Metabolomic Data Analysis with MetaboAnalyst 6.0

Name: guest9353209576210757826

March 12, 2024

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. 1 empty labels were detected and excluded from your data. The uploaded data file contains 49 (samples) by 2182 (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.

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.

[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

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

Table 1: Summary of data processing results

	Features (positive)	Missing/Zero	Features (processed)
X16.C9.3.neg	2124	58	2182
X19.C9.2.neg	1951	231	2182
X21.C9.4.neg	2075	107	2182
X45.C9.1.neg	2117	65	2182
X13.D9.4.neg	2149	33	2182
X20.D9.1.neg	2055	127	2182
X49.D9.2.neg	2122	60	2182
X48.D9.3.neg	2145	37	2182
X26.F9.1.neg	2126	56	2182
X34.F9.2.neg	924	1258	2182
X36.F9.3.neg	2112	70	2182
X46.F9.4.neg	861	1321	2182
X14.X9.2.neg	2151	31	2182
X23.X9.1.neg	2125	57	2182
X27.X9.3.neg	2133	49	2182
X40.X9.4.neg	2148	34	2182
X53.Blank.neg	1051	1131	2182
X10.QC1.neg	2165	17	2182
X24.QC.2.neg	2174	8	2182
X39.QC3.neg	2151	31	2182
X02.C12.2.neg	2097	85	2182
X18.C12.3.neg	2111	71	2182
X33.C12.4.neg	2140	42	2182
X12.D12.2.neg	2158	24	2182
X28.D12.3.neg	2129	53	2182
X44.D12.1.neg	792	1390	2182
X52.D12.4.neg	2144	38	2182
X05.F12.4.neg	2149	33	2182
X22.F12.1.neg	2071	111	2182
X38.F12.2.neg	2130	52	2182
X43.F12.3.neg	2090	92	2182
X09.X12.3.neg	796	1386	2182
X32.X12.2.neg	2117	65	2182
X41.X12.1.neg	2148	34	2182
X31.C15.3.neg	2120	62	2182
X47.C15.1.neg	2127	55	2182
X50.C15.2.neg	2085	97	2182
X04.D15.4.neg	755	1427	2182
X06.D15.2.neg	2096	86	2182
X35.D15.1.neg	844	1338	2182
X37.D15.3.neg	881	1301	2182
X17.F15.3.neg	834	1348	2182
X25.F15.1.neg	838	1344	2182
X30.F15.2.neg	1909	273	2182
X51.F15.4.neg	2053	129	2182
X08.X15.4.neg	2139	43	2182
X11.X15.3.neg	2124	58	2182
X29.X15.1.neg	2099	83	2182
X42.X15.2.neg	847	1335	2182

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.

Row-wise normalization: Normalization by a reference feature; Data transformation: Log10 Normalization; Data scaling: Pareto Scaling.

[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

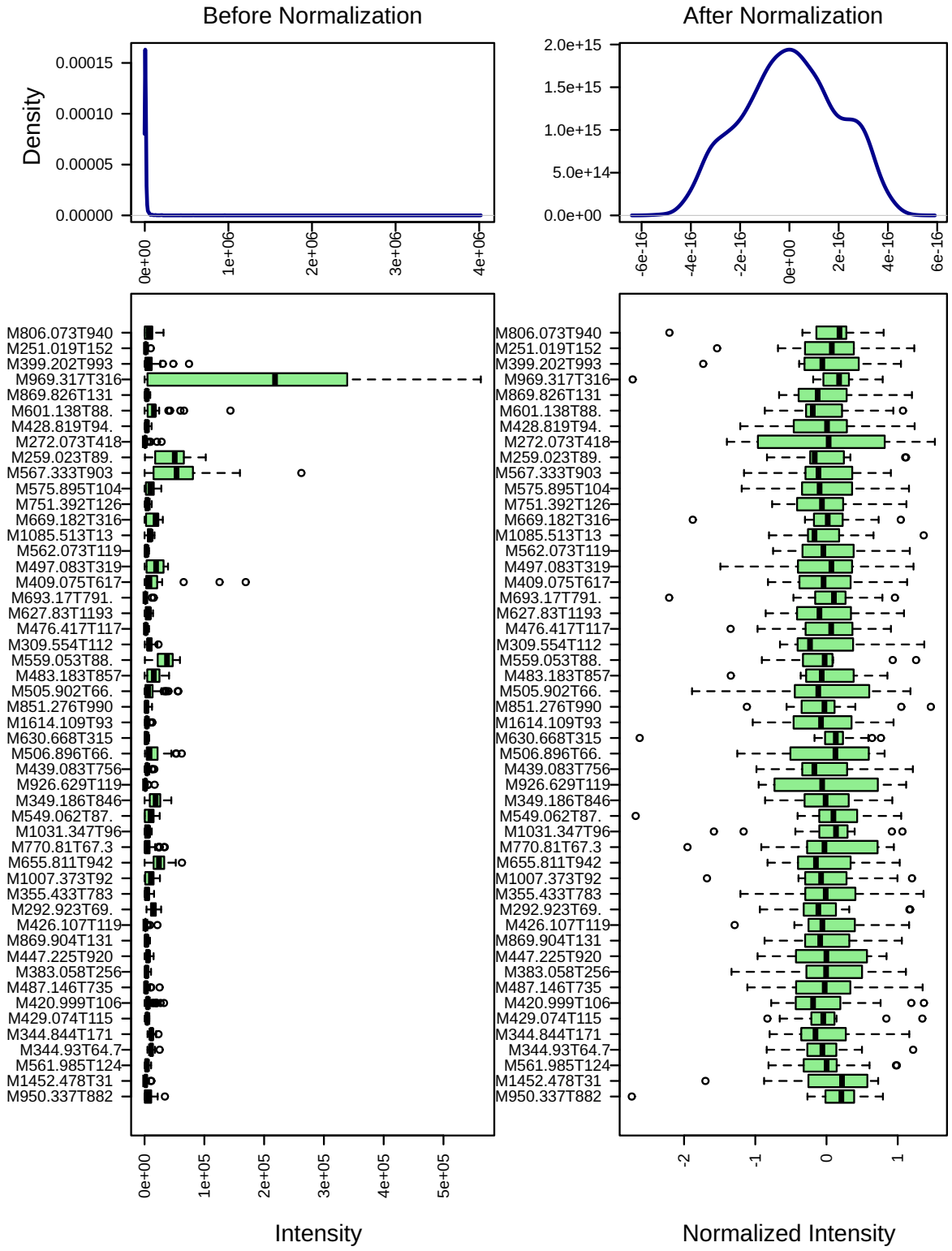

Figure 1: Box plots and kernel density plots before and after normalization. The boxplots show at most 50 features due to space limit. The density plots are based on all samples.

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.

