Metabolomic Data Analysis with MetaboAnalyst 5.0

Name: guest14509913071070504278

July 15, 2022

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 columns and features in rows. The uploaded file is in comma separated values (.csv) format. The uploaded data file contains 20 (samples) by 91 (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

Missing variables were replaced by LoDs (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 filtering was applied

Table 1: Summary of data processing results

	Features (positive)	Missing/Zero	Features (processed)
LA_6_4_C.raw	63	28	91
LA_6_3_C.raw	59	32	91
LA_6_6_C.raw	66	25	91
LA_6_7_C.raw	68	23	91
QC_C_4.raw	80	11	91
QC_C_5.raw	73	18	91
QC_C_9.raw	75	16	91
QC_C_8.raw	81	10	91
QC_C_6.raw	83	8	91
QC_C_10.raw	83	8	91
QC_C_11.raw	82	9	91
QC_C_12.raw	79	12	91
SP_6_2_C.raw	64	27	91
SP_6_4_C.raw	74	17	91
SP_6_3_C.raw	77	14	91
SP_6_1_C.raw	71	20	91
SP_6_L_1_C.raw	46	45	91
SP_6_L_3_C.raw	52	39	91
SP_6_L_4_C.raw	50	41	91
SP_6_L_2_C.raw	50	41	91

[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 :
    - Log transformation (base 10)
    - Square root transformation
    - 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

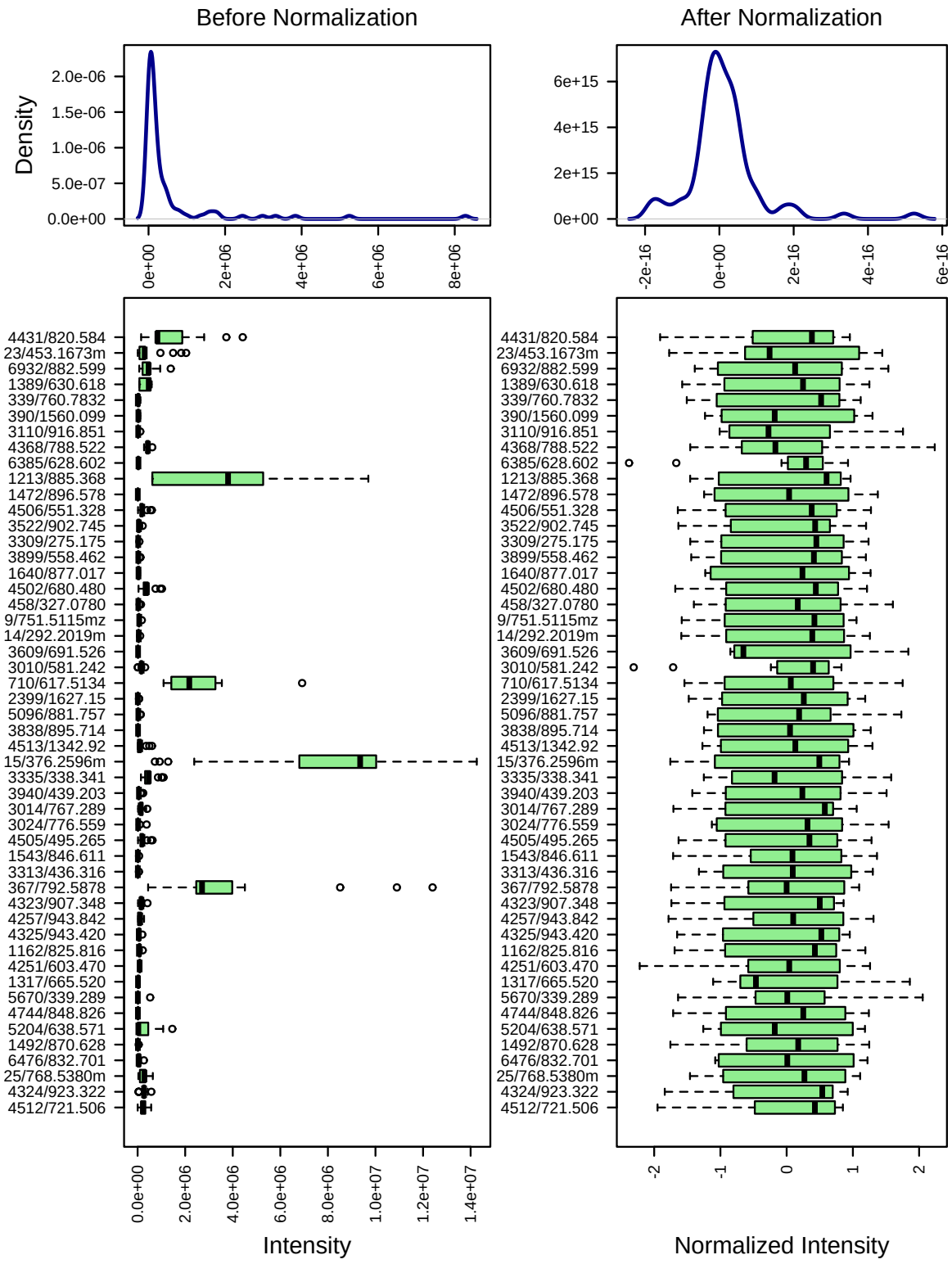

Figure 1: Box plots and kernel density plots before and after normalization. The boxplots show at most 50 features due to space limit. The density plots are based on all samples. Selected methods : Row-wise normalization: Normalization to constant sum; Data transformation: Log10 Normalization; Data scaling: 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 One-way ANOVA

Univariate analysis methods are the most common methods used for exploratory data analysis. For multi-group analysis, MetaboAnalyst provides one-way Analysis of Variance (ANOVA). As ANOVA only tells whether the overall comparison is significant or not, it is usually followed by post-hoc analyses in order to identify which two levels are different. MetaboAnalyst provides two most commonly used methods for this purpose - Fisher's least significant difference method (Fisher's LSD) and Tukey's Honestly Significant Difference (Tukey's HSD). The univariate analyses provide a preliminary overview about features that are potentially significant in discriminating the conditions under study.

Figure 2 shows the important features identified by ANOVA analysis. Table 2 shows the details of these features. The `post-hoc Sig. Comparison` column shows the comparisons between different levels that are significant given the p value threshold.

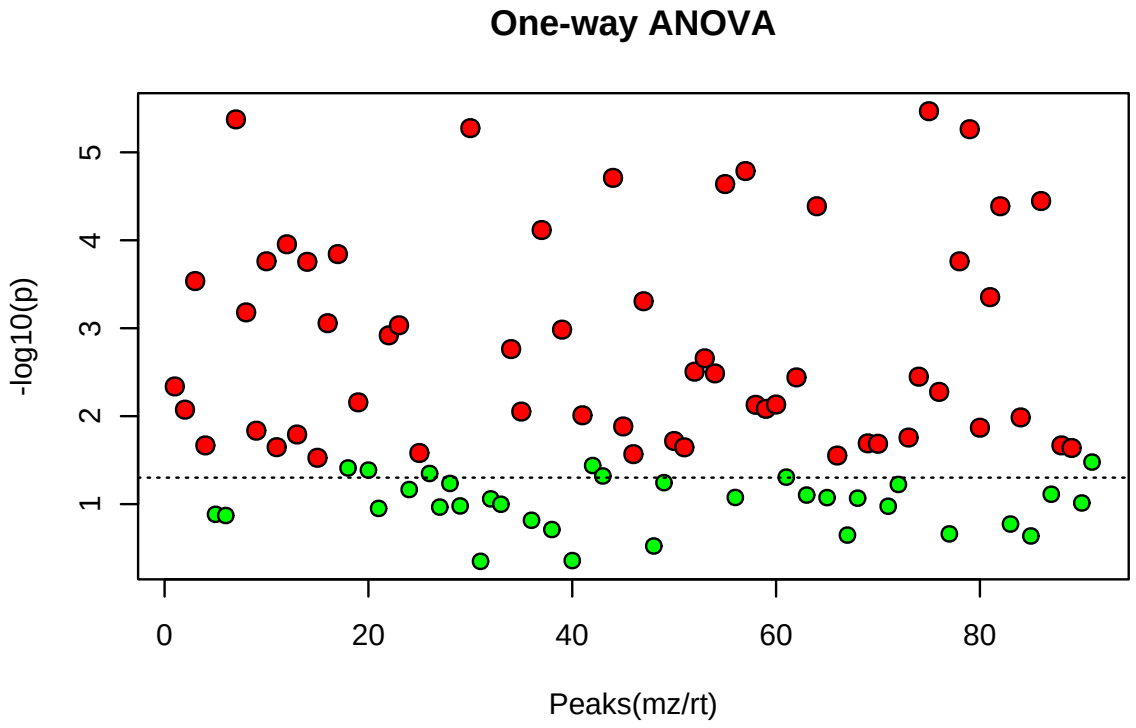

Figure 2: Important features selected by ANOVA plot with p value threshold 0.05.

