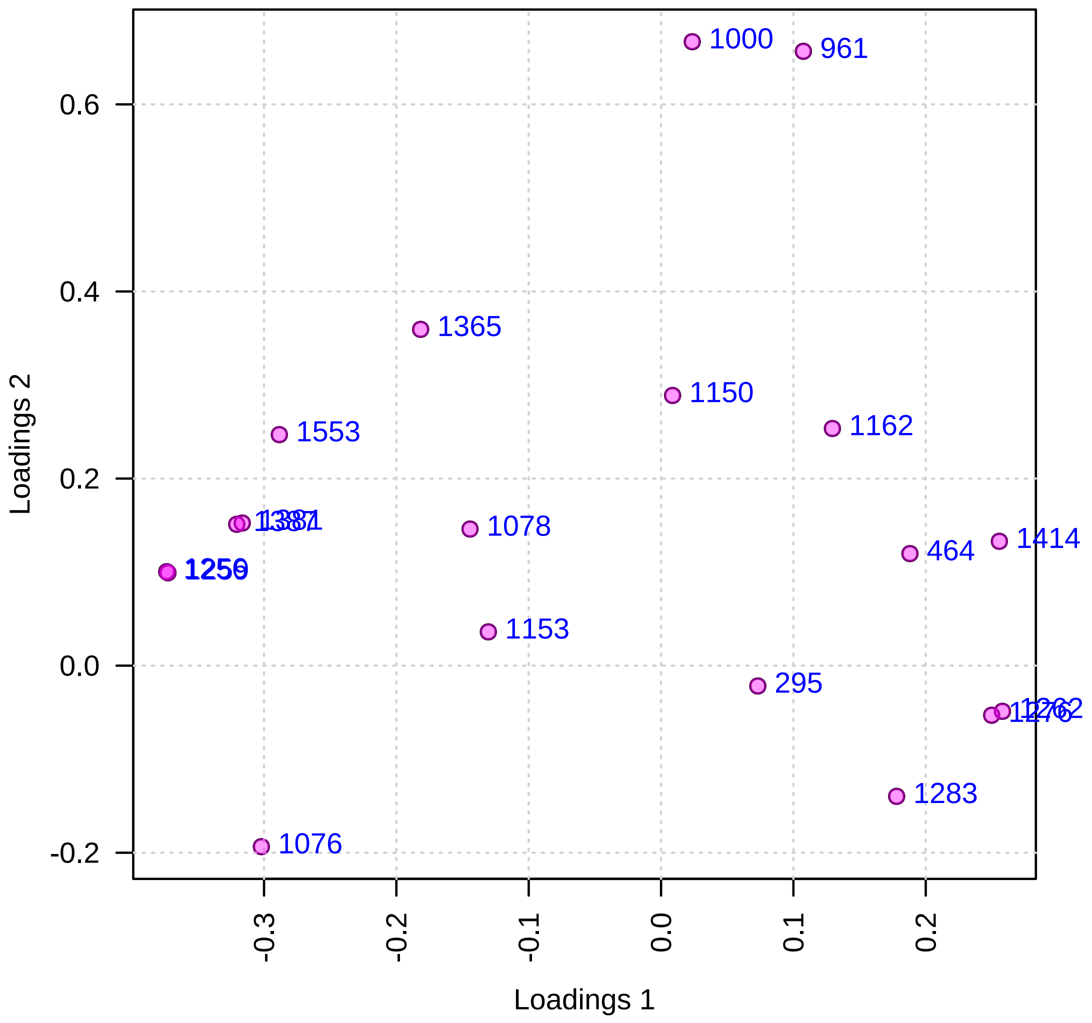

Figure 14: Loadings plot between the selected PCs.