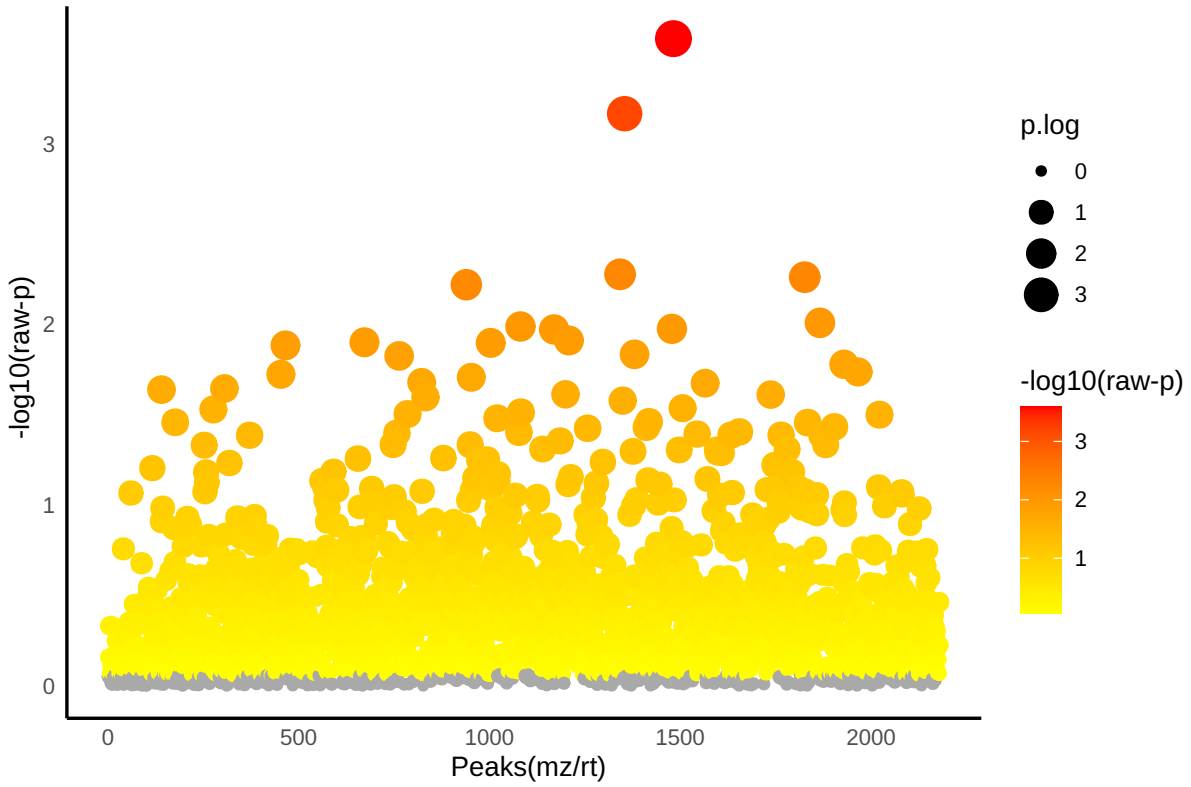

Figure 2: Important features selected by ANOVA plot with p value threshold 0.99435.

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

	Peaks(mz/rt)	F.stat	p.value	-log10(p)	FDR
1	M283.086T201.788	17.56	0.00026053	3.5841	0.56796
2	M315.405T313.892	13.863	0.00067835	3.1685	0.7394
3	M1712.298T909.092	7.9626	0.005258	2.2792	0.99434
4	M374.109T731.372	7.8777	0.0054535	2.2633	0.99434
5	M698.888T1003.439	7.6562	0.0060055	2.2214	0.99434
6	M1231.291T836.99	6.6024	0.0097604	2.0105	0.99434
7	M449.004T832.738	6.5065	0.010226	1.9903	0.99434
8	M698.86T68.208	6.4448	0.010539	1.9772	0.99434
9	M1130.437T924.326	6.4295	0.010619	1.9739	0.99434
10	M433.038T831.69	6.1472	0.012223	1.9128	0.99434
11	M410.078T615.621	6.102	0.012506	1.9029	0.99434
12	M272.073T418.674	6.0867	0.012604	1.8995	0.99434
13	M204.085T413.719	6.0235	0.013016	1.8855	0.99434
14	M469.036T876.272	5.803	0.01459	1.836	0.99434
15	M642.302T1183.207	5.765	0.014883	1.8273	0.99434
16	M637.177T857.418	5.5643	0.016555	1.7811	0.99434
17	M1168.035T908.053	5.3831	0.018261	1.7385	0.99434
18	M380.071T778.258	5.3284	0.018817	1.7254	0.99434
19	M246.097T864.37	5.2595	0.019547	1.7089	0.99434
20	M857.325T1034.658	5.1435	0.020852	1.6808	0.99434
21	M975.202T831.412	5.125	0.021069	1.6763	0.99434
22	M404.065T806.181	5.005	0.022549	1.6469	0.99434
23	M453.17T879.981	4.98	0.022873	1.6407	0.99434
24	M516.992T825.04	4.8757	0.024285	1.6147	0.99434
25	M401.027T415.826	4.8693	0.024375	1.6131	0.99434
26	M365.052T459.24	4.8124	0.025192	1.5987	0.99434
27	M1197.439T886.215	4.7375	0.02632	1.5797	0.99434
28	M539.034T616.33	4.5749	0.028979	1.5379	0.99434
29	M790.016T1120.766	4.5477	0.029456	1.5308	0.99434
30	M435.234T1304.756	4.4859	0.030572	1.5147	0.99434
31	M687.305T1181.604	4.4509	0.031226	1.5055	0.99434
32	M541.425T1223.783	4.4371	0.031488	1.5019	0.99434
33	M1176.528T983.06	4.3621	0.032963	1.482	0.99434
34	M721.9T1109.959	4.2885	0.03449	1.4623	0.99434
35	M781.218T930.113	4.279	0.034693	1.4598	0.99434
36	M1042.847T909.803	4.2719	0.034844	1.4579	0.99434
37	M319.076T917.392	4.185	0.03678	1.4344	0.99434
38	M1130.433T884.599	4.1773	0.036958	1.4323	0.99434
39	M271.07T418.897	4.1567	0.03744	1.4267	0.99434
40	M338.196T1226.316	4.0831	0.039217	1.4065	0.99434
41	M459.275T1179.554	4.074	0.039444	1.404	0.99434
42	M609.777T834.854	4.0575	0.03986	1.3995	0.99434
43	M463.02T875.074	4.0389	0.040334	1.3943	0.99434
44	M365.135T389.484	4.0182	0.04087	1.3886	0.99434
45	M469.019T418.674	4.0176	0.040886	1.3884	0.99434
46	M203.083T418.775	4.0173	0.040892	1.3884	0.99434
47	M861.253T1033.436	4.0161	0.040924	1.388	0.99434
48	M375.129T729.569	3.9042	0.043976	1.3568	0.99434
49	M387.034T831.689	3.8311	0.046117	1.3361	0.99434
50	M339.058T415.891	3.8294	0.046168	1.3357	0.99434

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 biplot between the selected PCs. Interactive 3-D scores plots are not included here and can be directly downloaded from website.