Table 2: Top 50 features identified by One-way ANOVA and post-hoc analysis

	Peaks(mz/rt)	f.value	p.value	-log10(p)	FDR	Fisher's LSD
1	90/902.3935mz/4.32min	69.3300	3.4073e-06	5.4676	0.00012420	LA_6 - SP_6_L; SP_6 - SP_6_L
2	15/376.2596mz/0.72min	65.8920	4.2229e-06	5.3744	0.00012420	LA_6 - SP_6_L; SP_6 - SP_6_L
3	4612/647.4585mz/9.21min	62.4030	5.3082e-06	5.2751	0.00012420	LA_6 - SP_6_L; SP_6 - SP_6_L
4	4323/907.3484mz/4.31min	61.9870	5.4592e-06	5.2629	0.00012420	LA_6 - SP_6_L; SP_6 - SP_6_L
5	1162/825.8169mz/11.56min	47.6220	1.6325e-05	4.7871	0.00029599	LA_6 - SP_6_L; SP_6 - SP_6_L
6	3014/767.2898mz/4.34min	45.5950	1.9516e-05	4.7096	0.00029599	LA_6 - SP_6_L; SP_6 - SP_6_L
7	3333/807.3211mz/4.34min	43.8240	2.2947e-05	4.6393	0.00029831	LA_6 - SP_6_L; SP_6 - SP_6_L
8	4325/943.4203mz/4.31min	39.2790	3.5790e-05	4.4462	0.00037329	LA_6 - SP_6_L; SP_6 - SP_6_L
9	4356/869.3726mz/6.26min	37.9730	4.1015e-05	4.3871	0.00037329	SP_6 - LA_6; LA_6 - SP_6_L; SP_6 - SP_6_L
10	4324/923.3227mz/4.32min	37.9720	4.1021e-05	4.3870	0.00037329	LA_6 - SP_6_L; SP_6 - SP_6_L
11	4502/680.4800mz/7.91min	32.4830	7.6463e-05	4.1166	0.00063256	LA_6 - SP_6_L; SP_6 - SP_6_L
12	23/453.1673mz/0.78min	29.5410	1.1104e-04	3.9545	0.00084204	SP_6_L - LA_6; SP_6_L - SP_6
13	4506/551.3285mz/7.91min	27.6510	1.4358e-04	3.8429	0.00099936	LA_6 - SP_6_L; SP_6 - SP_6_L
14	1200/437.1933mz/0.74min	26.3360	1.7327e-04	3.7613	0.00099936	SP_6_L - LA_6; SP_6_L - SP_6
15	1774/904.7549mz/11.32min	26.3290	1.7343e-04	3.7609	0.00099936	LA_6 - SP_6; LA_6 - SP_6_L; SP_6_L - SP_6
16	4505/495.2658mz/7.91min	26.2400	1.7571e-04	3.7552	0.00099936	LA_6 - SP_6_L; SP_6 - SP_6_L
17	3304/311.1854mz/0.58min	22.9840	2.9080e-04	3.5364	0.00155660	SP_6_L - LA_6; SP_6_L - SP_6
18	3110/916.8512mz/9.76min	20.5300	4.4296e-04	3.3536	0.00223940	SP_6 - LA_6; SP_6 - SP_6_L
19	3024/776.5593mz/6.38min	19.9400	4.9314e-04	3.3070	0.00236190	SP_6 - LA_6; SP_6_L - LA_6; SP_6_L - SP_6
20	22/393.2858mz/0.71min	18.4040	6.6041e-04	3.1802	0.00300490	LA_6 - SP_6_L; SP_6 - SP_6_L
21	1330/548.5038mz/8.44min	17.0080	8.7656e-04	3.0572	0.00379840	SP_6_L - LA_6; SP_6_L - SP_6
22	4507/607.3911mz/7.89min	16.7490	9.2566e-04	3.0336	0.00382880	LA_6 - SP_6_L; SP_6 - SP_6_L
23	4512/721.5067mz/7.89min	16.2160	1.0378e-03	2.9839	0.00410620	LA_6 - SP_6_L; SP_6 - SP_6_L
24	4251/603.4708mz/10.95min	15.5430	1.2041e-03	2.9193	0.00456540	SP_6 - LA_6; SP_6_L - LA_6; SP_6_L - SP_6
25	4510/663.4537mz/7.88min	13.9990	1.7268e-03	2.7628	0.00628550	LA_6 - SP_6_L; SP_6 - SP_6_L
26	1114/797.7856mz/11.64min	13.0310	2.1997e-03	2.6576	0.00769880	LA_6 - SP_6_L; SP_6 - SP_6_L
27	1338/794.6870mz/8.82min	11.7230	3.1183e-03	2.5061	0.01051000	LA_6 - SP_6; SP_6_L - SP_6
28	219/806.6296mz/6.69min	11.5580	3.2644e-03	2.4862	0.01060900	SP_6_L - LA_6; SP_6_L - SP_6
29	5240/898.6011mz/10.20min	11.2560	3.5555e-03	2.4491	0.01098700	SP_6_L - LA_6; SP_6_L - SP_6
30	1657/853.8484mz/11.72min	11.1920	3.6221e-03	2.4410	0.01098700	LA_6 - SP_6_L; SP_6 - SP_6_L
31	3309/275.1753mz/0.75min	10.3870	4.5905e-03	2.3381	0.01347500	LA_6 - SP_6_L; SP_6 - SP_6_L
32	3522/902.7455mz/10.96min	9.9223	5.2942e-03	2.2762	0.01505500	SP_6 - LA_6; SP_6_L - LA_6
33	3899/558.4628mz/5.45min	9.0702	6.9634e-03	2.1572	0.01920200	LA_6 - SP_6_L; SP_6 - SP_6_L
34	1543/846.6118mz/10.13min	8.8996	7.3714e-03	2.1325	0.01931200	SP_6 - LA_6; SP_6_L - LA_6
35	6191/832.6828mz/6.77min	8.8769	7.4277e-03	2.1291	0.01931200	LA_6 - SP_6; SP_6_L - SP_6
36	6476/832.7010mz/8.92min	8.5659	8.2572e-03	2.0832	0.02073100	SP_6 - LA_6; SP_6_L - LA_6
37	14/292.2019mz/0.73min	8.5062	8.4291e-03	2.0742	0.02073100	LA_6 - SP_6_L; SP_6 - SP_6_L
38	1317/665.5204mz/7.54min	8.3610	8.8659e-03	2.0523	0.02123100	SP_6_L - LA_6; SP_6_L - SP_6
39	9/751.5115mz/0.73min	8.0877	9.7655e-03	2.0103	0.02278600	LA_6 - SP_6_L; SP_6 - SP_6_L
40	3749/927.4253mz/6.19min	7.9312	1.0331e-02	1.9859	0.02350300	SP_6 - LA_6; SP_6 - SP_6_L
41	25/768.5380mz/0.74min	7.2953	1.3085e-02	1.8832	0.02904200	LA_6 - SP_6_L; SP_6 - SP_6_L
42	5100/907.7727mz/11.89min	7.2076	1.3532e-02	1.8687	0.02931800	SP_6 - LA_6
43	3313/436.3166mz/0.72min	7.0090	1.4614e-02	1.8352	0.03092800	LA_6 - SP_6_L; SP_6 - SP_6_L
44	5301/473.2646mz/0.79min	6.7559	1.6153e-02	1.7918	0.03340700	SP_6_L - LA_6; SP_6_L - SP_6
45	1472/896.5787mz/10.07min	6.5571	1.7501e-02	1.7569	0.03539200	SP_6 - LA_6
46	367/792.5878mz/7.16min	6.3376	1.9154e-02	1.7177	0.03789100	SP_6_L - LA_6; SP_6_L - SP_6
47	6932/882.5998mz/6.74min	6.1905	2.0369e-02	1.6910	0.03904300	LA_6 - SP_6
48	1213/885.3683mz/4.37min	6.1644	2.0594e-02	1.6863	0.03904300	LA_6 - SP_6_L; SP_6 - SP_6_L
49	4513/1342.9265mz/7.91min	6.0641	2.1488e-02	1.6678	0.03913400	LA_6 - SP_6_L; SP_6 - SP_6_L
50	458/327.0780mz/7.92min	6.0626	2.1502e-02	1.6675	0.03913400	LA_6 - SP_6_L; SP_6 - SP_6_L

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 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 scores plot between selected PCs; Figure 6 shows the 3-D scores plot between selected PCs; Figure 7 shows the loadings plot between the selected PCs; Figure 8 shows the biplot between the selected PCs.

