Metabolomic Data Analysis with MetaboAnalyst 6.0

Name: guest9259121545104474613

March 6, 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 172 (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)
C12.1	168	4	172
C12.3	172	0	172
C12.4	172	0	172
D12.1	172	0	172
D12.2	172	0	172
D12.3	172	0	172
D12.4	172	0	172
F12.1	172	0	172
F12.2	172	0	172
F12.3	163	9	172
F12.4	170	2	172
BLANK.2	167	5	172
QC.1	172	0	172
QC.2	172	0	172
QC.3	172	0	172
X12.1	172	0	172
X12.2	172	0	172
X12.3	172	0	172

[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

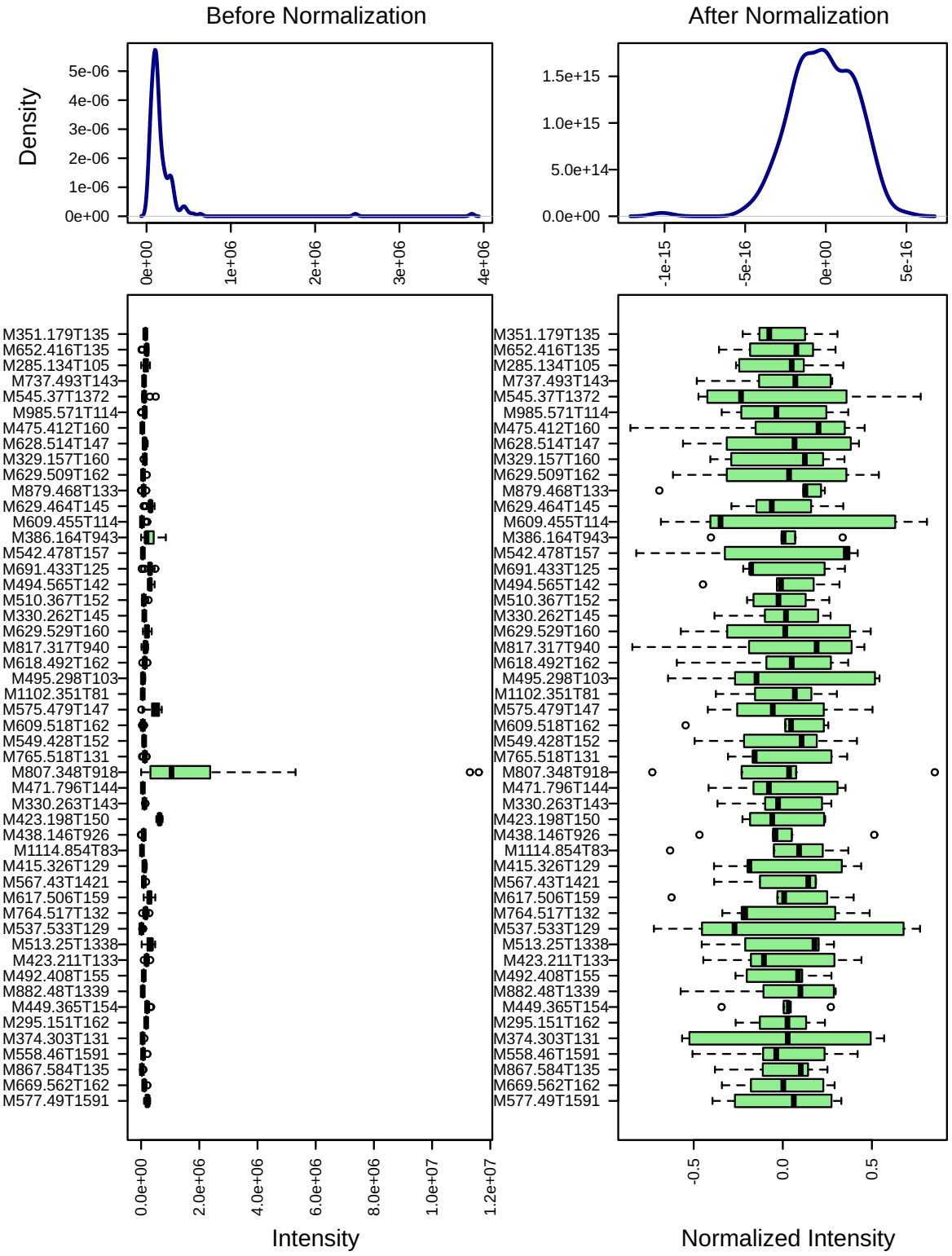

Figure 1: Box plots and kernel density plots before and after normalization. The boxplots show at most 50 features due to space limit. The density plots are based on all samples.

