Metabolomic Data Analysis with MetaboAnalyst 6.0

Name: guest2711281677997763362

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. The uploaded data file contains 18 (samples) by 1006 (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)
X12.D12.2.neg	1005	1	1006
X28.D12.3.neg	1005	1	1006
X44.D12.1.neg	372	634	1006
X52.D12.4.neg	1006	0	1006
X05.F12.4.neg	1004	2	1006
X22.F12.1.neg	998	8	1006
X38.F12.2.neg	1005	1	1006
X43.F12.3.neg	1003	3	1006
X02.C12.2.neg	996	10	1006
X18.C12.3.neg	984	22	1006
X33.C12.4.neg	994	12	1006
X53.Blank.neg	479	527	1006
X10.QC1.neg	1000	6	1006
X24.QC.2.neg	1006	0	1006
X39.QC3.neg	994	12	1006
X09.X12.3.neg	365	641	1006
X32.X12.2.neg	997	9	1006
X41.X12.1.neg	998	8	1006

[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.

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

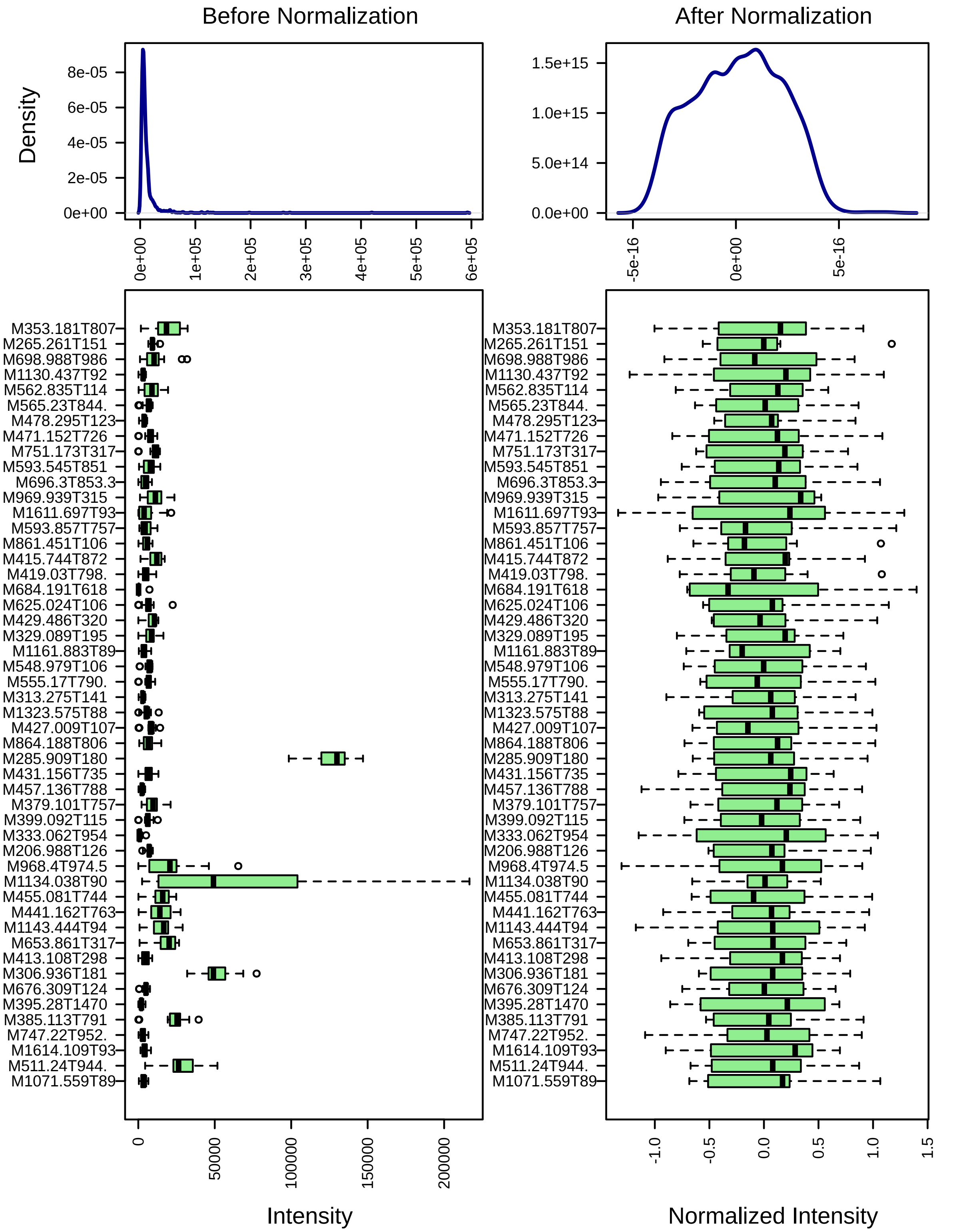

Figure 1: Box plots and kernel density plots before and after normalization. The boxplots show at most 50 features due to space limit. The density plots are based on all samples.

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 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.

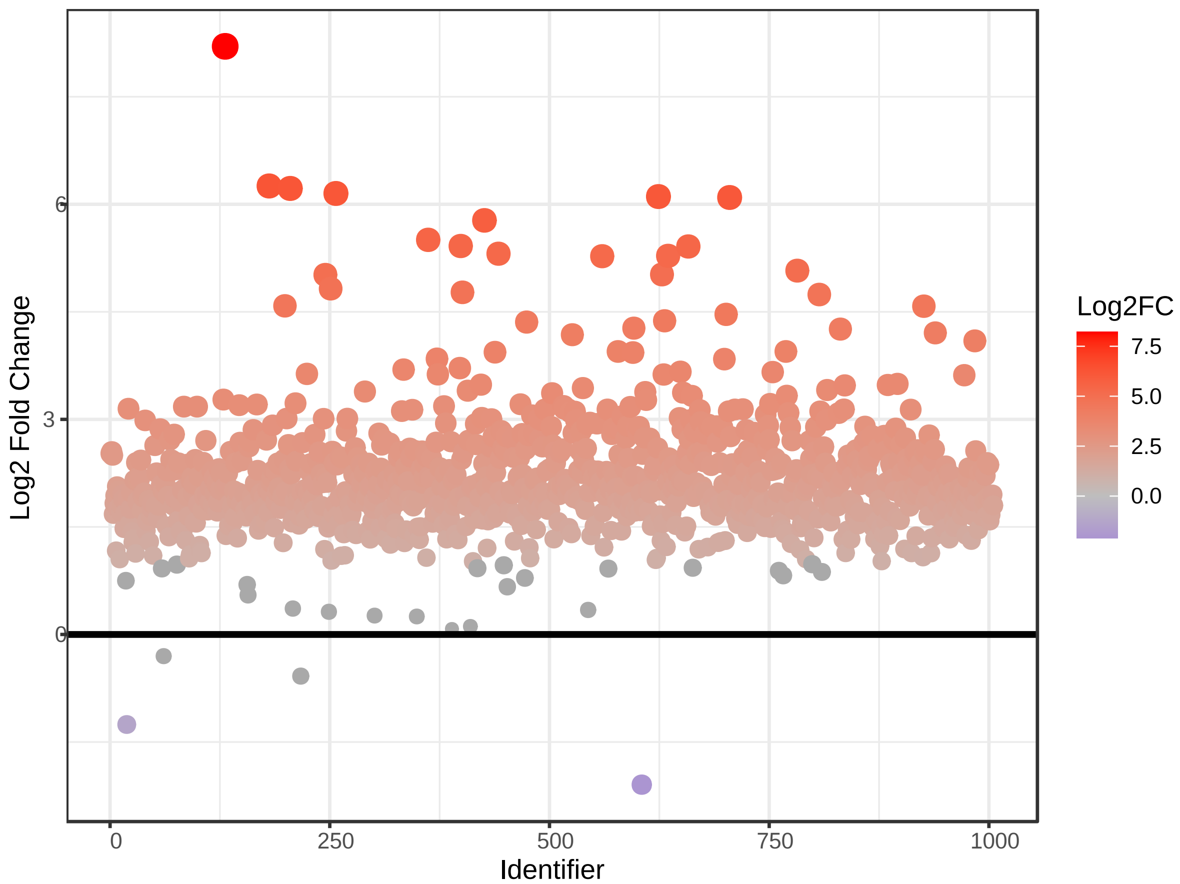

Figure 2: Important features selected by fold-change analysis with threshold 2. The red circles represent features above the threshold. Note the values are on log scale, so that both up-regulated and down-regulated features can be plotted in a symmetrical way