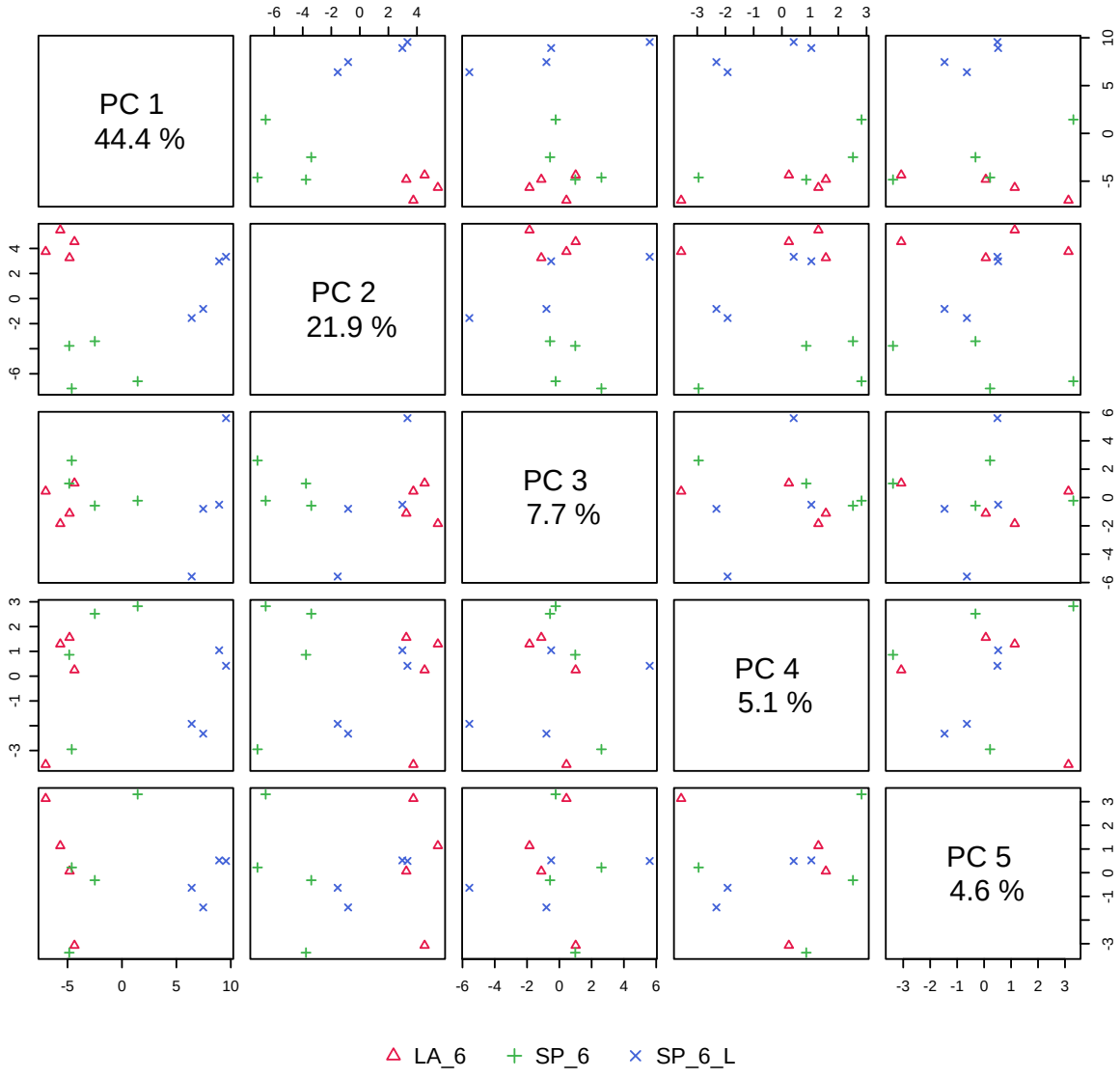

Figure 3: Pairwise score plots between the selected PCs. The explained variance of each PC is shown in the corresponding diagonal cell.

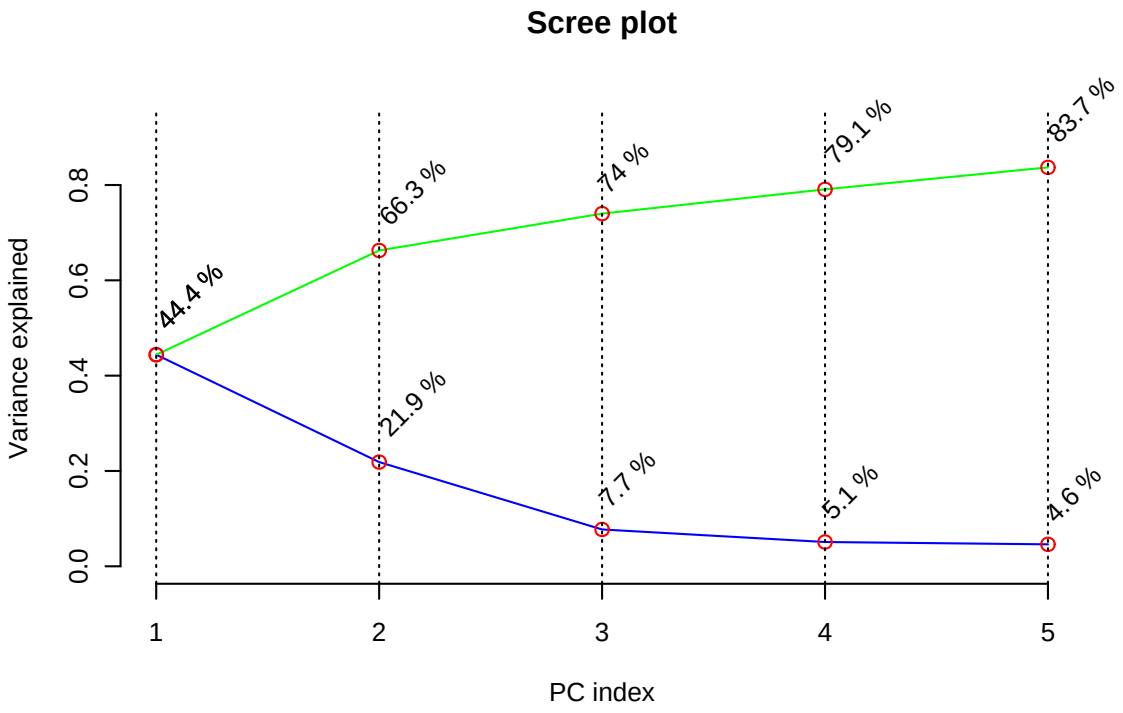

Figure 4: Scree plot shows the variance explained by PCs. The green line on top shows the accumulated variance explained; the blue line underneath shows the variance explained by individual PC.

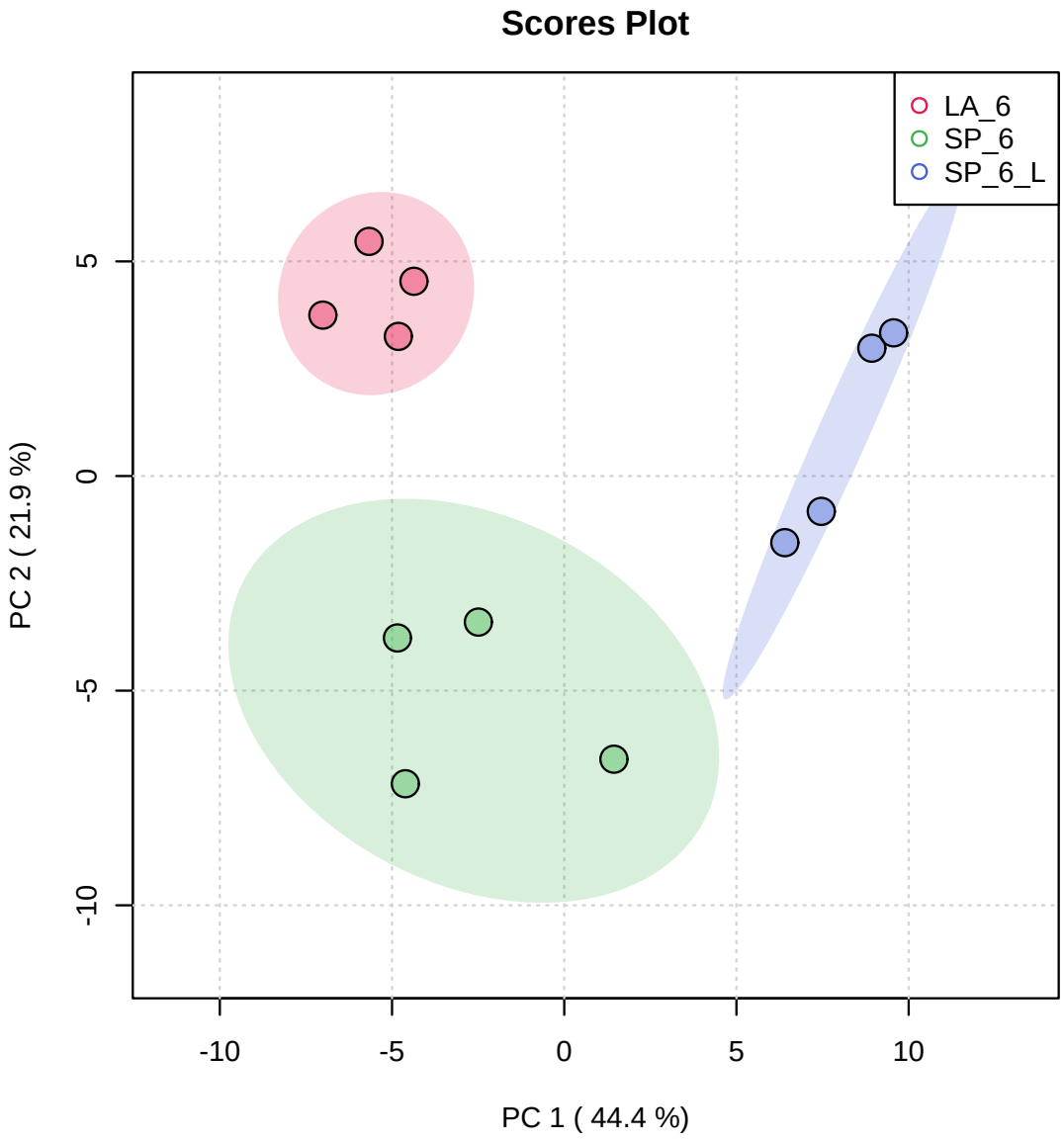

Figure 5: Scores plot between the selected PCs. The explained variances are shown in brackets.