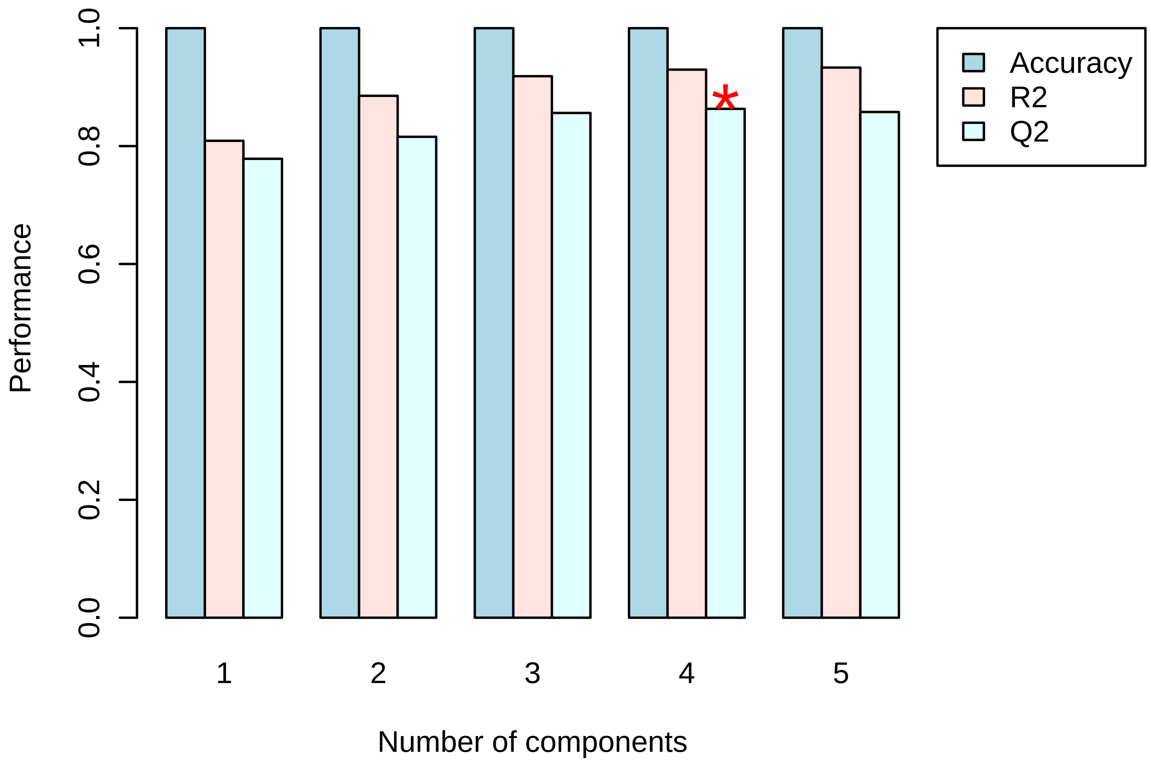

Figure 15: PLS-DA classification using different number of components. The red star indicates the best classifier.