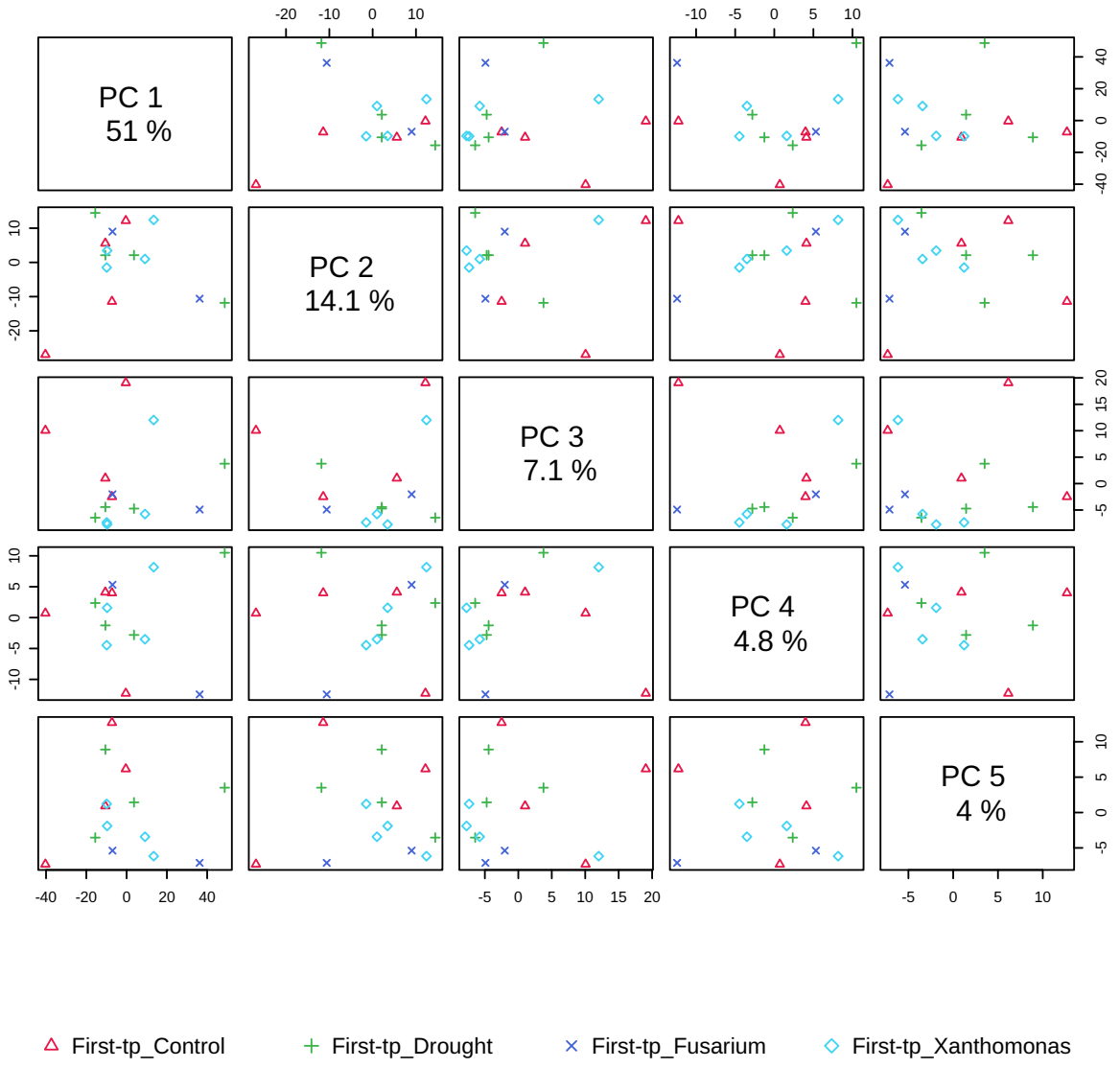

Figure 3: Pairwise score plots between the selected PCs. The explained variance of each PC is shown in the corresponding diagonal cell.

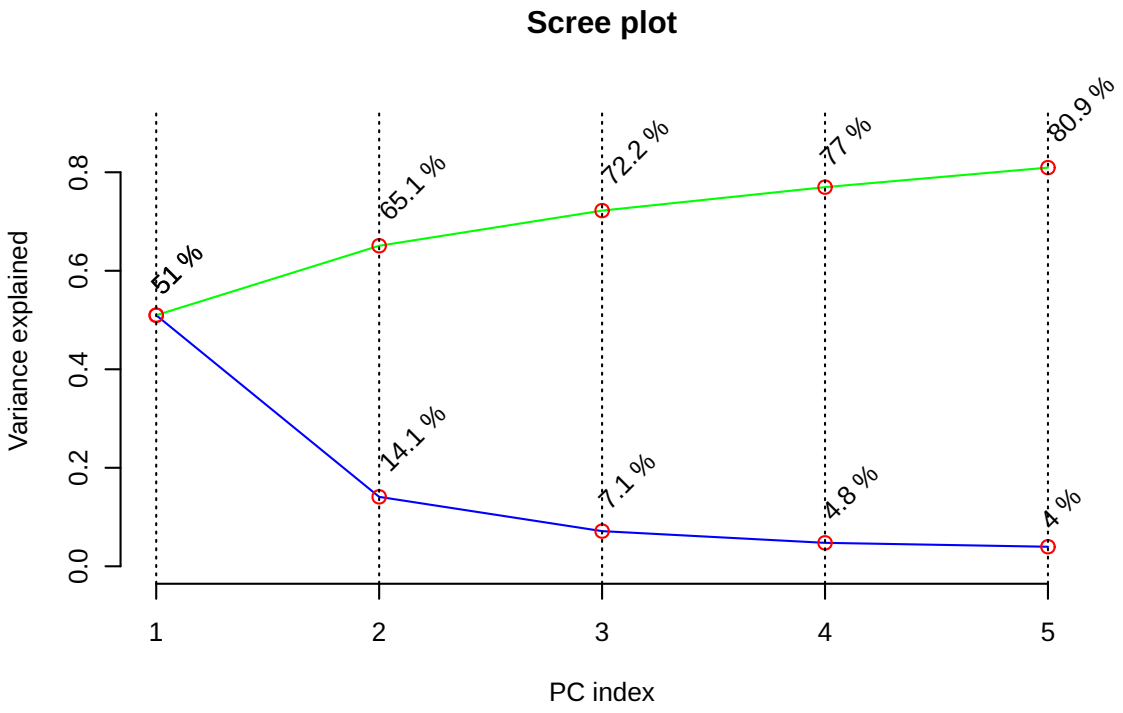

Figure 4: Scree plot shows the variance explained by PCs. The green line on top shows the accumulated variance explained; the blue line underneath shows the variance explained by individual PC.