Table 2: Top 50 features identified by fold change analysis

	Peaks(mz/rt)	Fold Change	log2(FC)
1	M279.139T1150.344	294.62	8.2027
2	M293.118T1042.937	76.397	6.2554
3	M433.039T456.919	74.59	6.2209
4	M204.085T413.719	71.058	6.1509
5	M516.992T825.04	68.964	6.1078
6	M469.036T876.272	68.345	6.0948
7	M206.081T834.983	54.776	5.7755
8	M410.078T615.621	45.38	5.504
9	M387.034T831.689	42.759	5.4181
10	M365.528T823.98	42.562	5.4115
11	M365.052T459.24	39.688	5.3106
12	M531.009T876.272	38.888	5.2812
13	M449.004T832.738	38.744	5.2759
14	M463.02T875.074	33.695	5.0744
15	M379.565T877.71	32.476	5.0213
16	M1611.697T937.933	32.362	5.0162
17	M380.071T778.258	28.306	4.8231
18	M455.021T825.04	27.324	4.7721
19	M413.179T861.909	26.745	4.7412
20	M861.253T1033.436	23.977	4.5836
21	M319.076T917.392	23.894	4.5786
22	M471.046T618.546	22.078	4.4646
23	M433.038T831.69	20.749	4.375
24	M379.067T781.128	20.495	4.3572
25	M1613.665T937.741	19.324	4.2723
26	M365.143T832.741	19.148	4.2591
27	M401.05T875.074	18.459	4.2063
28	M684.191T618.708	18.139	4.181
29	M447.05T875.673	17.087	4.0949
30	M402.053T876.272	15.425	3.9472
31	M815.832T67.938	15.425	3.9472
32	M857.325T1034.658	15.314	3.9368
33	M964.359T921.966	15.262	3.9319
34	M641.302T1183.734	14.363	3.8443
35	M872.81T67.519	14.323	3.8403
36	M341.535T617.585	13.133	3.7151
37	M430.136T769.014	12.944	3.6942
38	M661.142T919.275	12.672	3.6636
39	M529.184T941.182	12.636	3.6595
40	M867.206T809.129	12.42	3.6346
41	M967.36T964.868	12.367	3.6284
42	M477.061T615.621	12.344	3.6257
43	M775.338T1039.594	12.261	3.616
44	M759.145T875.232	11.273	3.4948
45	M687.305T1181.604	11.189	3.4841
46	M333.062T954.304	11.155	3.4797
47	M886.794T67.143	11.111	3.4739
48	M517.897T66.885	10.827	3.4365
49	M341.465T625.973	10.627	3.4097
50	M642.302T1183.207	10.556	3.4

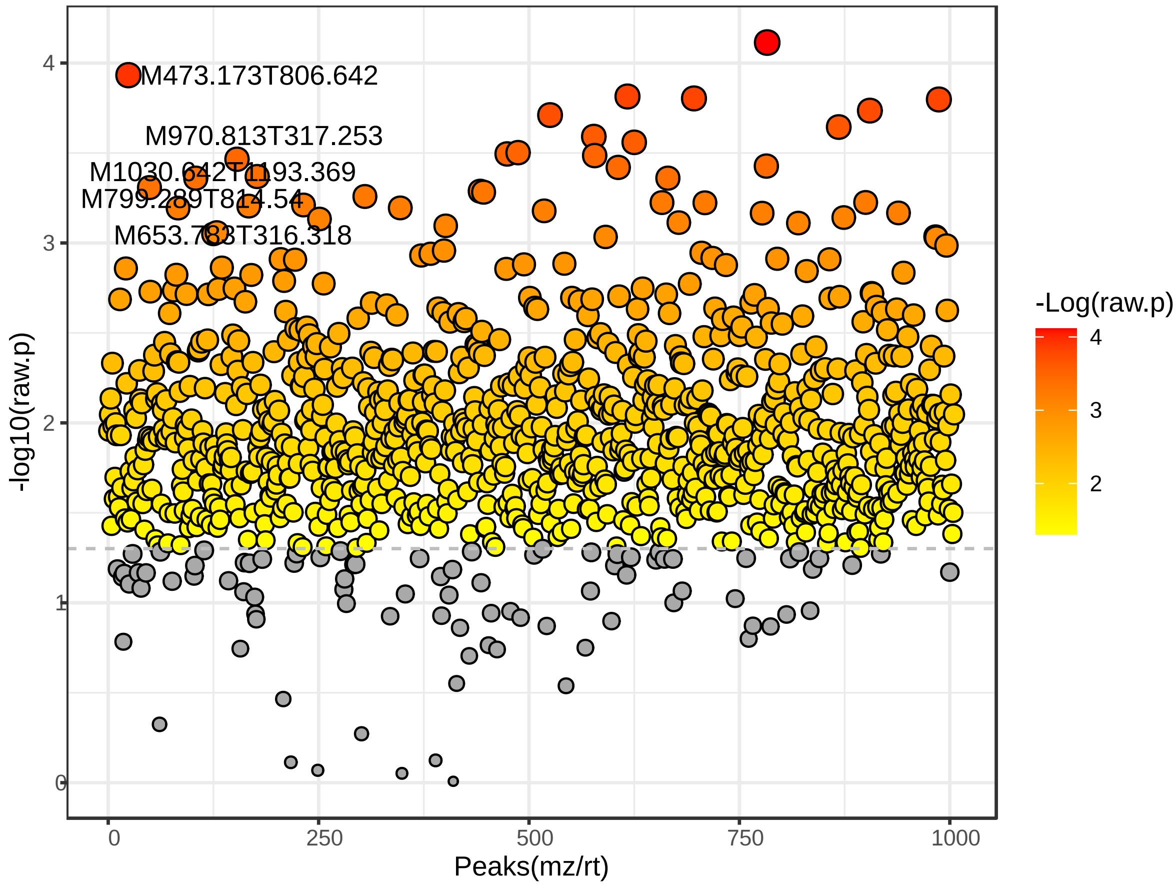

Figure 3: Important features selected by t-tests with threshold 0.05. The red circles represent features above the threshold. Note the p values are transformed by -log10 so that the more significant features (with smaller p values) will be plotted higher on the graph.

Table 3: Top 50 features identified by t-tests

	Peaks(mz/rt)	t.stat	p.value	-log10(p)	FDR
1	M289.12T769.482	11.799	7.694e-05	4.1138	0.0208
2	M473.173T806.642	10.824	0.0001168	3.9326	0.0208
3	M1014.782T1351.411	10.228	0.00015344	3.8141	0.0208
4	M674.842T67.64	10.175	0.00015728	3.8033	0.0208
5	M615.195T761.983	10.147	0.00015938	3.7976	0.0208
6	M238.069T801.926	9.8495	0.00018383	3.7356	0.0208
7	M513.129T730.532	9.7326	0.00019465	3.7107	0.0208
8	M1014.811T1344.847	9.426	0.00022682	3.6443	0.0208
9	M847.188T757.14	9.1878	0.00025624	3.5914	0.0208
10	M447.151T677.76	9.0464	0.00027586	3.5593	0.0208
11	M631.833T319.717	8.7983	0.00031478	3.502	0.0208
12	M379.067T781.128	8.7677	0.00032002	3.4948	0.0208
13	M402.053T876.272	8.7288	0.00032682	3.4857	0.0208
14	M970.813T317.253	8.6425	0.00034257	3.4652	0.0208
15	M463.02T875.074	8.4839	0.00037391	3.4272	0.0208
16	M645.252T899.393	8.4535	0.00038029	3.4199	0.0208
17	M970.001T316.098	8.2468	0.00042735	3.3692	0.0208
18	M1030.642T1193.369	8.2108	0.00043624	3.3603	0.0208
19	M365.136T768.138	8.2104	0.00043633	3.3602	0.0208
20	M799.289T814.54	8.002	0.00049233	3.3077	0.0208
21	M365.052T459.24	7.9238	0.0005155	3.2878	0.0208
22	M505.029T765.672	7.9021	0.00052218	3.2822	0.0208
23	M398.263T1317.762	7.81	0.00055161	3.2584	0.0208
24	M824.341T916.725	7.683	0.00059552	3.2251	0.0208
25	M365.528T823.98	7.6802	0.00059652	3.2244	0.0208
26	M621.037T758.225	7.6758	0.00059812	3.2232	0.0208
27	M999.268T316.316	7.6294	0.00061524	3.211	0.0208
28	M481.183T769.795	7.6066	0.00062389	3.2049	0.0208
29	M533.169T900.984	7.5686	0.00063862	3.1948	0.0208
30	M653.783T316.318	7.5656	0.0006398	3.194	0.0208
31	M568.907T66.858	7.5076	0.00066313	3.1784	0.0208
32	M401.05T875.074	7.4651	0.00068084	3.167	0.0208
33	M799.852T67.938	7.4601	0.00068298	3.1656	0.0208
34	M395.28T1470.273	7.3706	0.00072233	3.1413	0.021061
35	M380.071T778.258	7.341	0.00073593	3.1332	0.021061
36	M776.814T66.22	7.2682	0.00077068	3.1131	0.021061
37	M472.886T66.482	7.2587	0.00077538	3.1105	0.021061
38	M455.021T825.04	7.2037	0.00080313	3.0952	0.021241
39	M821.606T900.199	7.0728	0.00087409	3.0584	0.021855
40	M642.242T817.768	7.0428	0.00089138	3.0499	0.021855
41	M658.83T67.591	6.9921	0.00092153	3.0355	0.021855
42	M431.156T735.48	6.9849	0.00092593	3.0334	0.021855
43	M447.05T875.673	6.97	0.00093507	3.0292	0.021855
44	M1614.109T935.709	6.8199	0.0010333	2.9858	0.023496
45	M387.034T831.689	6.724	0.0011025	2.9576	0.023496
46	M469.036T876.272	6.6825	0.0011342	2.9453	0.023496
47	M240.933T64.89	6.6644	0.0011483	2.94	0.023496
48	M641.302T1183.734	6.6324	0.0011738	2.9304	0.023496
49	M508.215T930.69	6.5869	0.0012112	2.9168	0.023496
50	M341.088T439.204	6.5715	0.0012242	2.9122	0.023496