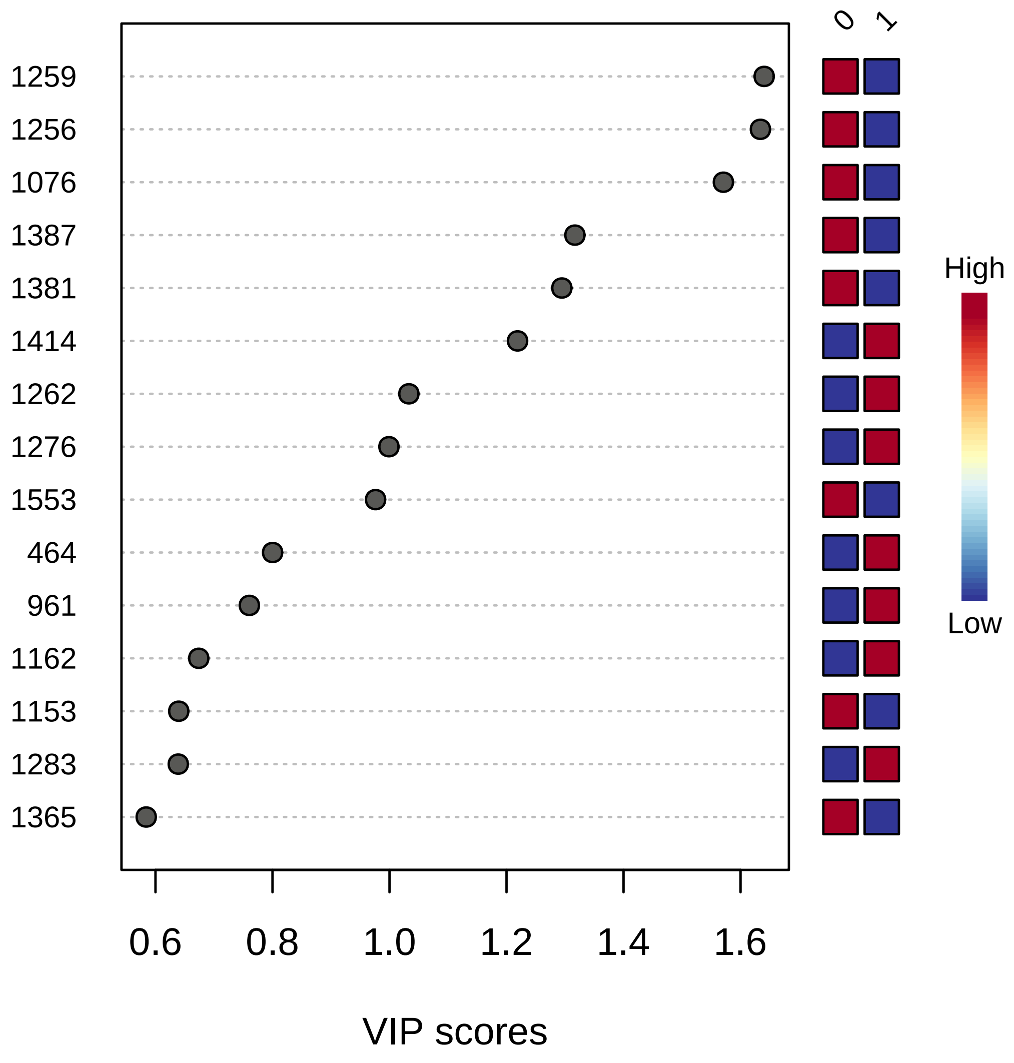

Figure 16: Important features identified by PLS-DA. The colored boxes on the right indicate the relative concentrations of the corresponding metabolite in each group under study.

2.4 K-means Clustering

K-means clustering is a nonhierarchical clustering technique. It begins by creating k random clusters (k is supplied by user). The program then calculates the mean of each cluster. If an observation is closer to the centroid of another cluster then the observation is made a member of that cluster. This process is repeated until none of the observations are reassigned to a different cluster.

K-means analysis is performed using the `kmeans` function in the package `stat`. Figure 18 shows clustering the results. Table 5 shows the members in each cluster from K-means analysis.