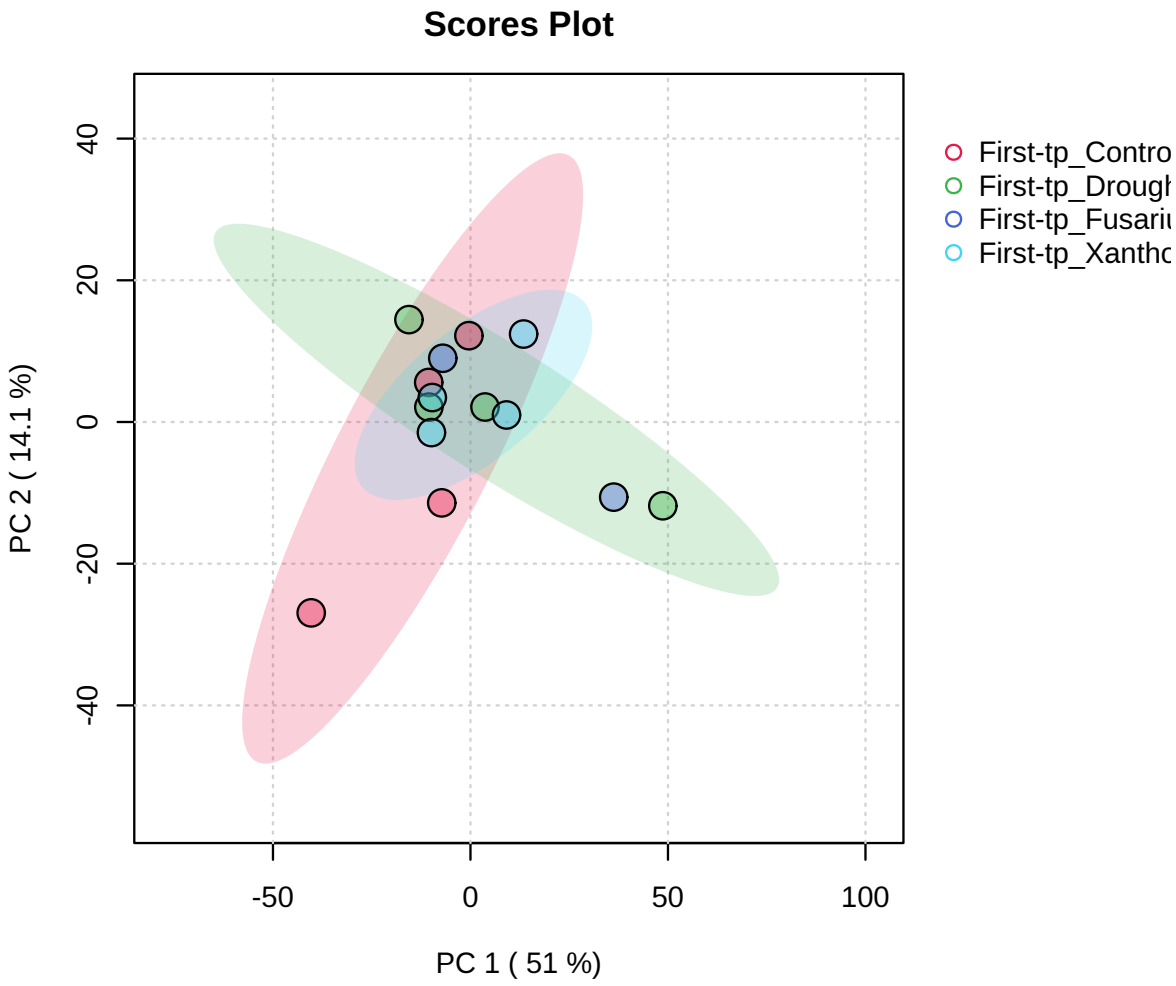

Figure 5: Scores plot between the selected PCs. The explained variances are shown in brackets.

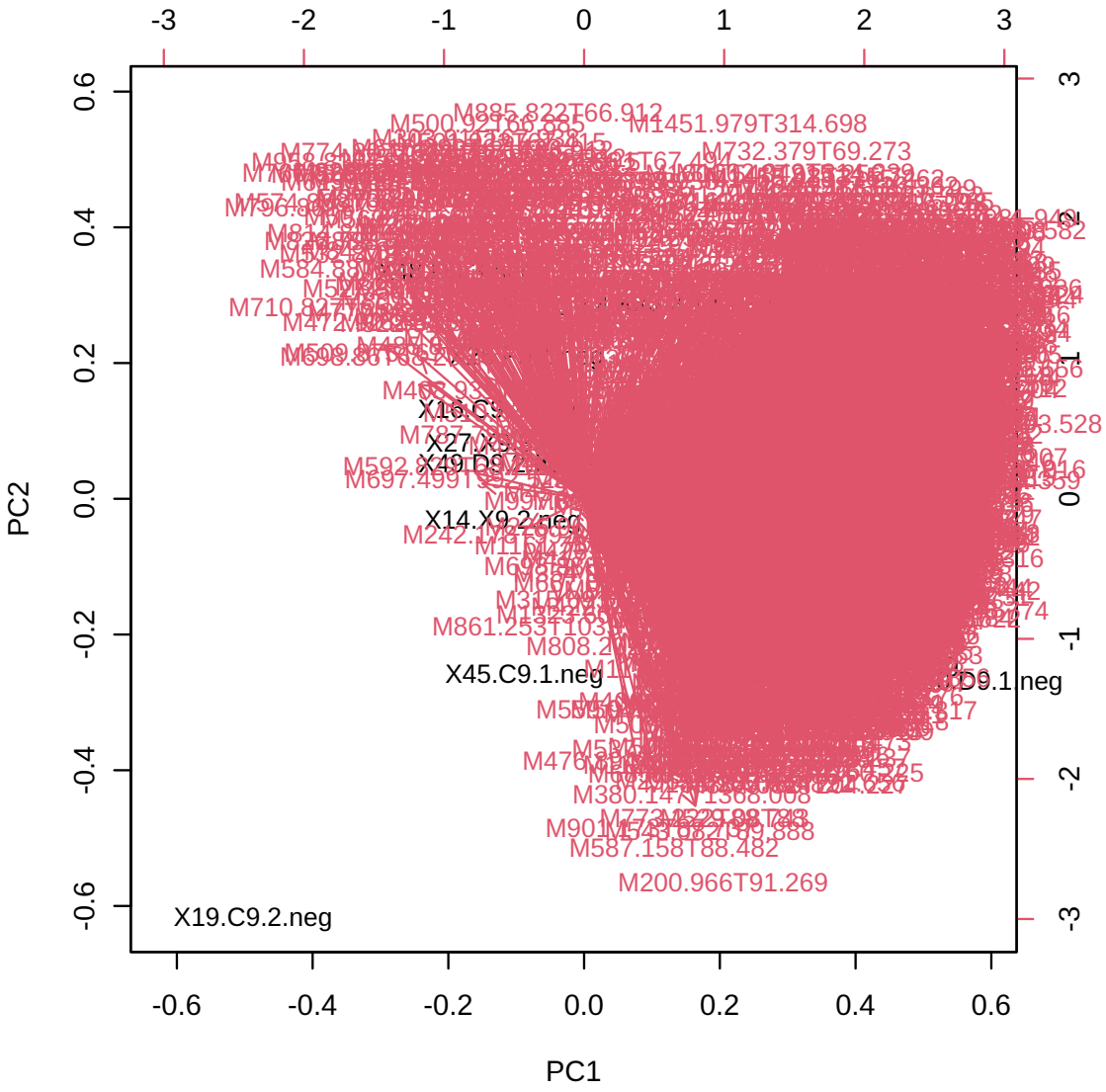

Figure 6: PCA biplot between the selected PCs. Note, you may want to test different centering and scaling normalization methods for the biplot to be displayed properly.

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 7 shows the overview of scores plots; Figure 8 shows the 2-D scores plot between selected components; Figure 9 shows the 3-D scores plot between selected components; Figure 10 shows the loading plot between the selected components;Figure 11 shows the classification performance with different number of components; Figure 12 shows the results of permutation test for model validation; Figure 13 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