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.

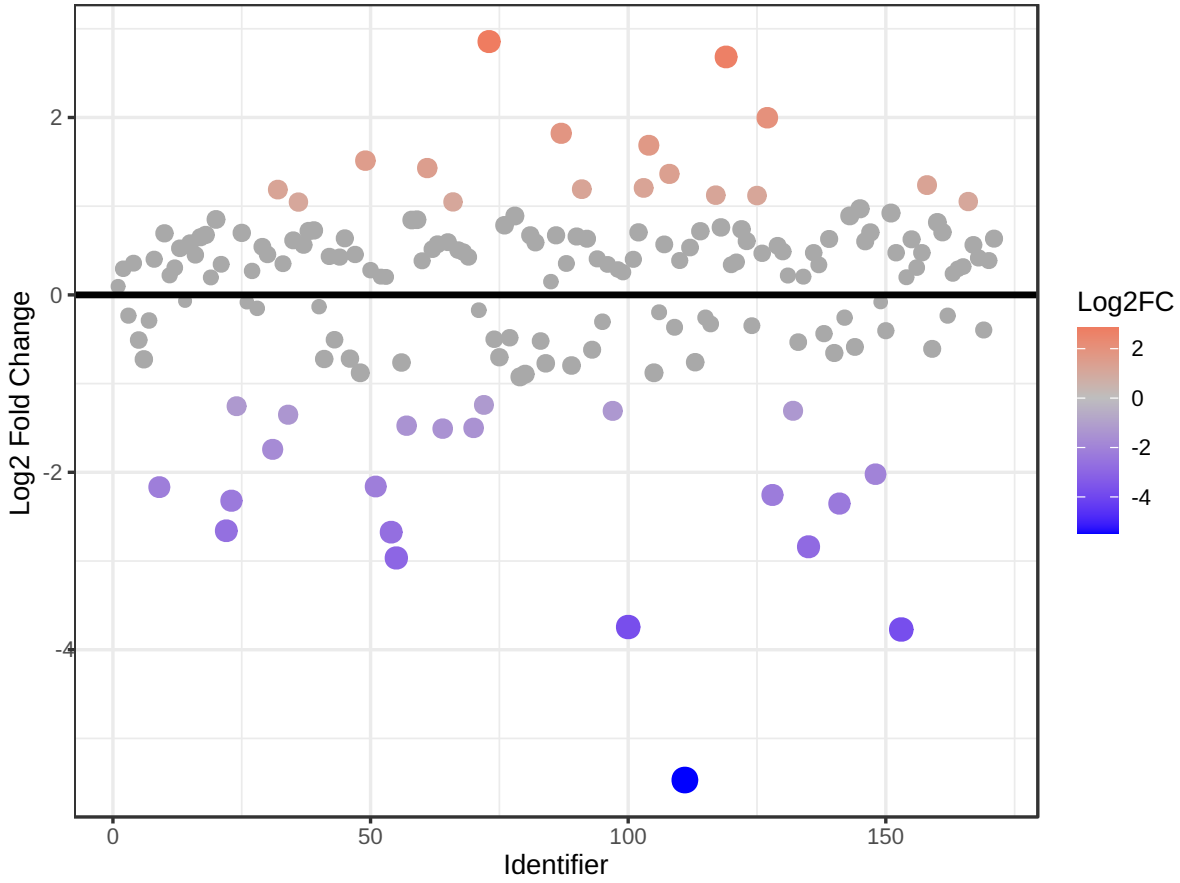

Figure 2: Important features selected by fold-change analysis with threshold 2. The red circles represent features above the threshold. Note the values are on log scale, so that both up-regulated and down-regulated features can be plotted in a symmetrical way