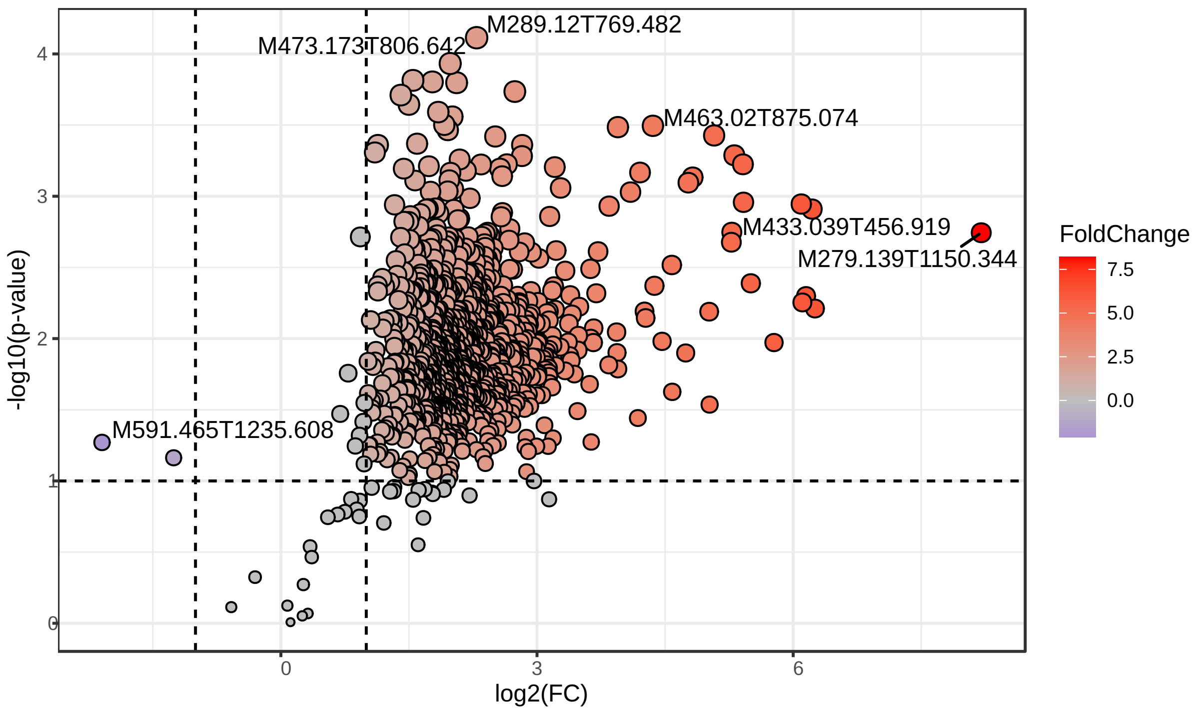

Figure 4: Important features selected by volcano plot with fold change threshold (x) 2 and t-tests threshold (y) 0.1. The red circles represent features above the threshold. Note both fold changes and p values are log transformed. The further its position away from the (0,0), the more significant the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

Table 4: Top 50 features identified by volcano plot

	Peaks(mz/rt)	FC	log2(FC)	raw.pval	-log10(p)
1	M289.12T769.482	4.8995	2.2926	7.694e-05	4.1138
2	M473.173T806.642	3.9531	1.983	0.0001168	3.9326
3	M1014.782T1351.411	2.9222	1.5471	0.00015344	3.8141
4	M674.842T67.64	3.4207	1.7743	0.00015728	3.8033
5	M615.195T761.983	4.1651	2.0583	0.00015938	3.7976
6	M238.069T801.926	6.6807	2.74	0.00018383	3.7356
7	M513.129T730.532	2.649	1.4054	0.00019465	3.7107
8	M1014.811T1344.847	2.8274	1.4995	0.00022682	3.6443
9	M847.188T757.14	3.589	1.8436	0.00025624	3.5914
10	M447.151T677.76	4.0222	2.008	0.00027586	3.5593
11	M631.833T319.717	3.7684	1.9139	0.00031478	3.502
12	M379.067T781.128	20.495	4.3572	0.00032002	3.4948
13	M402.053T876.272	15.425	3.9472	0.00032682	3.4857
14	M970.813T317.253	3.8771	1.955	0.00034257	3.4652
15	M463.02T875.074	33.695	5.0744	0.00037391	3.4272
16	M645.252T899.393	5.7006	2.5111	0.00038029	3.4199
17	M970.001T316.098	3.0216	1.5953	0.00042735	3.3692
18	M1030.642T1193.369	2.1986	1.1366	0.00043624	3.3603
19	M365.136T768.138	7.0893	2.8256	0.00043633	3.3602
20	M799.289T814.54	2.1448	1.1008	0.00049233	3.3077
21	M365.052T459.24	39.688	5.3106	0.0005155	3.2878
22	M505.029T765.672	7.0782	2.8234	0.00052218	3.2822
23	M398.263T1317.762	4.2681	2.0936	0.00055161	3.2584
24	M824.341T916.725	6.264	2.6471	0.00059552	3.2251
25	M365.528T823.98	42.562	5.4115	0.00059652	3.2244
26	M621.037T758.225	5.0775	2.3441	0.00059812	3.2232
27	M999.268T316.316	3.325	1.7333	0.00061524	3.211
28	M481.183T769.795	9.2388	3.2077	0.00062389	3.2049
29	M533.169T900.984	5.9218	2.566	0.00063862	3.1948
30	M653.783T316.318	2.7111	1.4389	0.0006398	3.194
31	M568.907T66.858	4.4923	2.1674	0.00066313	3.1784
32	M401.05T875.074	18.459	4.2063	0.00068084	3.167
33	M799.852T67.938	3.9618	1.9862	0.00068298	3.1656
34	M395.28T1470.273	6.0291	2.5919	0.00072233	3.1413
35	M380.071T778.258	28.306	4.8231	0.00073593	3.1332
36	M776.814T66.22	3.925	1.9727	0.00077068	3.1131
37	M472.886T66.482	2.973	1.5719	0.00077538	3.1105
38	M455.021T825.04	27.324	4.7721	0.00080313	3.0952
39	M821.606T900.199	9.6883	3.2762	0.00087409	3.0584
40	M642.242T817.768	4.0552	2.0198	0.00089138	3.0499
41	M658.83T67.591	3.878	1.9553	0.00092153	3.0355
42	M431.156T735.48	3.3699	1.7527	0.00092593	3.0334
43	M447.05T875.673	17.087	4.0949	0.00093507	3.0292
44	M1614.109T935.709	4.6411	2.2145	0.0010333	2.9858
45	M387.034T831.689	42.759	5.4181	0.0011025	2.9576
46	M469.036T876.272	68.345	6.0948	0.0011342	2.9453
47	M240.933T64.89	2.5165	1.3314	0.0011483	2.94
48	M641.302T1183.734	14.363	3.8443	0.0011738	2.9304
49	M508.215T930.69	3.5038	1.8089	0.0012112	2.9168
50	M341.088T439.204	3.3051	1.7247	0.0012242	2.9122