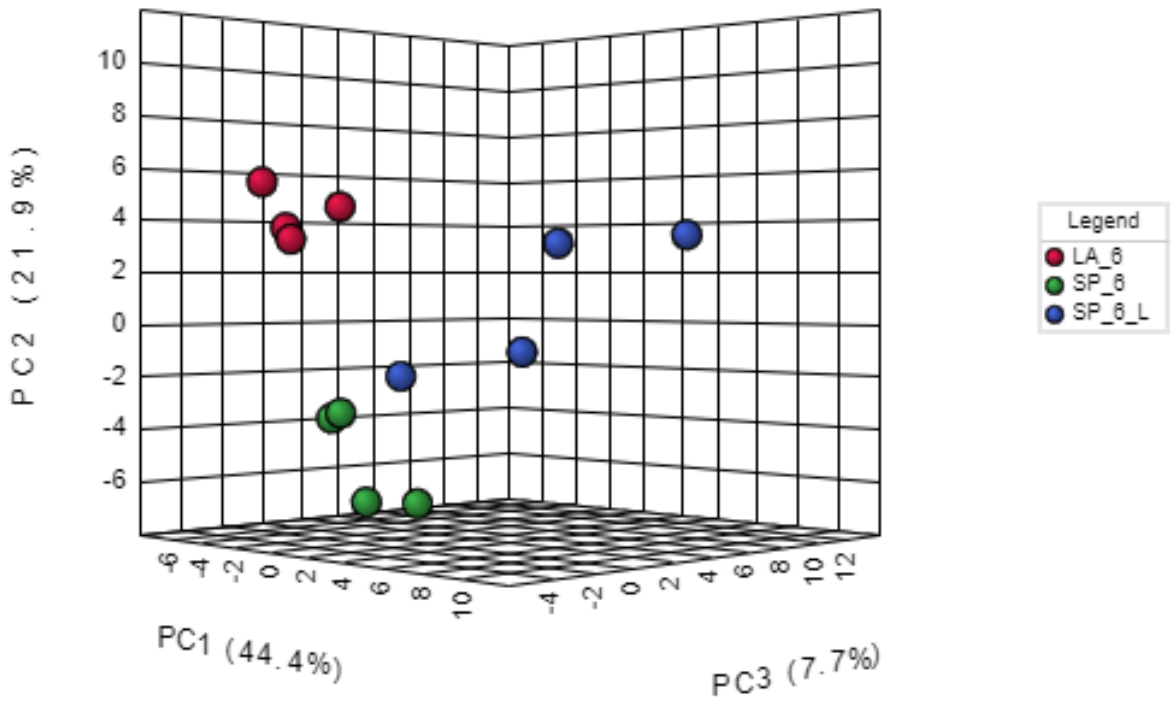

Figure 6: 3D score plot between the selected PCs. The explained variances are shown in brackets.

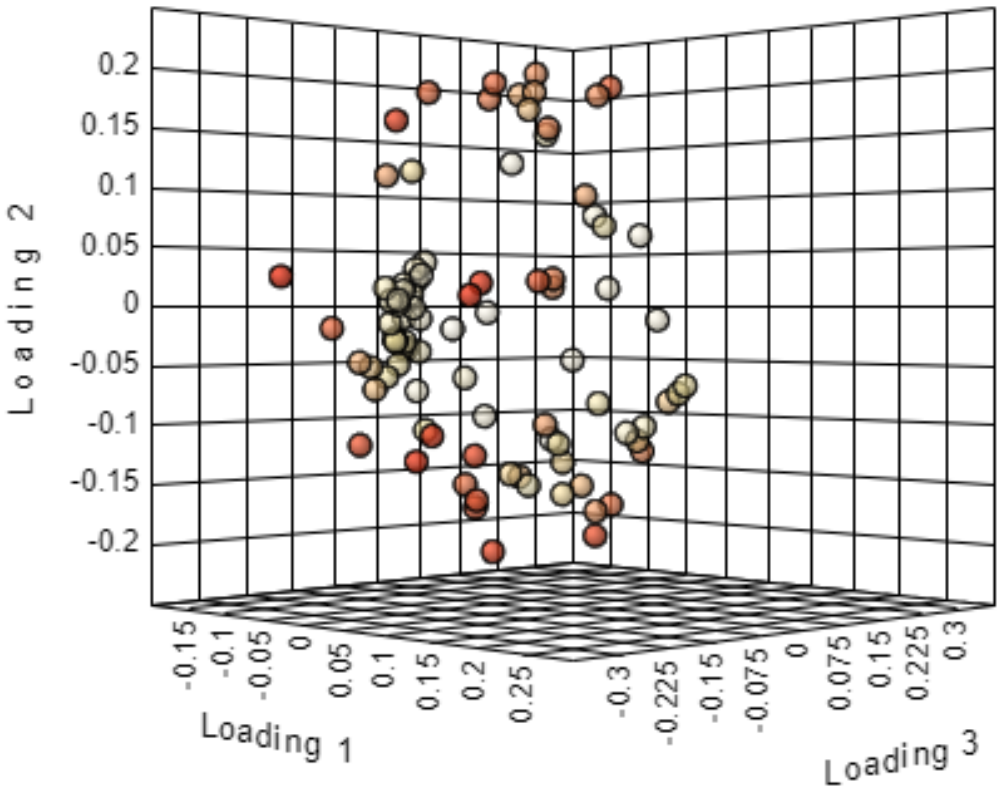

Figure 7: Loadings plot for the selected PCs.

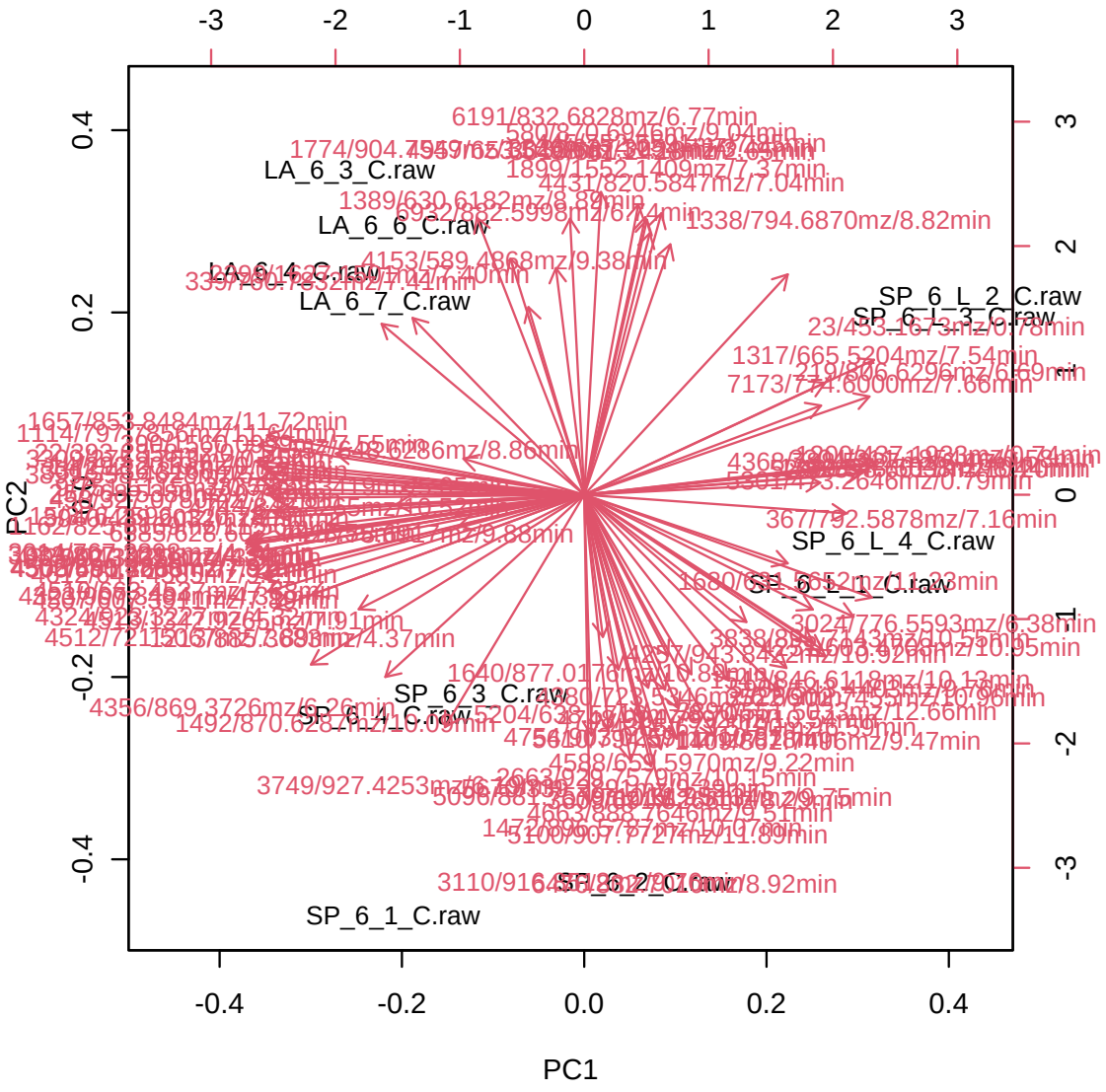

Figure 8: PCA biplot between the selected PCs. Note, you may want to test different centering and scaling normalization methods for the biplot to be displayed properly.