Table 2: Important features identified by fold change analysis

	Peaks(mz/rt)	Fold Change	log2(FC)
1	M722.259T892.9	0.022572	-5.4693
2	M443.333T1262.042	0.073198	-3.7721
3	M399.308T1262.042	0.074614	-3.7444
4	M561.396T1195.638	0.12802	-2.9656
5	M479.188T913.185	7.2402	2.856
6	M722.581T1518.016	0.1397	-2.8396
7	M951.603T1424.253	6.4202	2.6826
8	M537.533T1295.888	0.15658	-2.675
9	M609.455T1145.906	0.15835	-2.6588
10	M341.283T1305.932	0.19581	-2.3525
11	M771.504T1195.638	0.20021	-2.3204
12	M619.438T1245.233	0.20941	-2.2556
13	M556.445T1212.039	0.22266	-2.1671
14	M847.517T1163.004	0.22375	-2.1601
15	M722.58T1595.75	0.24635	-2.0212
16	M576.533T1641.124	3.9926	1.9973
17	M787.545T1575.098	3.5353	1.8218
18	M545.37T1372.761	0.29909	-1.7413
19	M542.478T1573.977	3.2196	1.6869
20	M541.474T1573.977	2.8551	1.5136
21	M808.52T1220.972	0.3517	-1.5076
22	M374.303T1319.161	0.35373	-1.4993
23	M495.298T1036.333	0.3599	-1.4744
24	M817.317T940.819	2.6942	1.4299
25	M475.412T1607.605	2.5741	1.3641
26	M441.224T1179.269	0.39209	-1.3507
27	M392.284T1436.378	0.40403	-1.3075
28	M772.55T1170.538	0.40431	-1.3065
29	M828.538T1289.688	0.41908	-1.2547
30	M504.265T1099.137	0.42327	-1.2403
31	M629.509T1624.418	2.3588	1.238
32	M807.348T918.929	2.3073	1.2062
33	M677.438T1339.916	2.2849	1.1921
34	M1085.421T973.898	2.2771	1.1872
35	M576.484T1557.453	2.1801	1.1244
36	M629.529T1601.79	2.1732	1.1199
37	M628.514T1473.756	2.0742	1.0525
38	M813.332T1056.081	2.0679	1.0482
39	M659.418T1406.262	2.0654	1.0465