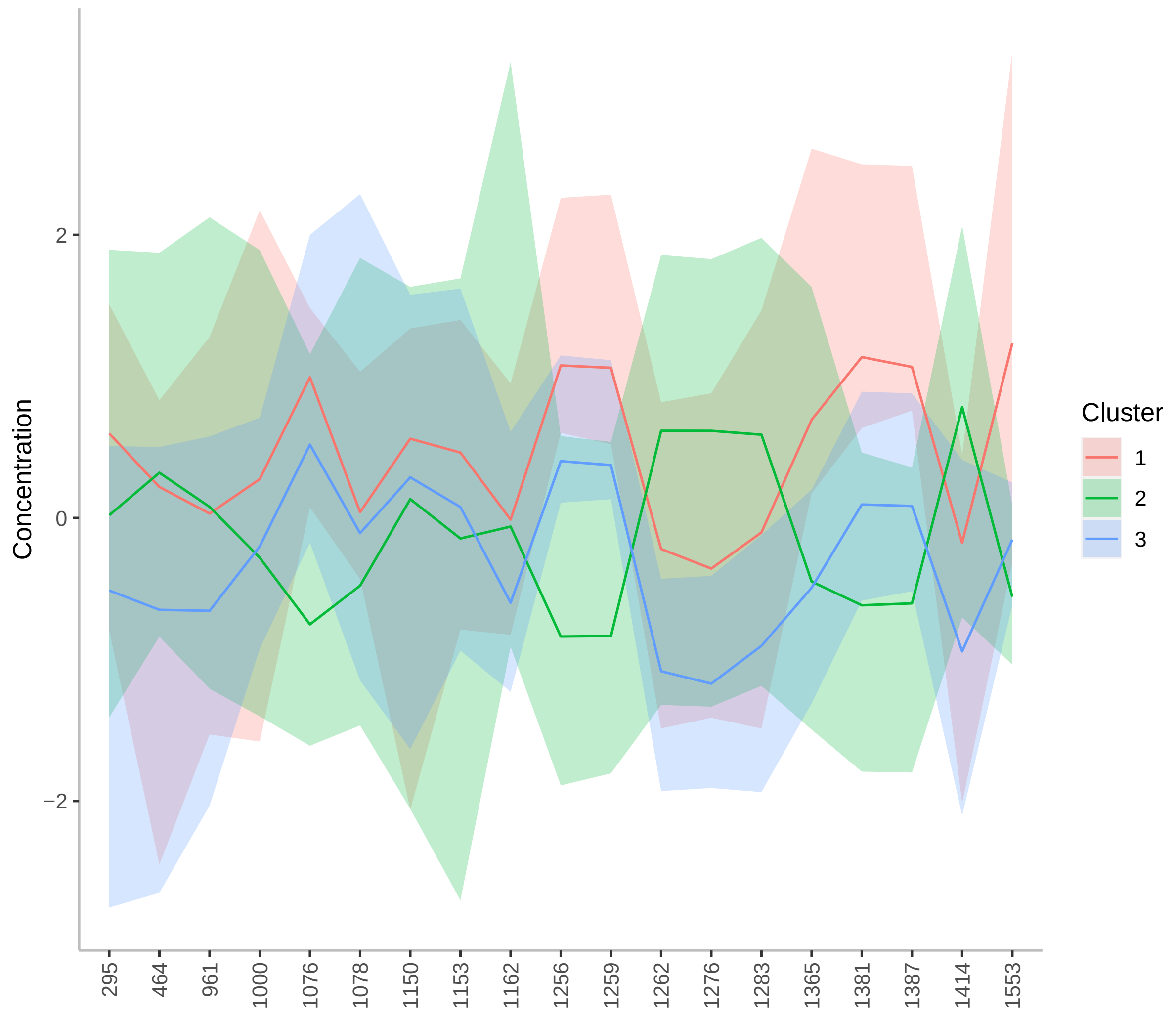

Figure 17: K-means cluster analysis. The x-axes are variable indices and y-axes are relative intensities. The blue lines represent median intensities of corresponding clusters

Table 5: Clustering result using K-means

	Samples in each cluster
Cluster(1)	NSCLC_A549_1 NSCLC_H1703_2 NSCLC_H1703_1 NSCLC_A549_2 NSCLC_H358_2 NSCLC_H358_1 NSCLC_PC9_1 NSCLC_PC9_2 NSCLC_HCC4006_2
Cluster(2)	NSCLC_H522_1 NSCLC_H522_2 SCLC_86M1_2 SCLC_86M1_1 SCLC_16HV_1 SCLC_16HV_2 SCLC_DMS79_1 SCLC_DMS79_2 SCLC_H187_2 SCLC_H187_1 SCLC_H209_1 SCLC_H524_1 SCLC_H209_2 SCLC_H524_2 SCLC_H69_1 SCLC_H82_1 SCLC_H82_2 SCLC_H69_2 SCLC_N417_2 SCLC_N417_1 SCLC_SW210-5_1 SCLC_SW210_5_2
Cluster(3)	NSCLC_H1437_1 NSCLC_H2228_1 NSCLC_H2228_2 NSCLC_H1437_2 NSCLC_H3122_1 NSCLC_H322_2 NSCLC_H322_1 NSCLC_H3122_2 NSCLC_HCC4006_1

2.5 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, variable importance measure, and outlier measures. 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. The outlier measures are based on the proximities during tree construction.

RF analysis is performed using the `randomForest` package[7]. Table 6 shows the confusion matrix of random forest. Figure 19 shows the cumulative error rates of random forest analysis for given parameters. Figure 20 shows the important features ranked by random forest. Figure 21 shows the outlier measures of all samples for the given parameters. The OOB error is 0