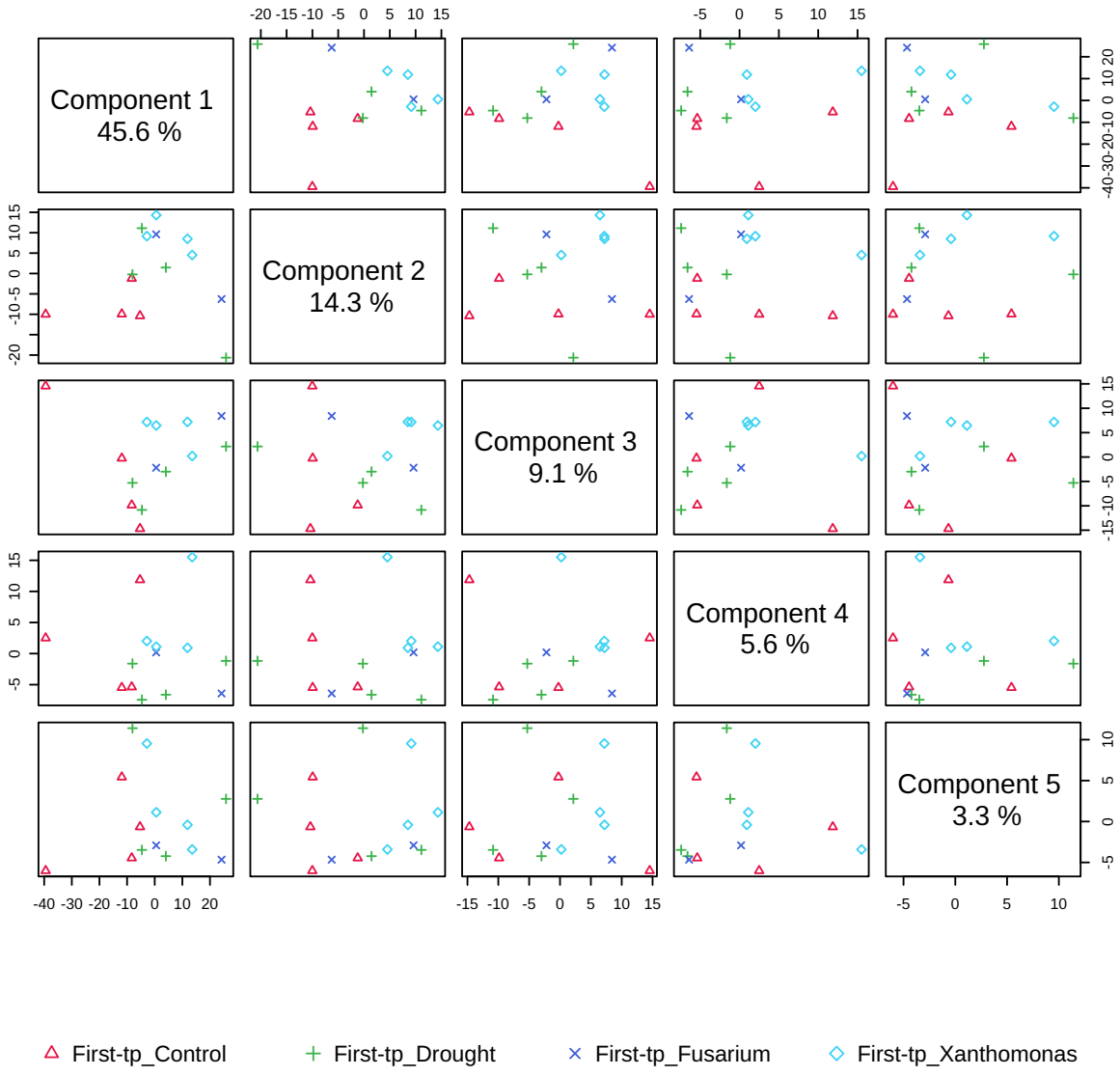

Figure 7: Pairwise scores plots between the selected components. The explained variance of each component is shown in the corresponding diagonal cell.

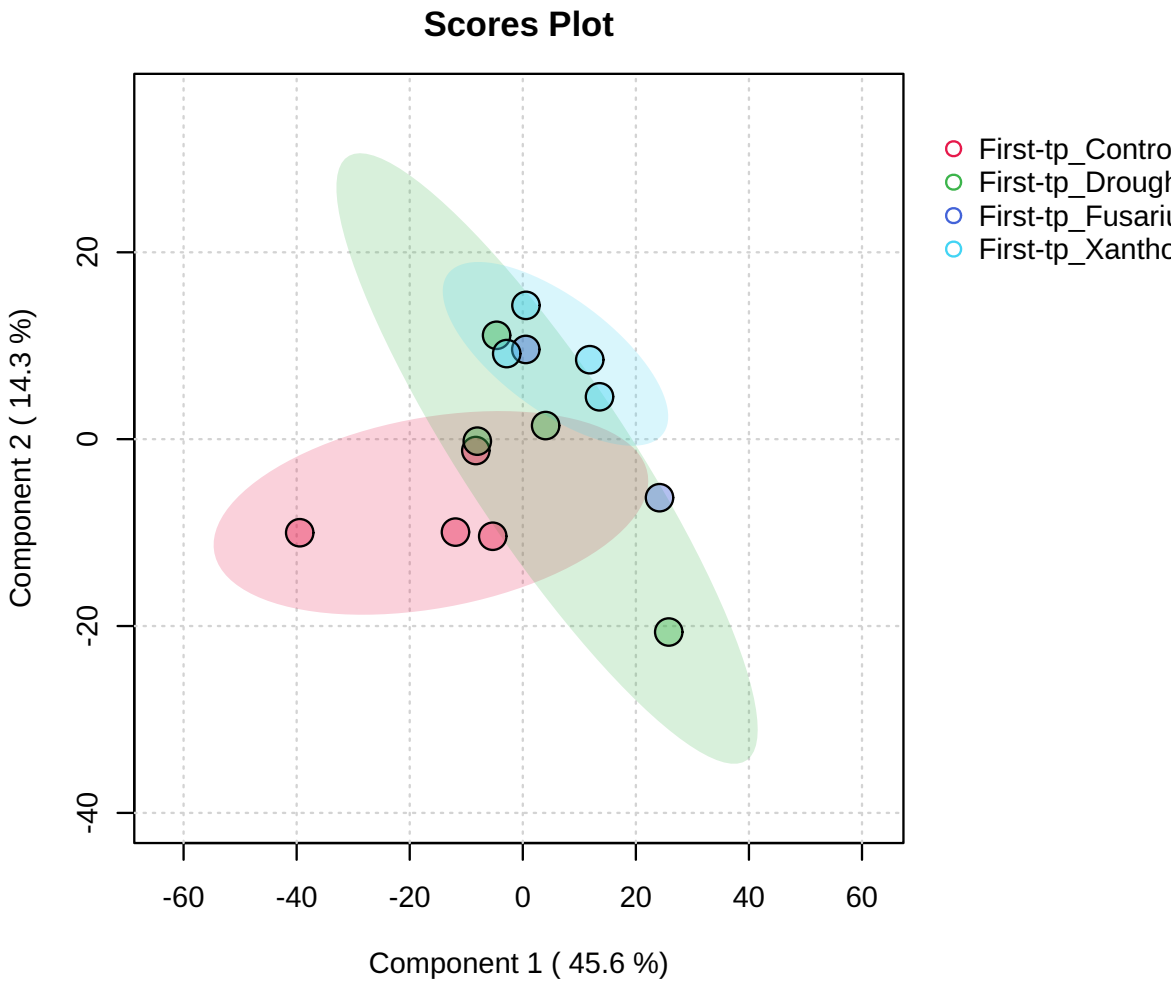

Figure 8: Scores plot between the selected PCs. The explained variances are shown in brackets.