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

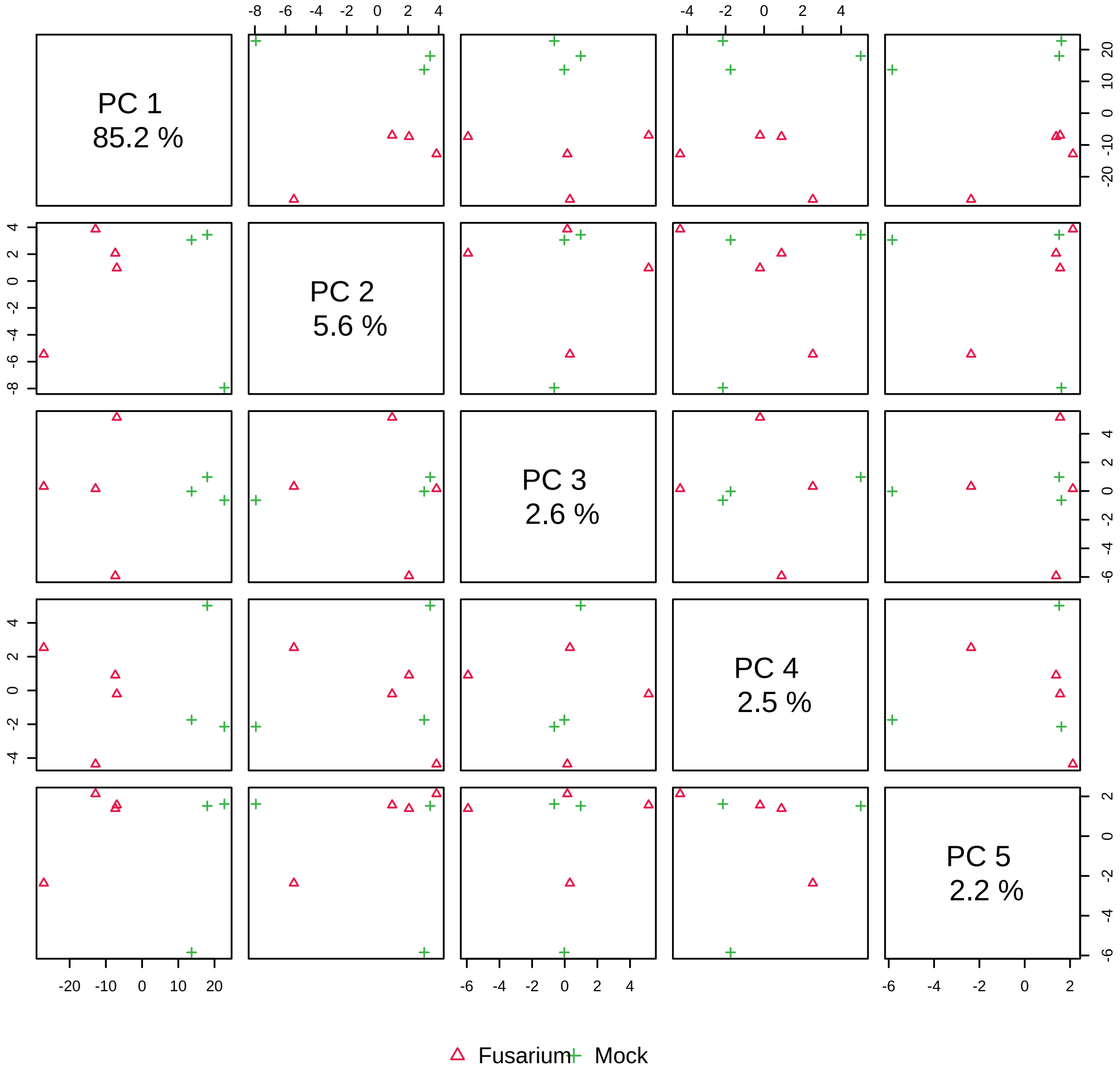

Figure 5: Pairwise score plots between the selected PCs. The explained variance of each PC is shown in the corresponding diagonal cell.

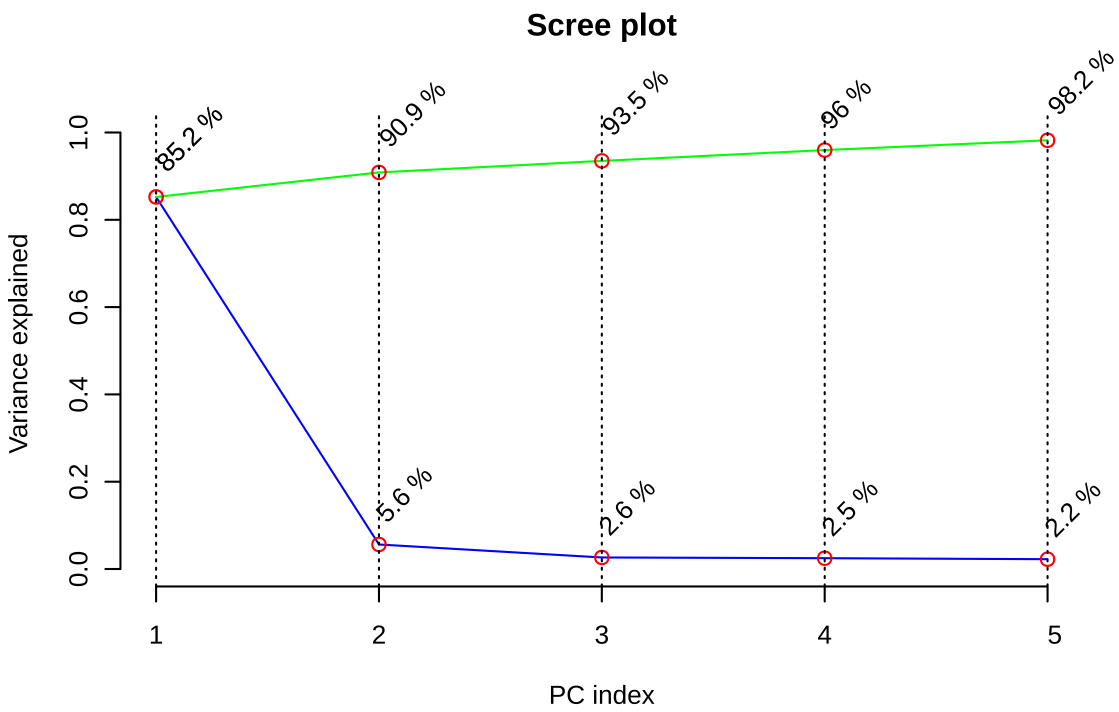

Figure 6: Scree plot shows the variance explained by PCs. The green line on top shows the accumulated variance explained; the blue line underneath shows the variance explained by individual PC.

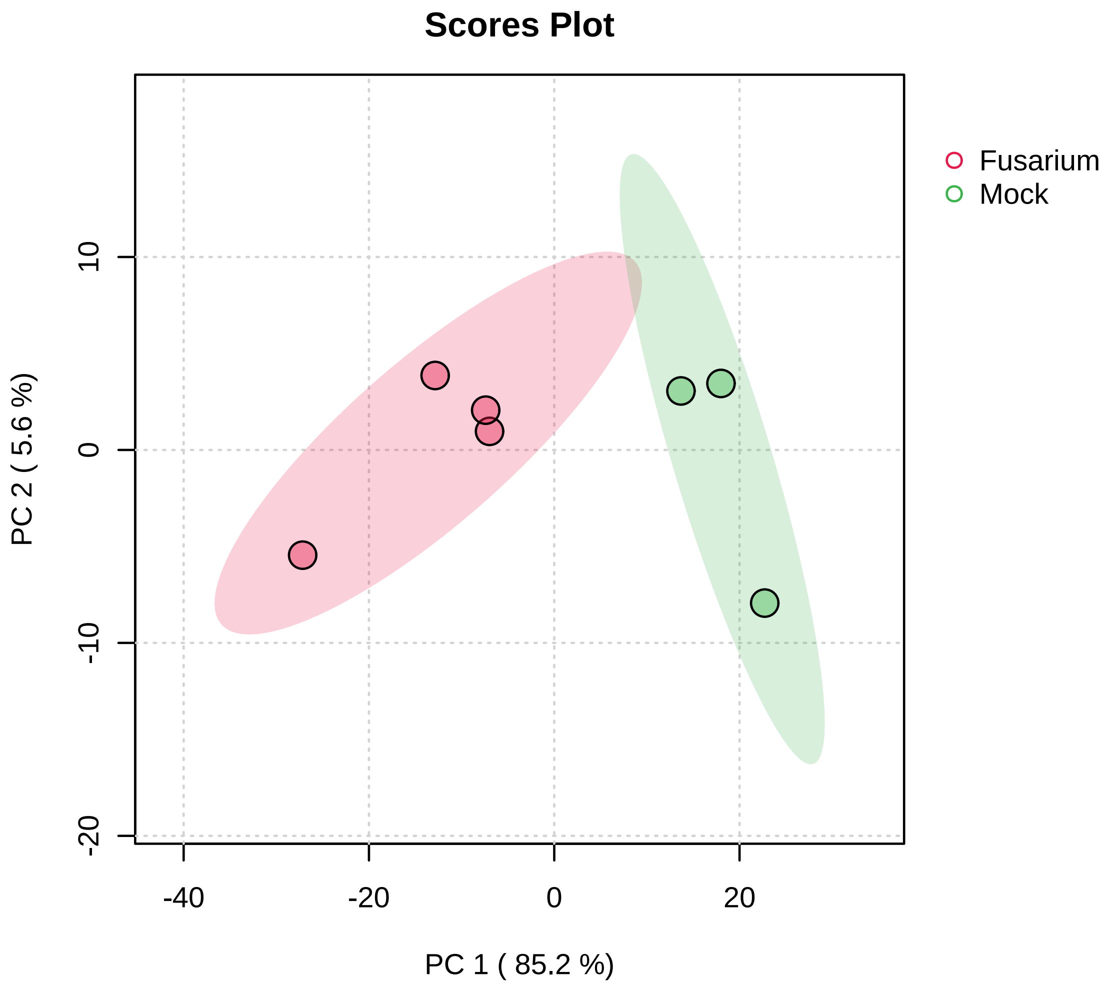

Figure 7: Scores plot between the selected PCs. The explained variances are shown in brackets.