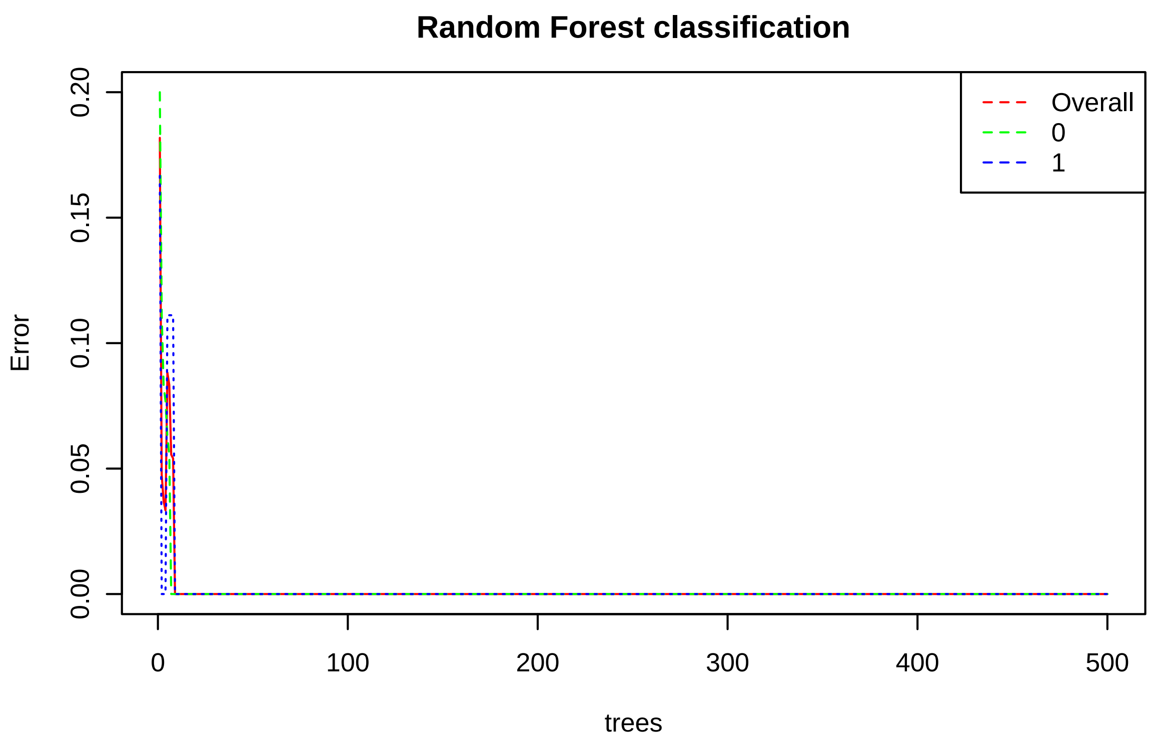

Figure 18: Cumulative error rates by Random Forest classification. The overall error rate is shown as the black line; the red and green lines represent the error rates for each class.