2.3 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 9 shows the overview of scores plots; Figure 10 shows the 2-D scores plot between selected components; Figure 11 shows the 3-D scores 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 results of permutation test for model validation; Figure 15 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

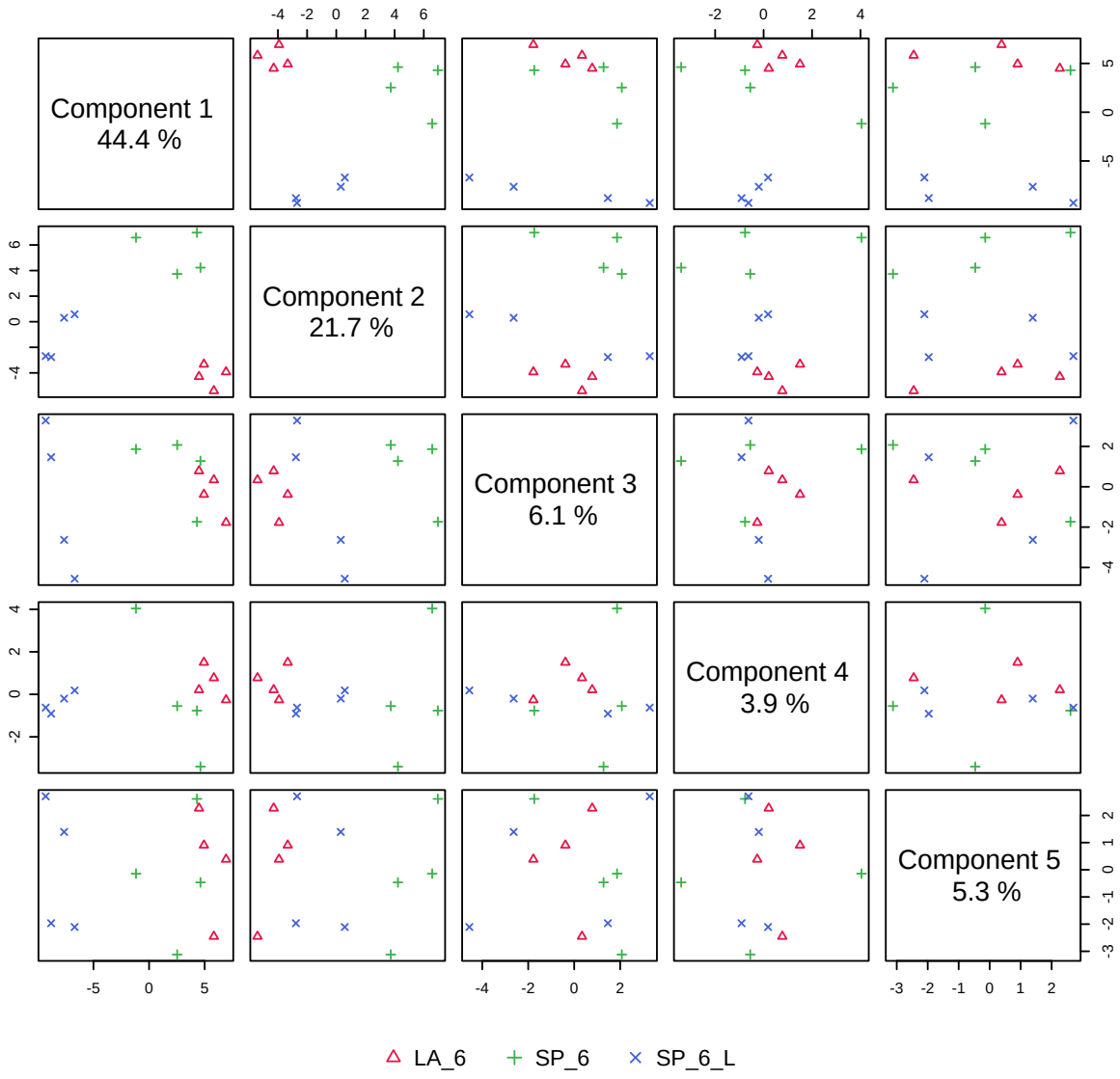

Figure 9: Pairwise scores plots between the selected components. The explained variance of each component is shown in the corresponding diagonal cell.

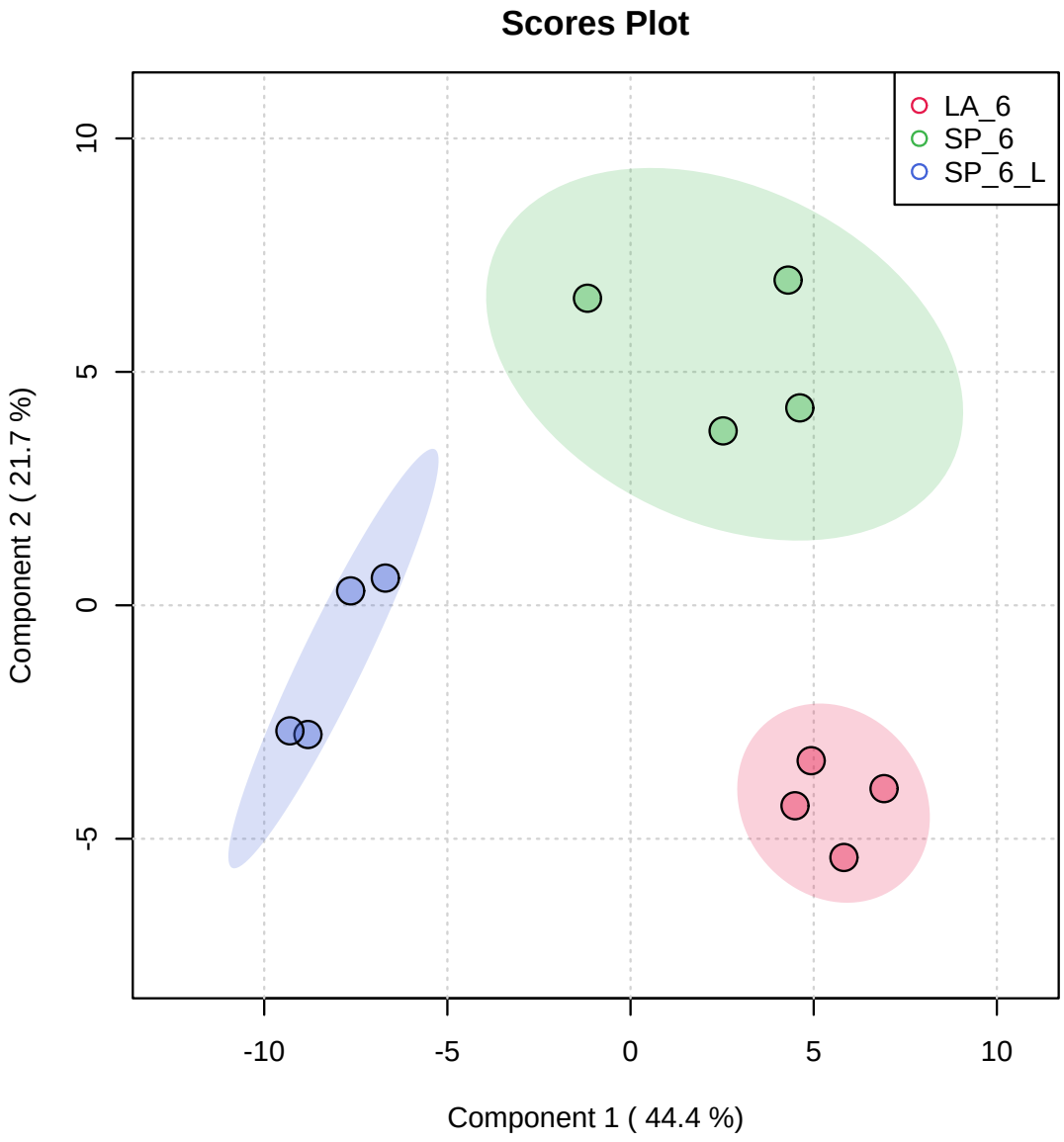

Figure 10: Scores plot between the selected PCs. The explained variances are shown in brackets.