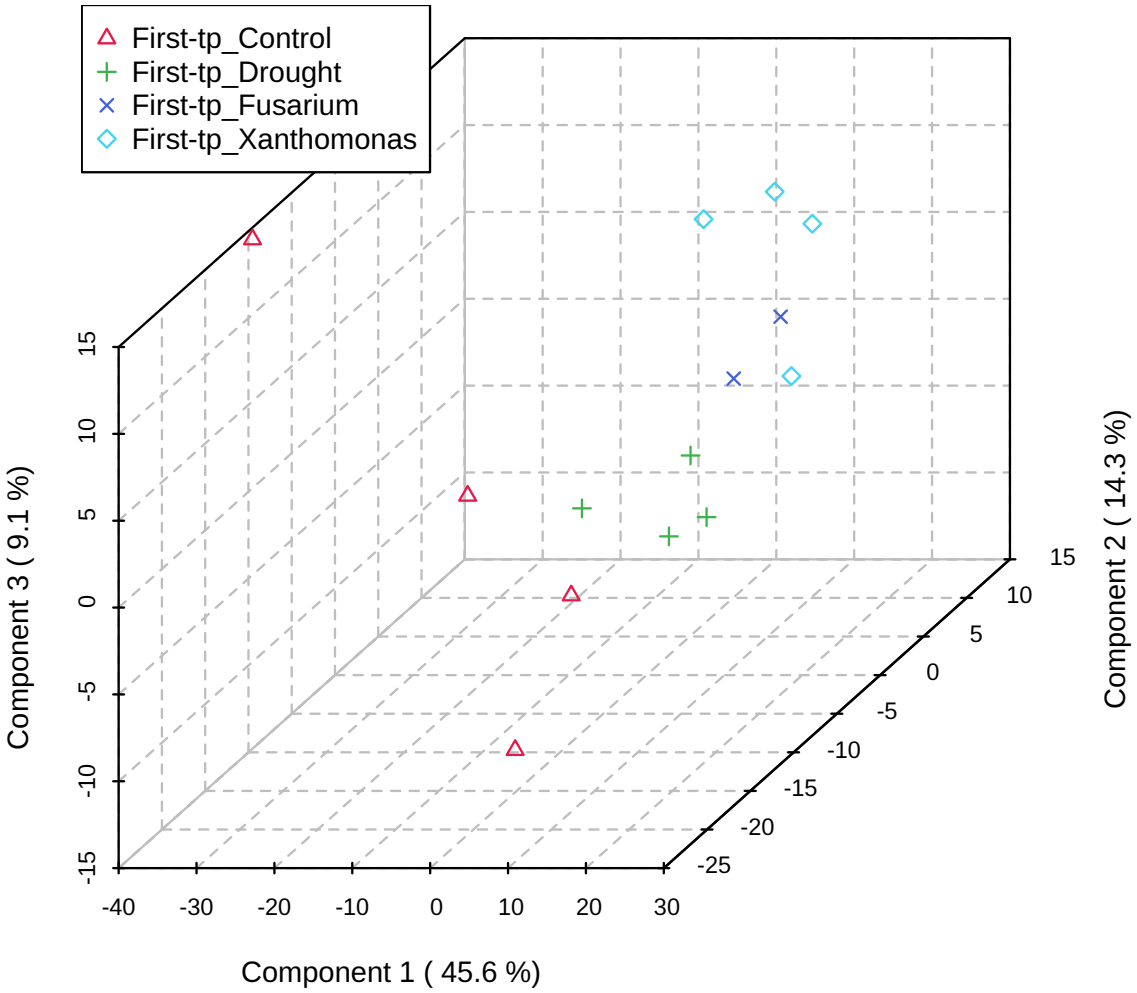

Figure 9: 3D scores plot between the selected PCs. The explained variances are shown in brackets.

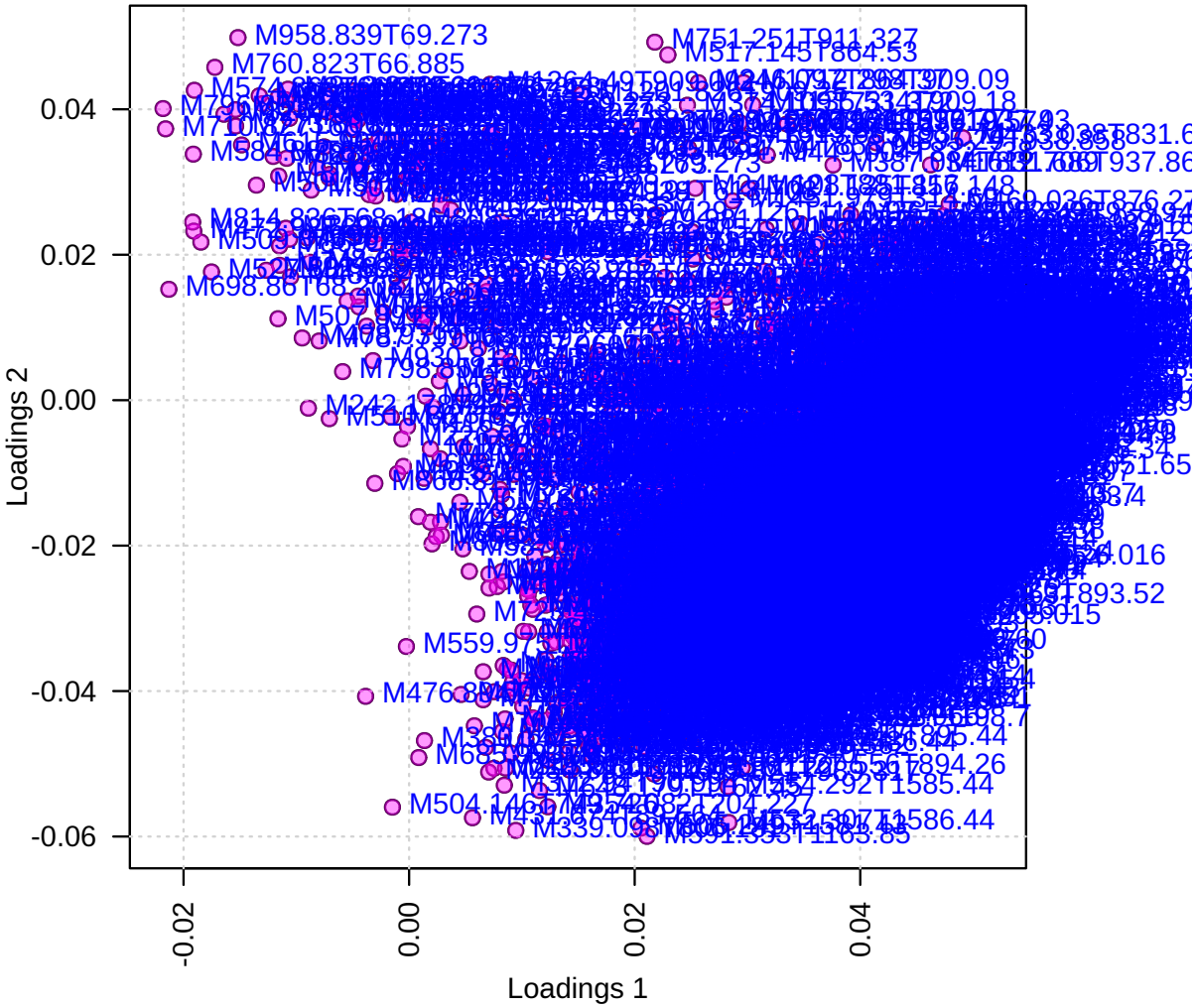

Figure 10: Loadings plot between the selected PCs.