	0	1	class.error
0	20.00	0.00	0.00
1	0.00	20.00	0.00

Table 6: Random Forest Classification Performance

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

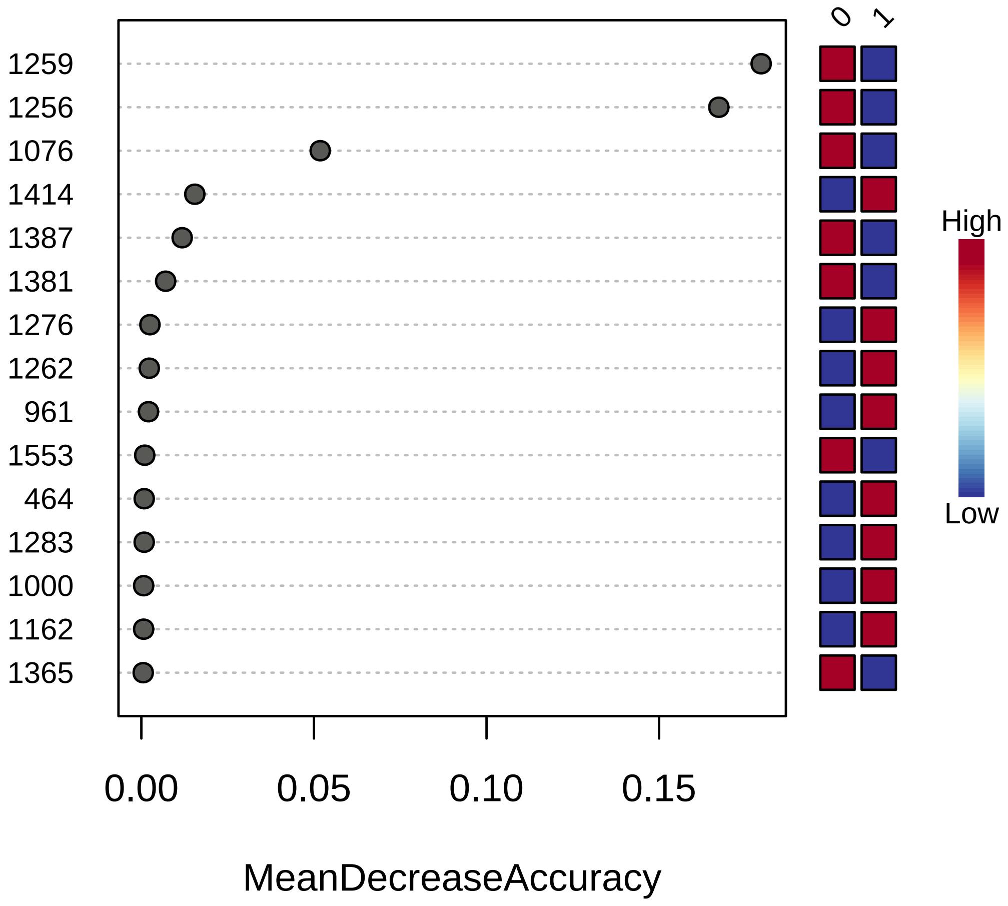

Figure 19: Significant features identified by Random Forest. The features are ranked by the mean decrease in classification accuracy when they are permuted.

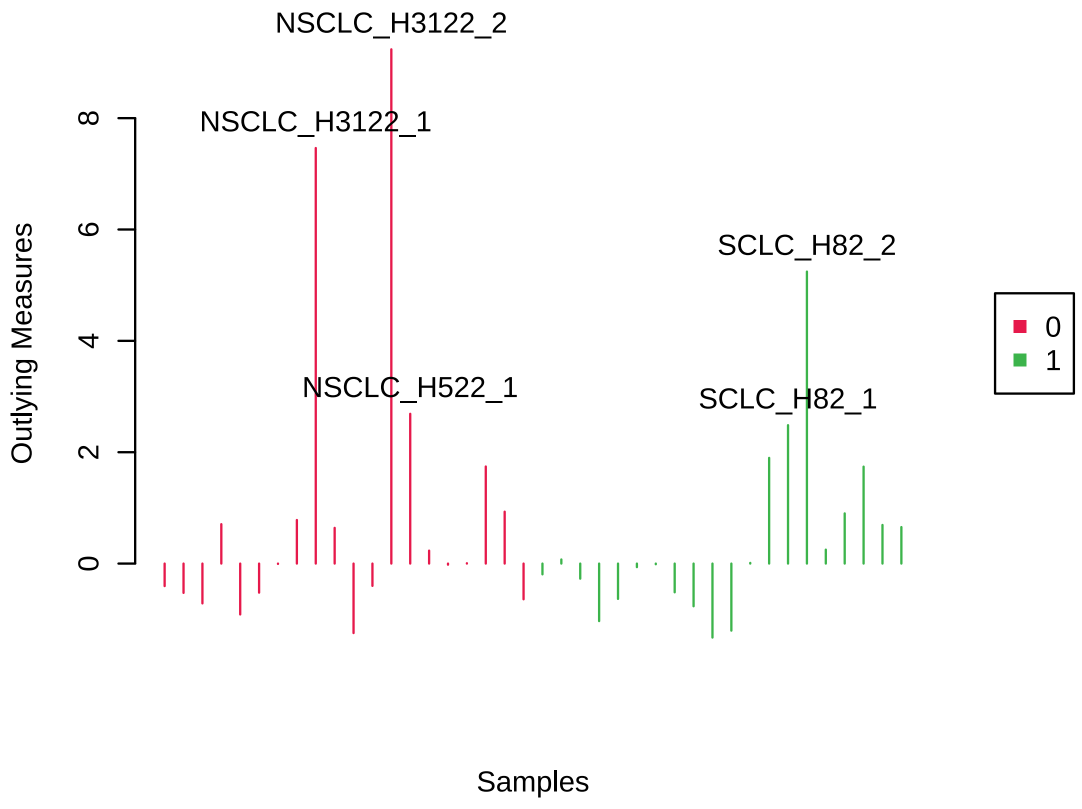

Figure 20: Potential outliers identified by Random Forest. Only the top five are labeled.