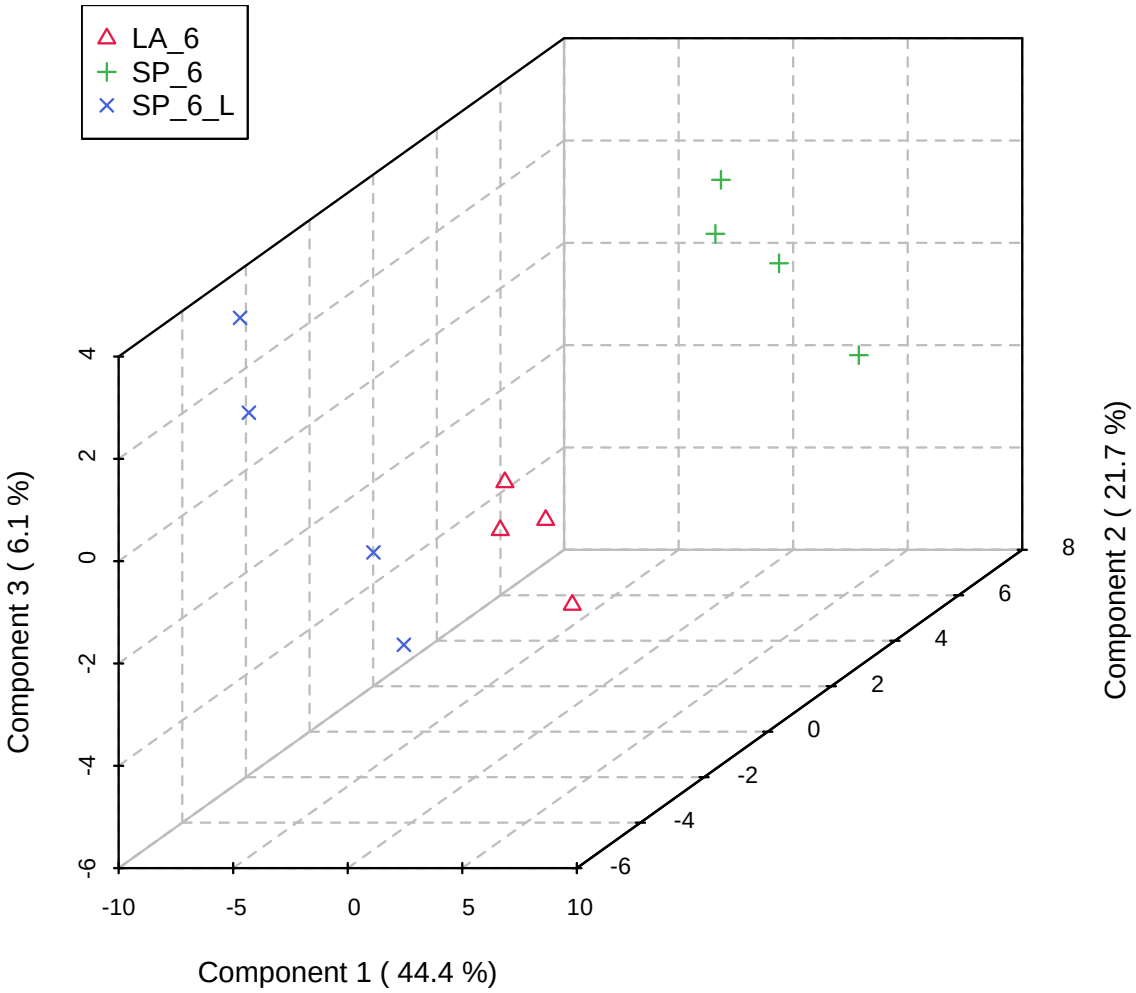

Figure 11: 3D scores plot between the selected PCs. The explained variances are shown in brackets.

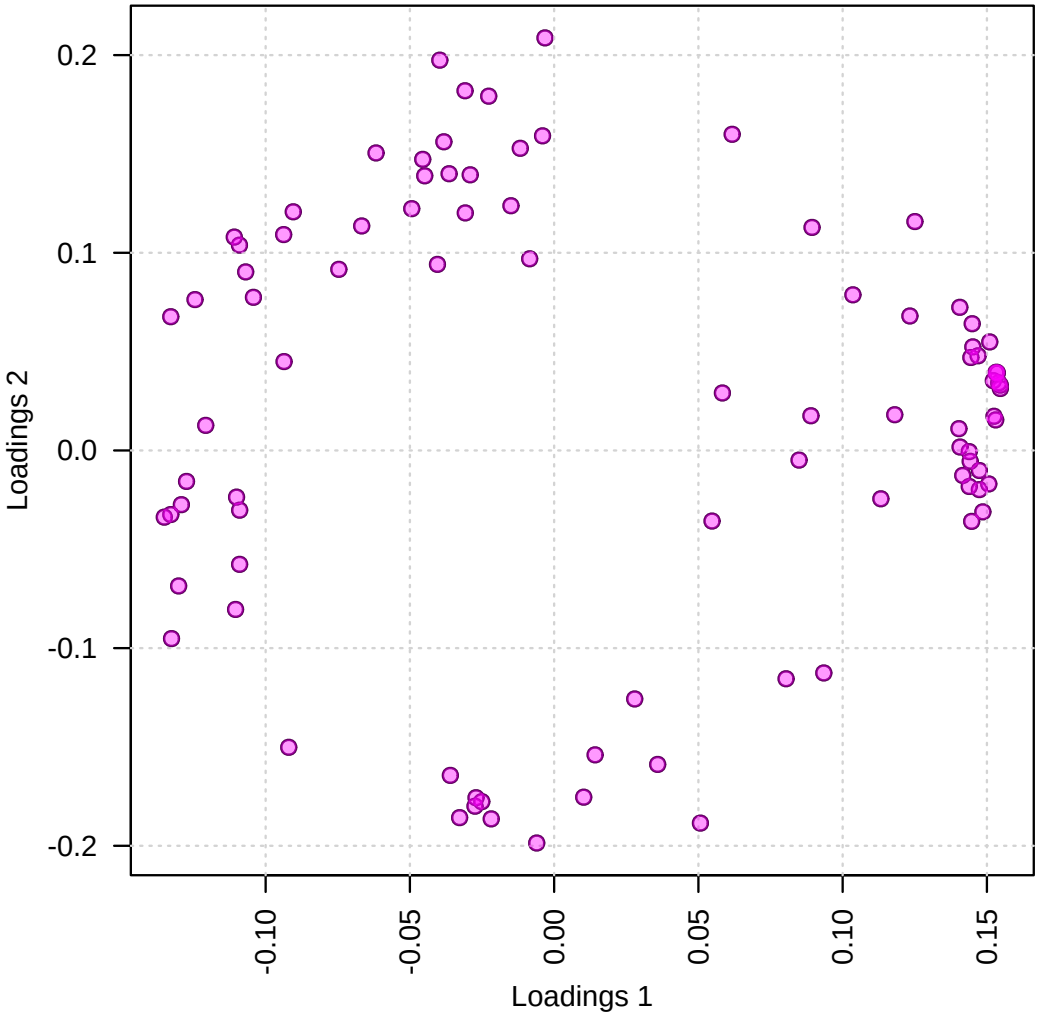

Figure 12: Loadings plot between the selected PCs.

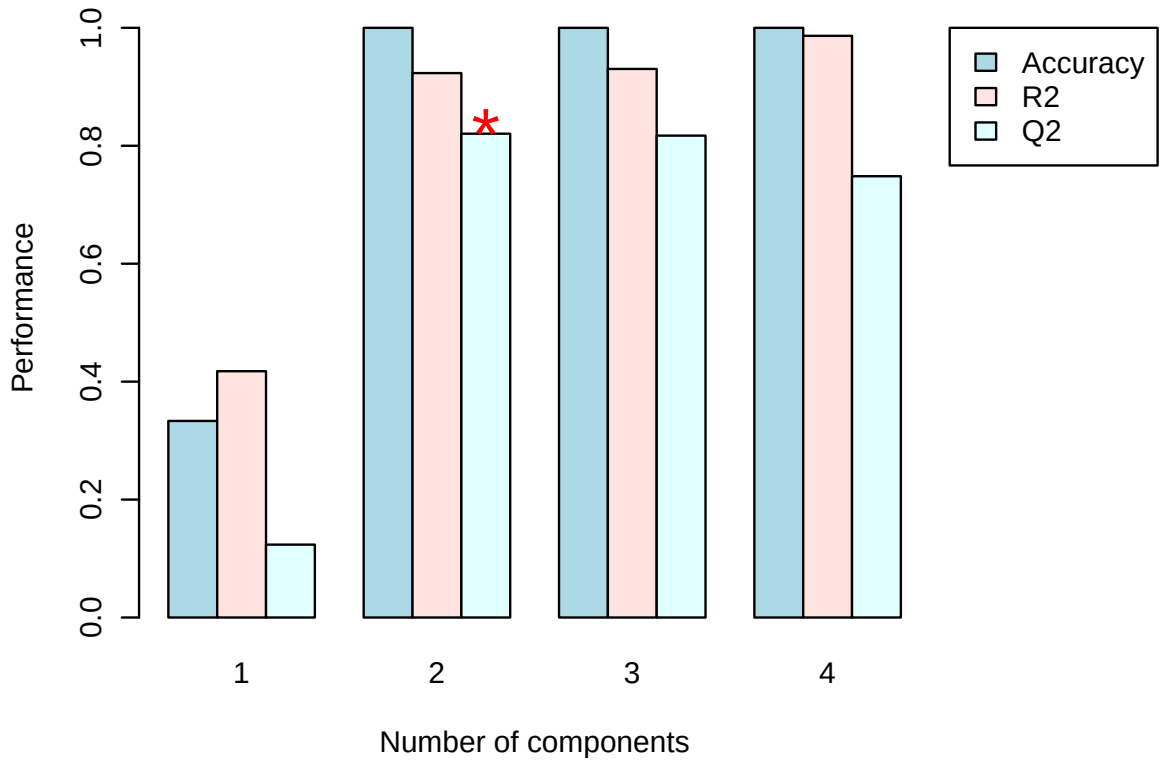

Figure 13: PLS-DA classification using different number of components. The red star indicates the best classifier.