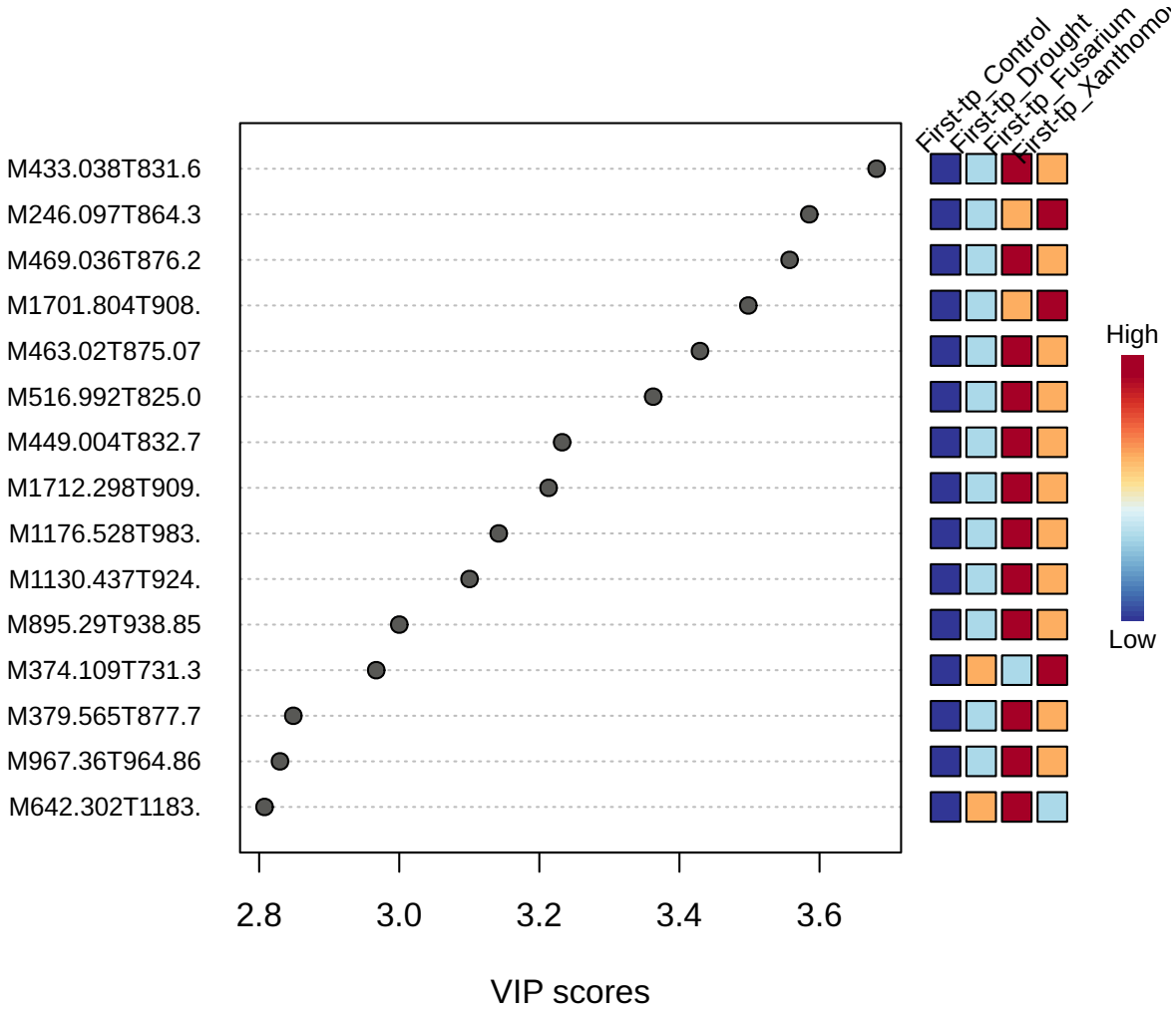

Figure 11: Important features identified by PLS-DA. The colored boxes on the right indicate the relative concentrations of the corresponding metabolite in each group under study.

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 14 shows the clustering result in the form of a dendrogram. Figure 15 shows the clustering result in the form of a heatmap.

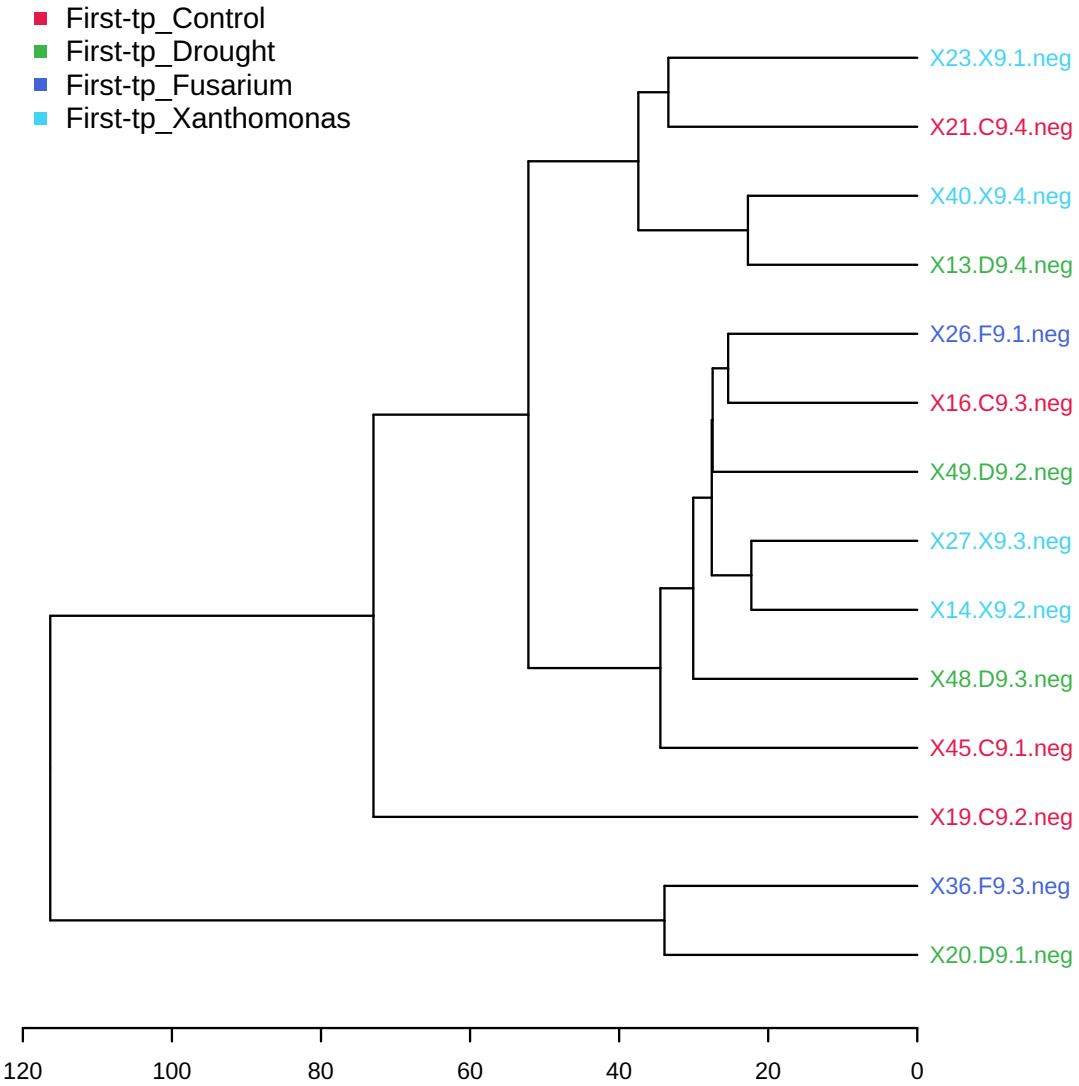

Figure 12: Clustering result shown as dendrogram (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