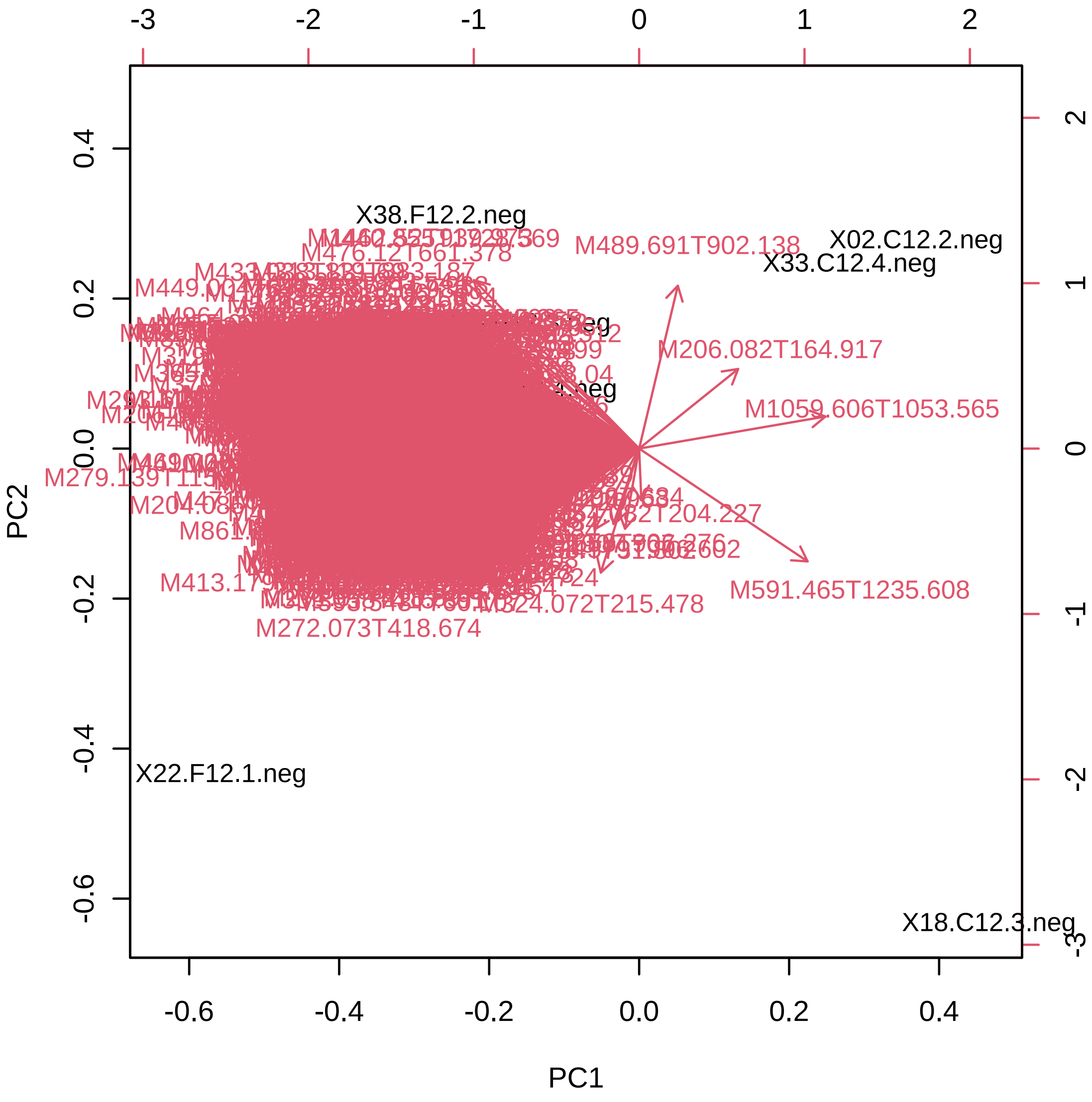

Figure 8: PCA biplot between the selected PCs. Note, you may want to test different centering and scaling normalization methods for the biplot to be displayed properly.

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

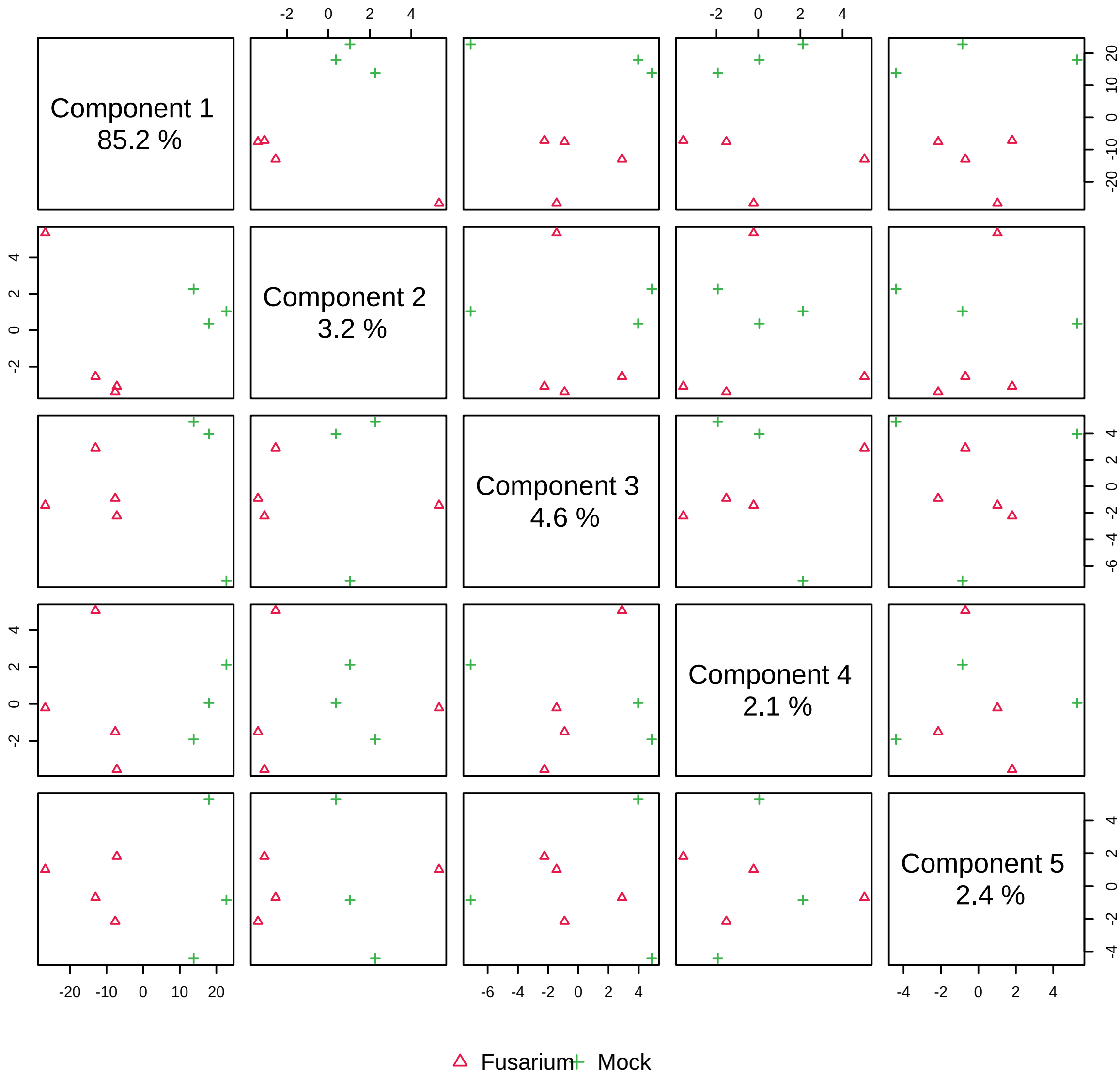

Figure 9: Pairwise scores plots between the selected components. The explained variance of each component is shown in the corresponding diagonal cell.