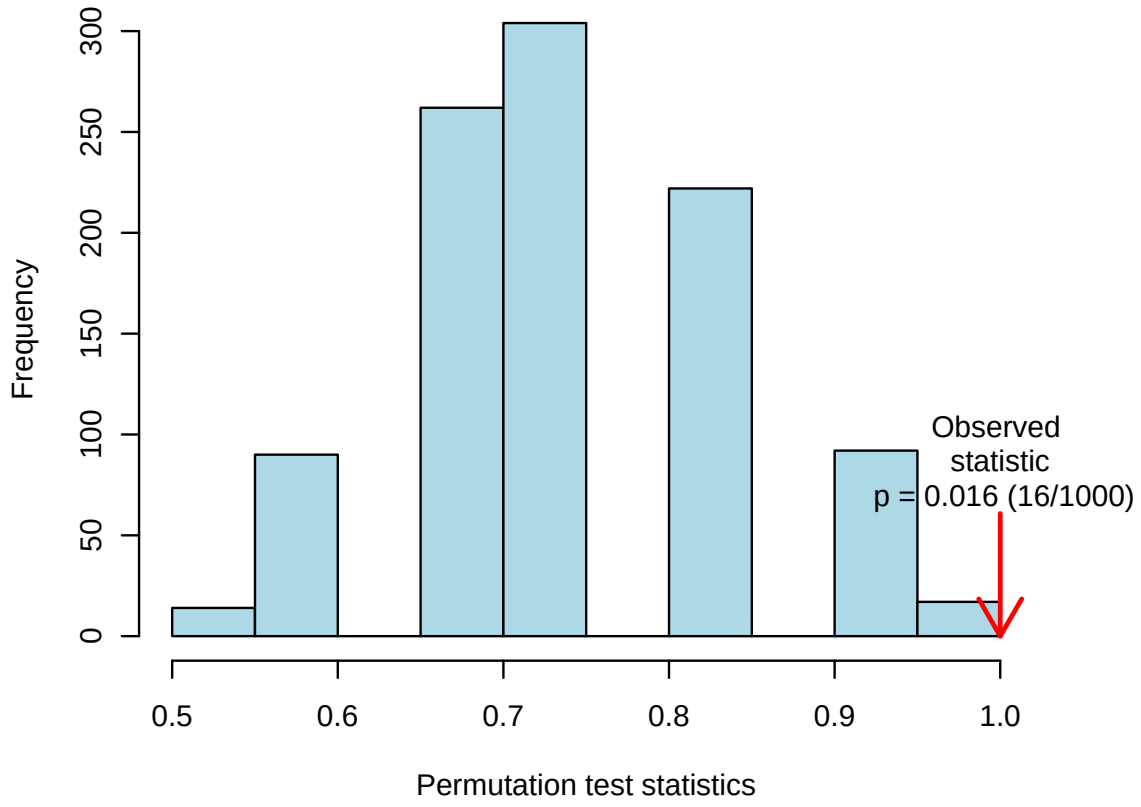

Figure 14: PLS-DA model validation by permutation tests based on prediction accuracy. The p value based on permutation is p = 0.016 (16/1000).

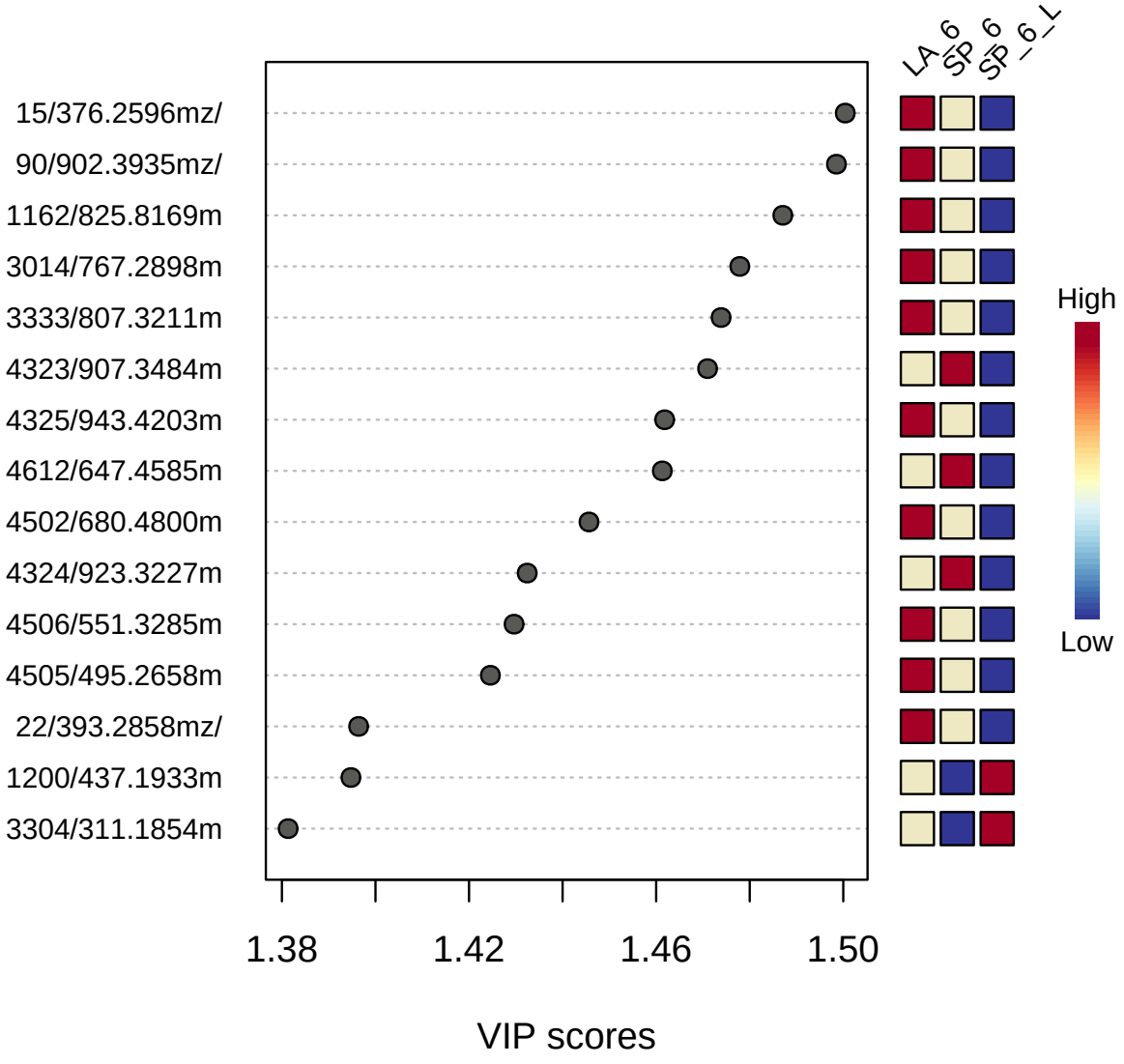

Figure 15: Important features identified by PLS-DA. The colored boxes on the right indicate the relative concentrations of the corresponding metabolite in each group under study.

2.4 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`. Figure 16 shows the clustering result in the form of a dendrogram. Figure 17 shows the clustering result in the form of a heatmap.

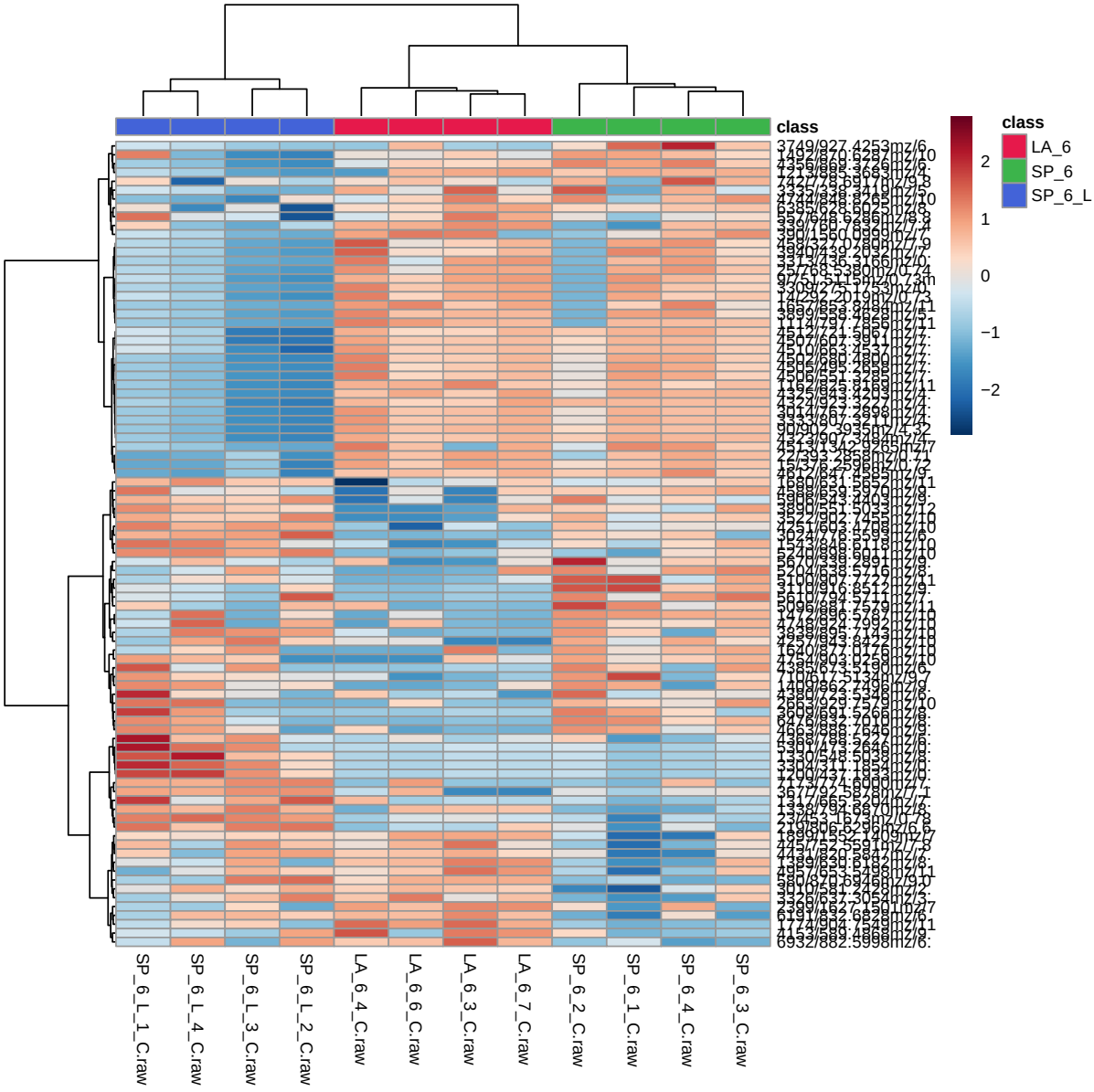

Figure 16: Clustering result shown as heatmap (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