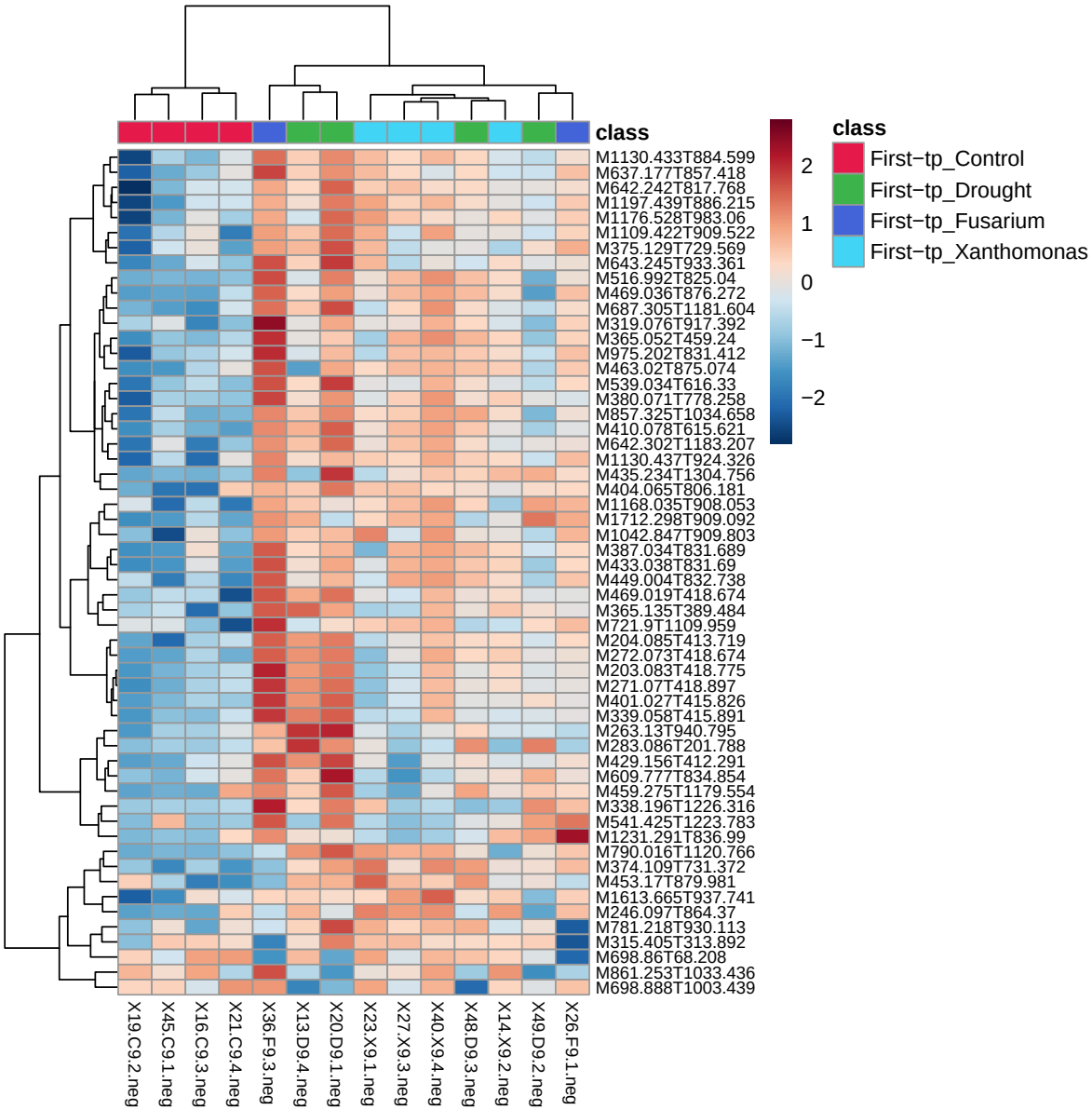

Figure 13: Clustering result shown as heatmap (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

3 Appendix: R Command History

```
 [1] "mSet<-InitDataObjects(\"pktable\", \"stat\", FALSE)"
 [2] "mSet<-Read.TextData(mSet, \"Replacing_with_your_file_path\", \"rowu\", \"disc\");"
 [3] "mSet<-SanityCheckData(mSet)"
 [4] "mSet<-ReplaceMin(mSet);"
 [5] "mSet<-SanityCheckData(mSet)"
 [6] "mSet<-FilterVariable(mSet, \"F\", 25, \"iqr\", 0, \"mean\", 0)"
 [7] "mSet<-PreparePrenormData(mSet)"
 [8] "mSet<-GetGroupNames(mSet, \"\")"
 [9] "feature.nm.vec <- c(\"\")"
[10] "smpl.nm.vec <- c(\"X34.F9.2.neg\",\"X46.F9.4.neg\")"
[11] "grp.nm.vec <- c(\"First-tp_Control\",\"First-tp_Drought\",\"First-tp_Fusarium\",\"First-tp_Xan
[12] "mSet<-UpdateData(mSet, T)"
[13] "mSet<-PreparePrenormData(mSet)"
[14] "mSet<-Normalization(mSet, \"CompNorm\", \"LogNorm\", \"ParetoNorm\", \"sodium_formate\", ratio
[15] "mSet<-PlotNormSummary(mSet, \"norm_0_\", \"png\", 72, width=NA)"
[16] "mSet<-PlotSampleNormSummary(mSet, \"snorm_0_\", \"png\", 72, width=NA)"
[17] "mSet<-ANOVA.Anal(mSet, F, 0.05, FALSE)"
[18] "mSet<-PlotANOVA(mSet, \"aov_0_\", \"png\", 72, width=NA)"
[19] "mSet<-ANOVA.Anal(mSet, F, 1.0, FALSE)"
[20] "mSet<-PlotANOVA(mSet, \"aov_1_\", \"png\", 72, width=NA)"
[21] "mSet<-Calculate.ANOVA.posthoc(mSet, \"fisher\", 0.05)"
[22] "mSet<-ANOVA.Anal(mSet, F, 0.99435, FALSE)"
[23] "mSet<-PlotANOVA(mSet, \"aov_2_\", \"png\", 72, width=NA)"
[24] "mSet<-PCA.Anal(mSet)"
[25] "mSet<-PlotPCAPairSummary(mSet, \"pca_pair_0_\", \"png\", 72, width=NA, 5)"
[26] "mSet<-PlotPCAScree(mSet, \"pca_scree_0_\", \"png\", 72, width=NA, 5)"
[27] "mSet<-PlotPCA2DScore(mSet, \"pca_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[28] "mSet<-PlotPCALoading(mSet, \"pca_loading_0_\", \"png\", 72, width=NA, 1,2);"
[29] "mSet<-PlotPCABiplot(mSet, \"pca_biplot_0_\", \"png\", 72, width=NA, 1,2)"
[30] "mSet<-PlotPCA3DLoading(mSet, \"pca_loading3d_0_\", \"json\", 1,2,3)"
[31] "mSet<-PLSR.Anal(mSet, reg=TRUE)"
[32] "mSet<-PlotPLSPairSummary(mSet, \"pls_pair_0_\", \"png\", 72, width=NA, 5)"
[33] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[34] "mSet<-PlotPLS3DScoreImg(mSet, \"pls_score3d_0_\", \"png\", 72, width=NA, 1,2,3, 40)"
[35] "mSet<-PlotPLSLoading(mSet, \"pls_loading_0_\", \"png\", 72, width=NA, 1, 2);"
[36] "mSet<-PlotPLS3DLoading(mSet, \"pls_loading3d_0_\", \"json\", 1,2,3)"
[37] "mSet<-PlotPLS.Imp(mSet, \"pls_imp_0_\", \"png\", 72, width=NA, \"vip\", \"Comp. 1\", 15,FALSE)
[38] "mSet<-PlotHCTree(mSet, \"tree_0_\", \"png\", 72, width=NA, \"euclidean\", \"ward.D\")"
[39] "mSet<-PlotHeatMap(mSet, \"heatmap_1_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\
[40] "mSet<-PlotSubHeatMap(mSet, \"heatmap_2_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclide
[41] "mSet<-SaveTransformedData(mSet)"
[42] "mSet<-PreparePDFReport(mSet, \"guest9353209576210757826\")\n"
```

The report was generated on Tue Mar 12 12:07:37 2024 with R version 4.3.2 (2023-10-31), OS system: Linux, version: -Ubuntu SMP Tue Jan 9 15:25:40 UTC 2024 .