3 Appendix: R Command History

```
 [1] "mSet<-InitDataObjects(\"pktable\", \"stat\", FALSE)"
 [2] "mSet<-Read.TextData(mSet, \"Replacing_with_your_file_path\", \"rowu\", \"disc\");"
 [3] "mSet<-SanityCheckData(mSet)"
 [4] "mSet<-ContainMissing(mSet)"
 [5] "mSet<-ReplaceMin(mSet);"
 [6] "mSet<-SanityCheckData(mSet)"
 [7] "mSet<-ContainMissing(mSet)"
 [8] "mSet<-PreparePrenormData(mSet)"
 [9] "mSet<-Normalization(mSet, \"NULL\", \"NULL\", \"AutoNorm\", ratio=FALSE, ratioNum=20)"
[10] "mSet<-PlotNormSummary(mSet, \"norm_0_\", \"png\", 72, width=NA)"
[11] "mSet<-PlotSampleNormSummary(mSet, \"snorm_0_\", \"png\", 72, width=NA)"
[12] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, TRUE, \"fdr\", FALSE)"
[13] "mSet<-PlotTT(mSet, \"tt_0_\", \"png\", 72, width=NA)"
[14] "mSet<-Volcano.Anal(mSet, FALSE, 2.0, 0, F, 0.1, TRUE, \"raw\")"
[15] "mSet<-PlotVolcano(mSet, \"volcano_0_\",1, \"png\", 72, width=NA)"
[16] "mSet<-PCA.Anal(mSet)"
[17] "mSet<-PlotPCAPairSummary(mSet, \"pca_pair_0_\", \"png\", 72, width=NA, 5)"
[18] "mSet<-PlotPCAScree(mSet, \"pca_scree_0_\", \"png\", 72, width=NA, 5)"
[19] "mSet<-PlotPCA2DScore(mSet, \"pca_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0)"
[20] "mSet<-PlotPCALoading(mSet, \"pca_loading_0_\", \"png\", 72, width=NA, 1,2);"
[21] "mSet<-PlotPCABiplot(mSet, \"pca_biplot_0_\", \"png\", 72, width=NA, 1,2)"
[22] "mSet<-PlotPCA3DLoading(mSet, \"pca_loading3d_0_\", \"json\", 1,2,3)"
[23] "mSet<-Kmeans.Anal(mSet, 3)"
[24] "mSet<-PlotKmeans(mSet, \"km_0_\", \"png\", 72, width=NA, \"default\", \"F\")"
[25] "mSet<-PlotClustPCA(mSet, \"km_pca_0_\", \"png\", 72, width=NA, \"default\", \"km\", \"F\")"
[26] "mSet<-PLSR.Anal(mSet, reg=TRUE)"
[27] "mSet<-PlotPLSPairSummary(mSet, \"pls_pair_0_\", \"png\", 72, width=NA, 5)"
[28] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0)"
[29] "mSet<-PlotPLS3DScoreImg(mSet, \"pls_score3d_0_\", \"png\", 72, width=NA, 1,2,3, 40)"
[30] "mSet<-PlotPLSLoading(mSet, \"pls_loading_0_\", \"png\", 72, width=NA, 1, 2);"
[31] "mSet<-PlotPLS3DLoading(mSet, \"pls_loading3d_0_\", \"json\", 1,2,3)"
[32] "mSet<-PLSDA.CV(mSet, \"T\",5, \"Q2\")"
[33] "mSet<-PlotPLS.Classification(mSet, \"pls_cv_0_\", \"png\", 72, width=NA)"
[34] "mSet<-PlotPLS.Imp(mSet, \"pls_imp_0_\", \"png\", 72, width=NA, \"vip\", \"Comp. 1\", 15,FALSE)
[35] "mSet<-RF.Anal(mSet, 500,7,1)"
[36] "mSet<-PlotRF.Classify(mSet, \"rf_cls_0_\", \"png\", 72, width=NA)"
[37] "mSet<-PlotRF.VIP(mSet, \"rf_imp_0_\", \"png\", 72, width=NA)"
[38] "mSet<-PlotRF.Outlier(mSet, \"rf_outlier_0_\", \"png\", 72, width=NA)"
[39] "mSet<-SaveTransformedData(mSet)"
[40] "mSet<-PreparePDFReport(mSet, \"guest14450995440294215590\")\n"
```

The report was generated on Tue Jul 13 16:12:43 2021 with R version 4.0.2 (2020-06-22).