3 Appendix: R Command History

```
 [1] "mSet<-InitDataObjects(\"pktable\", \"stat\", FALSE)"
 [2] "mSet<-Read.TextData(mSet, \"Replacing_with_your_file_path\", \"colu\", \"disc\");"
 [3] "mSet<-SanityCheckData(mSet)"
 [4] "mSet<-SanityCheckData(mSet)"
 [5] "mSet<-ImputeMissingVar(mSet, method=\"min\")"
 [6] "mSet<-SanityCheckData(mSet)"
 [7] "mSet<-FilterVariable(mSet, \"none\", \"F\", 25)"
 [8] "mSet<-PreparePrenormData(mSet)"
 [9] "mSet<-Normalization(mSet, \"SumNorm\", \"LogNorm\", \"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<-PCA.Anal(mSet)"
[13] "mSet<-PlotPCAPairSummary(mSet, \"pca_pair_0_\", \"png\", 72, width=NA, 5)"
[14] "mSet<-PlotPCAScree(mSet, \"pca_scree_0_\", \"png\", 72, width=NA, 5)"
[15] "mSet<-PlotPCA2DScore(mSet, \"pca_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0)"
[16] "mSet<-PlotPCALoading(mSet, \"pca_loading_0_\", \"png\", 72, width=NA, 1,2);"
[17] "mSet<-PlotPCABiplot(mSet, \"pca_biplot_0_\", \"png\", 72, width=NA, 1,2)"
[18] "mSet<-PlotPCA3DLoading(mSet, \"pca_loading3d_0_\", \"json\", 1,2,3)"
[19] "mSet<-GetGroupNames(mSet, \"null\")"
[20] "mSet<-PlotPCA2DScore(mSet, \"pca_score2d_0_\", \"png\", 300, width=NA, 1,2,0.95,0,0)"
[21] "mSet<-GetGroupNames(mSet, \"\")"
[22] "feature.nm.vec <- c(\"\")"
[23] "smpl.nm.vec <- c(\"QC_C_4.raw\",\"QC_C_5.raw\",\"QC_C_9.raw\",\"QC_C_8.raw\",\"QC_C_6.raw\",\"
[24] "grp.nm.vec <- c(\"\")"
[25] "mSet<-UpdateData(mSet)"
[26] "mSet<-PreparePrenormData(mSet)"
[27] "mSet<-Normalization(mSet, \"SumNorm\", \"LogNorm\", \"AutoNorm\", ratio=FALSE, ratioNum=20)"
[28] "mSet<-PlotNormSummary(mSet, \"norm_1_\", \"png\", 72, width=NA)"
[29] "mSet<-PlotSampleNormSummary(mSet, \"snorm_1_\", \"png\", 72, width=NA)"
[30] "mSet<-PCA.Anal(mSet)"
[31] "mSet<-PlotPCAPairSummary(mSet, \"pca_pair_0_\", \"png\", 72, width=NA, 5)"
[32] "mSet<-PlotPCAScree(mSet, \"pca_scree_0_\", \"png\", 72, width=NA, 5)"
[33] "mSet<-PlotPCA2DScore(mSet, \"pca_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0)"
[34] "mSet<-PlotPCALoading(mSet, \"pca_loading_0_\", \"png\", 72, width=NA, 1,2);"
[35] "mSet<-PlotPCABiplot(mSet, \"pca_biplot_0_\", \"png\", 72, width=NA, 1,2)"
[36] "mSet<-PlotPCA3DLoading(mSet, \"pca_loading3d_0_\", \"json\", 1,2,3)"
[37] "mSet<-PLSR.Anal(mSet, reg=TRUE)"
[38] "mSet<-PlotPLSPairSummary(mSet, \"pls_pair_0_\", \"png\", 72, width=NA, 5)"
[39] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0)"
[40] "mSet<-PlotPLS3DScoreImg(mSet, \"pls_score3d_0_\", \"png\", 72, width=NA, 1,2,3, 40)"
[41] "mSet<-PlotPLSLoading(mSet, \"pls_loading_0_\", \"png\", 72, width=NA, 1, 2);"
[42] "mSet<-PlotPLS3DLoading(mSet, \"pls_loading3d_0_\", \"json\", 1,2,3)"
[43] "mSet<-PLSDA.CV(mSet, \"L\",4, \"Q2\")"
[44] "mSet<-PlotPLS.Classification(mSet, \"pls_cv_0_\", \"png\", 72, width=NA)"
[45] "mSet<-PlotPLS.Imp(mSet, \"pls_imp_0_\", \"png\", 72, width=NA, \"vip\", \"Comp. 1\", 15,FALSE)
[46] "mSet<-PLSR.Anal(mSet, reg=FALSE)"
[47] "mSet<-PlotPLSPairSummary(mSet, \"pls_pair_1_\", \"png\", 72, width=NA, 5)"
[48] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_1_\", \"png\", 72, width=NA, 1,2,0.95,1,0)"
[49] "mSet<-PlotPLS3DScoreImg(mSet, \"pls_score3d_1_\", \"png\", 72, width=NA, 1,2,3, 40)"
[50] "mSet<-PlotPLSLoading(mSet, \"pls_loading_1_\", \"png\", 72, width=NA, 1, 2);"
[51] "mSet<-PLSDA.CV(mSet, \"L\",4, \"Q2\")"
[52] "mSet<-PlotPLS.Classification(mSet, \"pls_cv_1_\", \"png\", 72, width=NA)"
[53] "mSet<-PlotPLS.Imp(mSet, \"pls_imp_1_\", \"png\", 72, width=NA, \"vip\", \"Comp. 1\", 15,FALSE)
[54] "mSet<-PLSDA.Permut(mSet, 1000, \"accu\")"
[55] "mSet<-PlotPLS.Permutation(mSet, \"pls_perm_1_\", \"png\", 72, width=NA)"
[56] "mSet<-ANOVA.Anal(mSet, F, 0.05, \"fisher\", FALSE)"
```

```
[57] "mSet<-PlotANOVA(mSet, \"aov_0_\", \"png\", 72, width=NA)"
[58] "mSet<-PlotHeatMap(mSet, \"heatmap_0_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\
[59] "mSet<-GetGroupNames(mSet, \"null\")"
[60] "mSet<-PlotPCA2DScore(mSet, \"pca_score2d_0_\", \"png\", 300, width=NA, 1,2,0.95,0,0)"
[61] "mSet<-UpdatePCA.Loading(mSet, \"none\");"
[62] "mSet<-PlotPCALoading(mSet, \"pca_loading_1_\", \"png\", 72, width=NA, 1,2);"
[63] "mSet<-PlotPCALoading(mSet, \"pca_loading_1_\", \"png\", 300, width=NA, 1,2);"
[64] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_1_\", \"png\", 300, width=NA, 1,2,0.95,1,0)"
[65] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_2_\", \"png\", 72, width=NA, 1,2,0.95,0,0)"
[66] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_2_\", \"png\", 300, width=NA, 1,2,0.95,0,0)"
[67] "mSet<-UpdatePLS.Loading(mSet, \"none\");"
[68] "mSet<-PlotPLSLoading(mSet, \"pls_loading_2_\", \"png\", 72, width=NA, 1, 2);"
[69] "mSet<-PlotPLSLoading(mSet, \"pls_loading_2_\", \"png\", 300, width=NA, 1, 2);"
[70] "mSet<-PlotPLS.Classification(mSet, \"pls_cv_1_\", \"png\", 300, width=NA)"
[71] "mSet<-PLSDA.Permut(mSet, 1000, \"accu\")"
[72] "mSet<-PlotPLS.Permutation(mSet, \"pls_perm_2_\", \"png\", 72, width=NA)"
[73] "mSet<-PlotPLS.Permutation(mSet, \"pls_perm_2_\", \"png\", 300, width=NA)"
[74] "mSet<-SaveTransformedData(mSet)"
[75] "mSet<-PreparePDFReport(mSet, \"guest14509913071070504278\")\n"
```

The report was generated on Fri Jul 15 10:46:45 2022 with R version 4.1.3 (2022-03-10), OS system: Linux, version: 30 20.04.1-Ubuntu SMP Tue Apr 26 03:01:25 UTC 2022 .