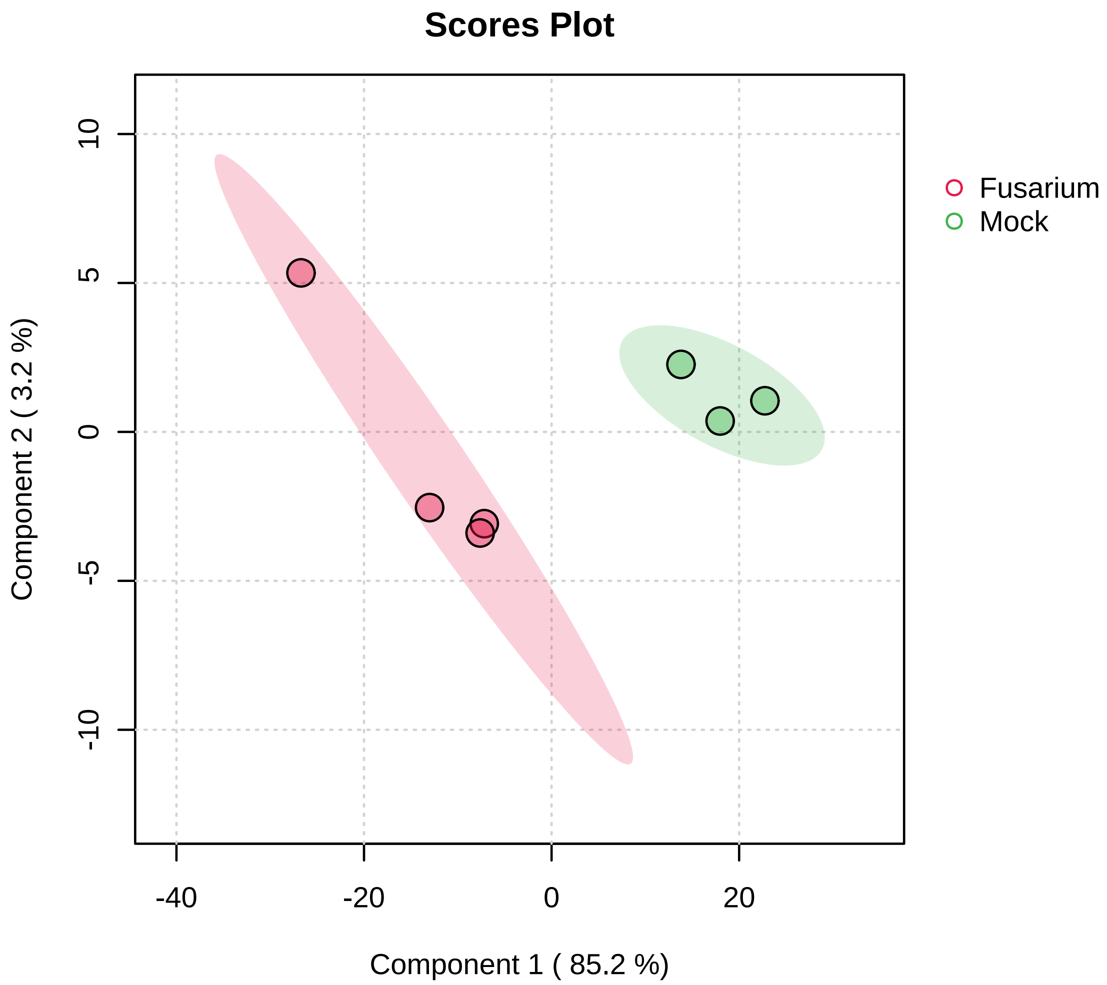

Figure 10: Scores plot between the selected PCs. The explained variances are shown in brackets.

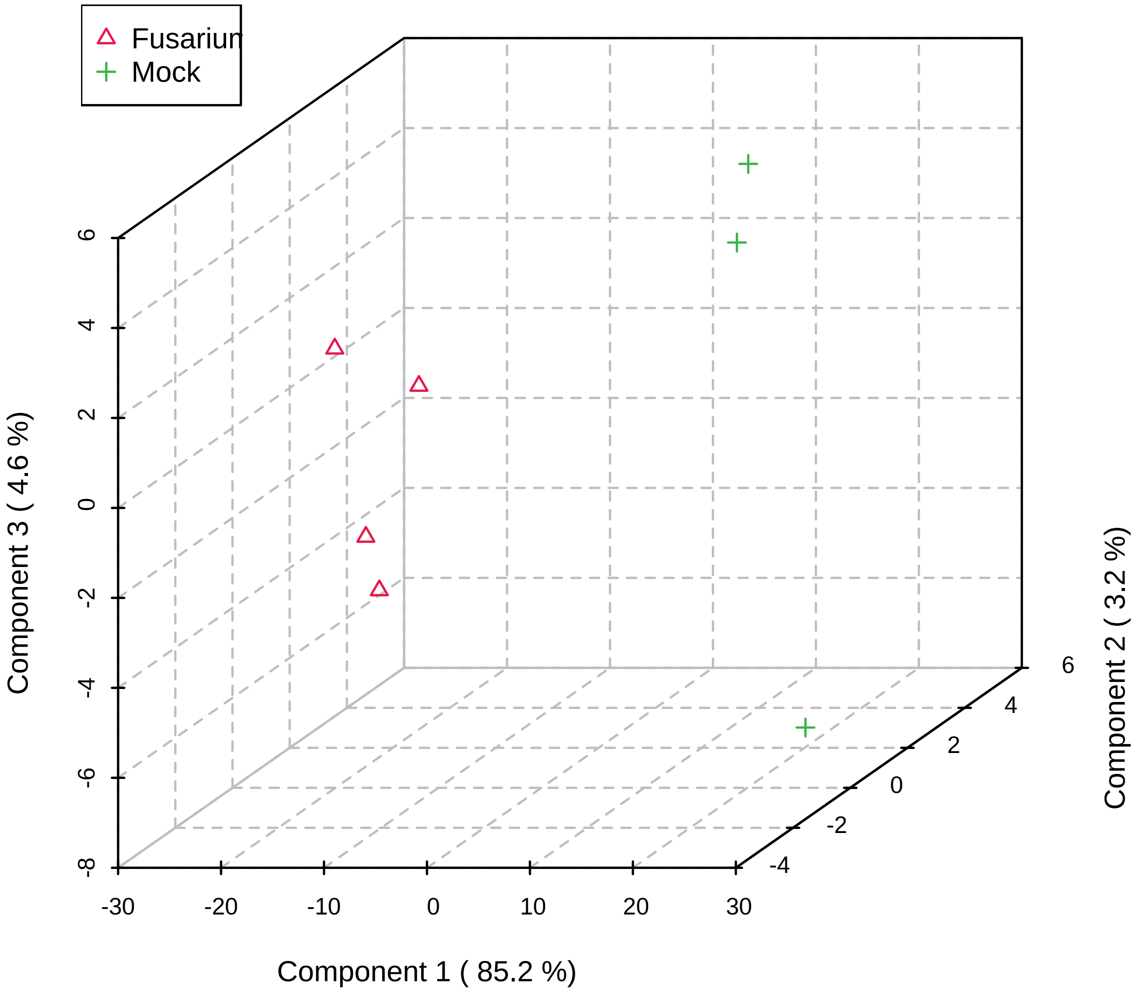

Figure 11: 3D scores plot between the selected PCs. The explained variances are shown in brackets.

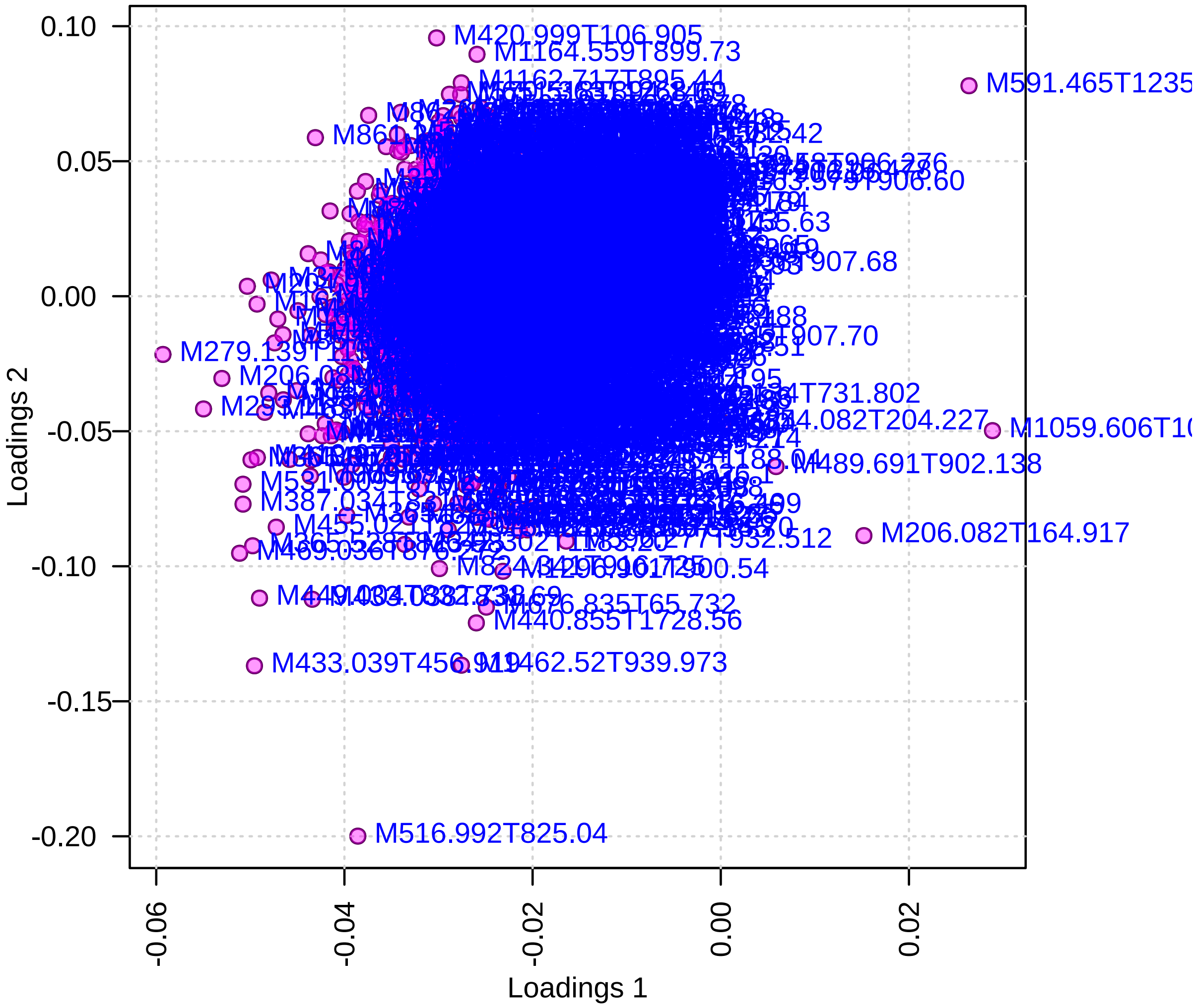

Figure 12: Loadings plot between the selected PCs.

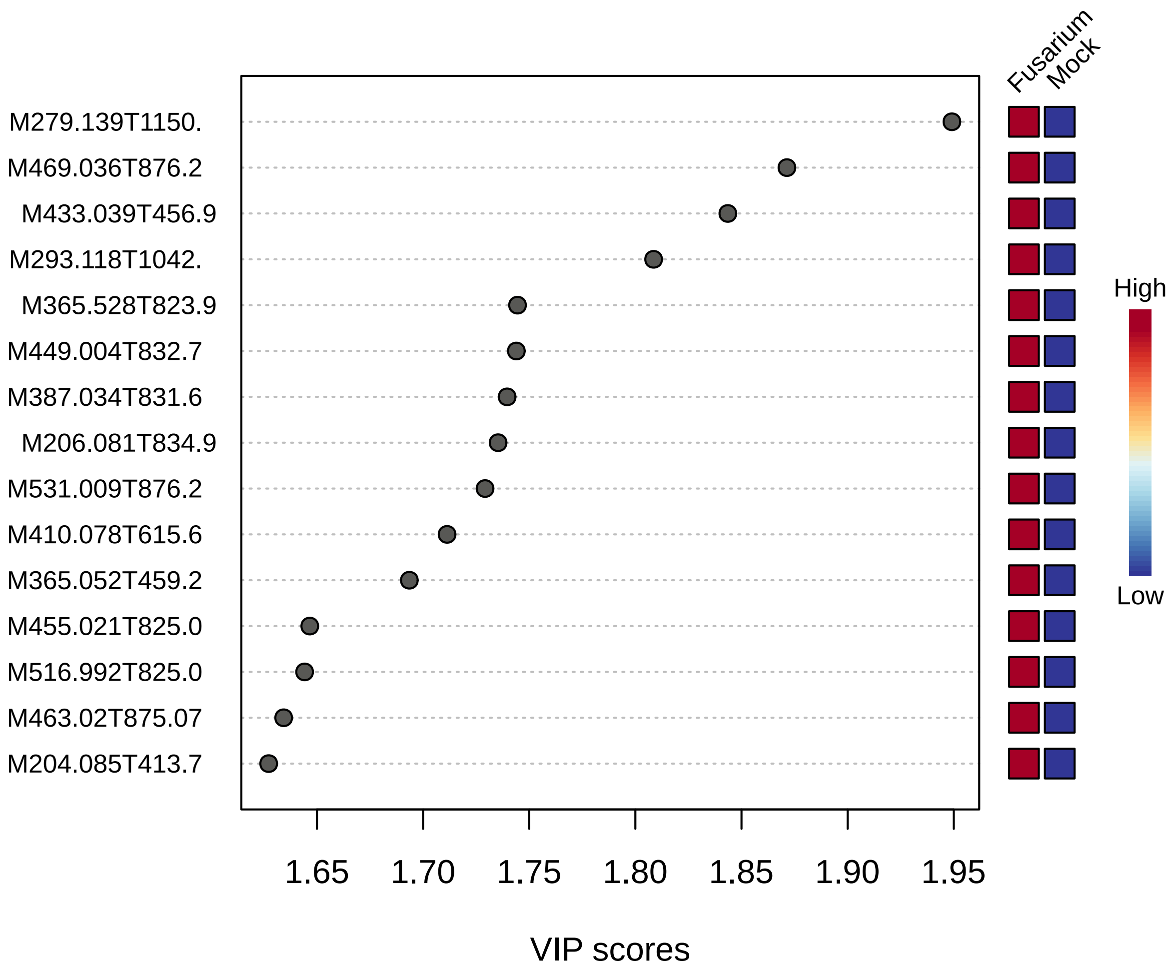

Figure 13: Important features identified by PLS-DA. The colored boxes on the right indicate the relative concentrations of the corresponding metabolite in each group under study.

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.

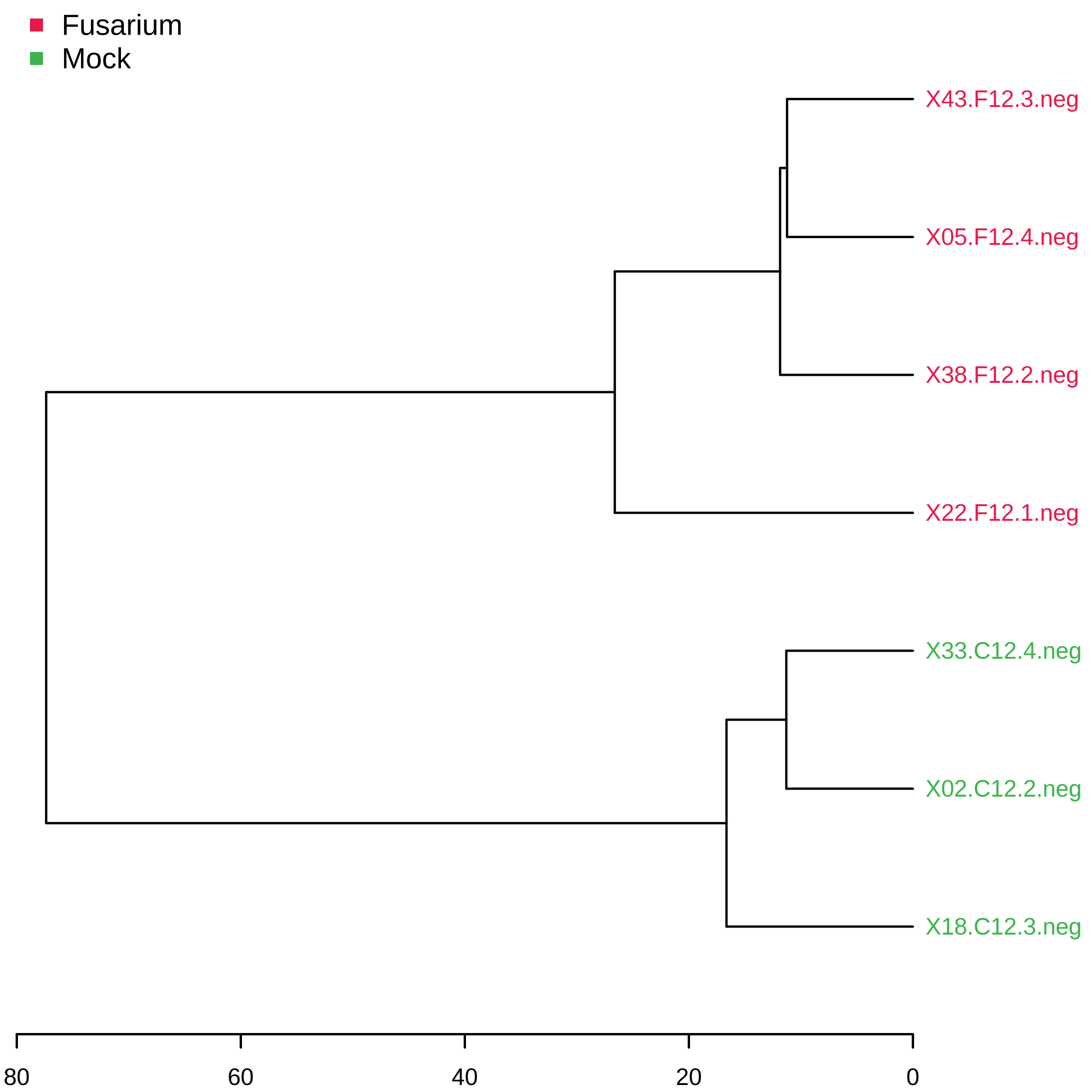

Figure 14: Clustering result shown as dendrogram (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

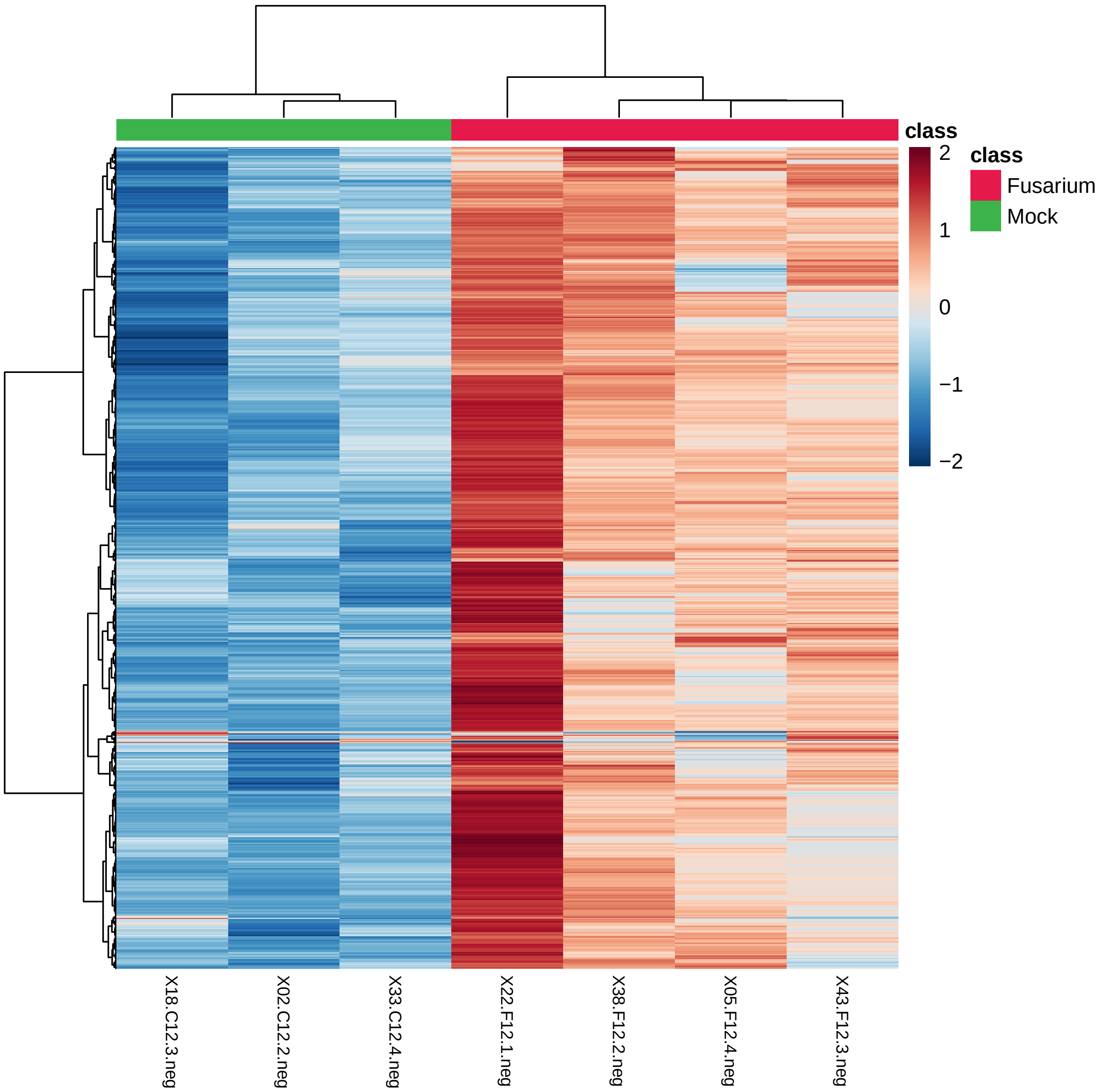

Figure 15: Clustering result shown as heatmap (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

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(\"X44.D12.1.neg\",\"X09.X12.3.neg\")"
[11] "grp.nm.vec <- c(\"Fusarium\",\"Mock\")"
[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<-FC.Anal(mSet, 2.0, 0, FALSE)"
[18] "mSet<-PlotFC(mSet, \"fc_0_\", \"png\", 72, width=NA)"
[19] "mSet<-FC.Anal(mSet, 2.0, 1, FALSE)"
[20] "mSet<-PlotFC(mSet, \"fc_1_\", \"png\", 72, width=NA)"
[21] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, TRUE, \"fdr\", FALSE)"
[22] "mSet<-PlotTT(mSet, \"tt_0_\", \"png\", 72, width=NA)"
[23] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, FALSE, \"raw\", FALSE)"
[24] "mSet<-PlotTT(mSet, \"tt_1_\", \"png\", 72, width=NA)"
[25] "mSet<-Volcano.Anal(mSet, FALSE, 2.0, 0, F, 0.1, TRUE, \"raw\")"
[26] "mSet<-PlotVolcano(mSet, \"volcano_0_\",1, 0, \"png\", 72, width=NA, -1)"
[27] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, FALSE, \"fdr\", FALSE)"
[28] "mSet<-PlotTT(mSet, \"tt_2_\", \"png\", 72, width=NA)"
[29] "mSet<-Ttests.Anal(mSet, T, 0.05, FALSE, FALSE, \"fdr\", FALSE)"
[30] "mSet<-PlotTT(mSet, \"tt_3_\", \"png\", 72, width=NA)"
[31] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, FALSE, \"fdr\", FALSE)"
[32] "mSet<-PlotTT(mSet, \"tt_4_\", \"png\", 72, width=NA)"
[33] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, FALSE, \"raw\", FALSE)"
[34] "mSet<-PlotTT(mSet, \"tt_5_\", \"png\", 72, width=NA)"
[35] "mSet<-PCA.Anal(mSet)"
[36] "mSet<-PlotPCAPairSummary(mSet, \"pca_pair_0_\", \"png\", 72, width=NA, 5)"
[37] "mSet<-PlotPCAScree(mSet, \"pca_scree_0_\", \"png\", 72, width=NA, 5)"
[38] "mSet<-PlotPCA2DScore(mSet, \"pca_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[39] "mSet<-PlotPCALoading(mSet, \"pca_loading_0_\", \"png\", 72, width=NA, 1,2);"
[40] "mSet<-PlotPCABiplot(mSet, \"pca_biplot_0_\", \"png\", 72, width=NA, 1,2)"
[41] "mSet<-PlotPCA3DLoading(mSet, \"pca_loading3d_0_\", \"json\", 1,2,3)"
[42] "mSet<-PLSR.Anal(mSet, reg=TRUE)"
[43] "mSet<-PlotPLSPairSummary(mSet, \"pls_pair_0_\", \"png\", 72, width=NA, 5)"
[44] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[45] "mSet<-PlotPLS3DScoreImg(mSet, \"pls_score3d_0_\", \"png\", 72, width=NA, 1,2,3, 40)"
[46] "mSet<-PlotPLSLoading(mSet, \"pls_loading_0_\", \"png\", 72, width=NA, 1, 2);"
[47] "mSet<-PlotPLS3DLoading(mSet, \"pls_loading3d_0_\", \"json\", 1,2,3)"
[48] "mSet<-PlotPLS.Imp(mSet, \"pls_imp_0_\", \"png\", 72, width=NA, \"vip\", \"Comp. 1\", 15,FALSE)
[49] "mSet<-PlotTT(mSet, \"tt_5_\", \"pdf\", 72, width=NA)"
[50] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, FALSE, \"raw\", FALSE)"
[51] "mSet<-PlotTT(mSet, \"tt_6_\", \"png\", 72, width=NA)"
[52] "mSet<-PlotTT(mSet, \"tt_6_\", \"pdf\", 72, width=NA)"
[53] "mSet<-PlotHCTree(mSet, \"tree_0_\", \"png\", 72, width=NA, \"euclidean\", \"ward.D\")"
[54] "mSet<-PlotHeatMap(mSet, \"heatmap_1_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\
[55] "mSet<-SaveTransformedData(mSet)"
[56] "mSet<-PreparePDFReport(mSet, \"guest2711281677997763362\")\n"
```

The report was generated on Tue Mar 12 14:28:33 2024 with R version 4.3.2 (2023-10-31), OS system: Linux, version: -Ubuntu SMP Tue Jan 9 15:25:40 UTC 2024 .