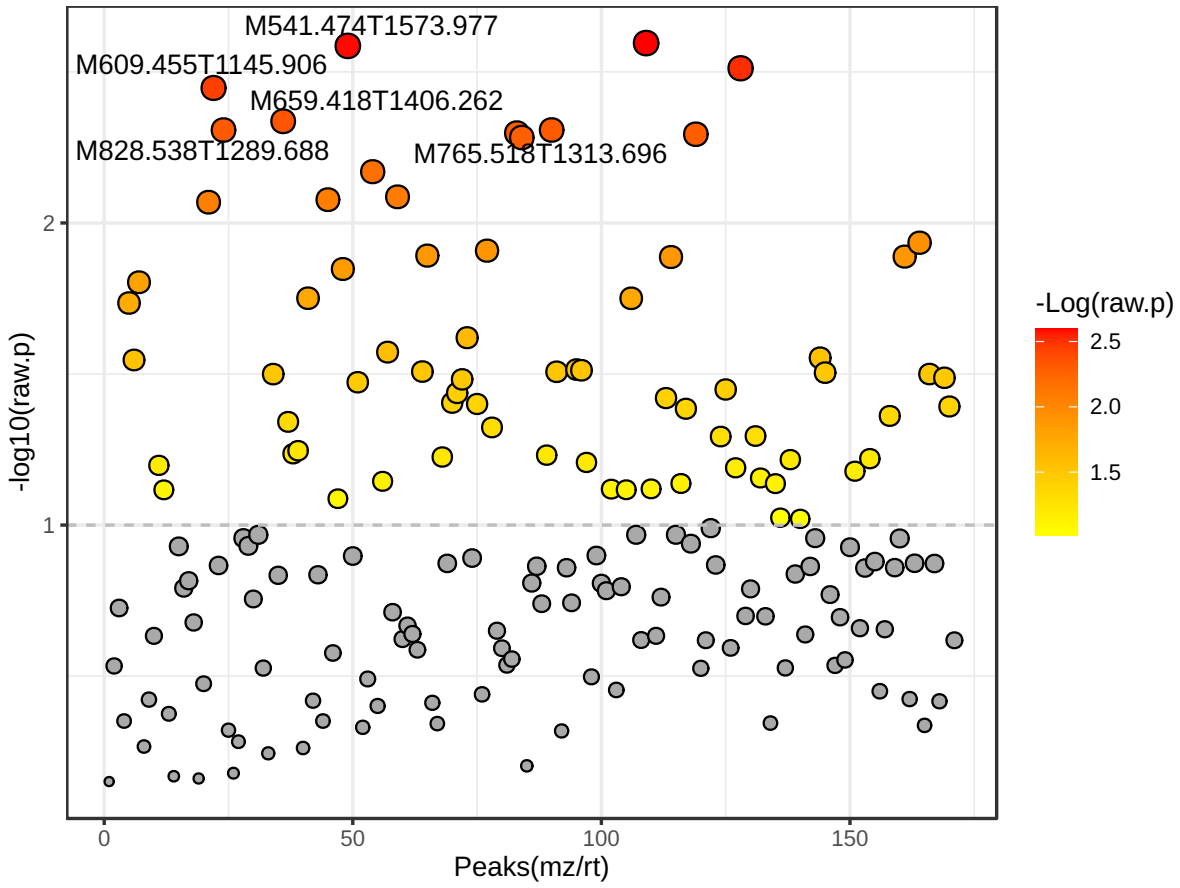

Figure 3: Important features selected by t-tests with threshold 0.1. The red circles represent features above the threshold. Note the p values are transformed by -log10 so that the more significant features (with smaller p values) will be plotted higher on the graph.

Table 3: Top 50 features identified by t-tests

	Peaks(mz/rt)	t.stat	p.value	-log10(p)	FDR
1	M330.263T1492.022	9.8802	0.0025391	2.5953	0.089203
2	M541.474T1573.977	-10.388	0.0025937	2.5861	0.089203
3	M619.438T1245.233	9.2813	0.0030763	2.512	0.089203
4	M609.455T1145.906	8.8009	0.0035742	2.4468	0.089203
5	M659.418T1406.262	-7.7786	0.0046079	2.3365	0.089203
6	M487.775T1256.694	-13.002	0.0049232	2.3078	0.089203
7	M828.538T1289.688	8.3988	0.0049255	2.3076	0.089203
8	M765.518T1313.696	8.0693	0.0050406	2.2975	0.089203
9	M951.603T1424.253	-9.0255	0.0050858	2.2936	0.089203
10	M415.326T1295.888	7.6102	0.0052165	2.2826	0.089203
11	M537.533T1295.888	8.6627	0.0067652	2.1697	0.10427
12	M535.409T1440.116	-7.0545	0.0081941	2.0865	0.10427
13	M329.157T1608.526	-6.6677	0.0083738	2.0771	0.10427
14	M343.224T1621.712	-6.2478	0.0085368	2.0687	0.10427
15	M492.408T1557.26	-5.8304	0.011626	1.9346	0.11668
16	M682.262T843.183	5.4153	0.012351	1.9083	0.11668
17	M577.49T1591.335	-5.3794	0.012827	1.8919	0.11668
18	M534.444T1557.26	-6.0235	0.012922	1.8887	0.11668
19	M549.431T1507.08	-5.5	0.012964	1.8873	0.11668
20	M423.212T1319.031	5.2077	0.014203	1.8476	0.12144
21	M423.198T1506.423	7.7981	0.015712	1.8038	0.12794
22	M423.211T1338.363	4.7509	0.017745	1.7509	0.13114
23	M425.215T1355.365	5.5693	0.017757	1.7506	0.13114
24	M985.571T1146.934	4.6868	0.018406	1.7351	0.13114
25	M479.188T913.185	-5.5146	0.023975	1.6203	0.15152
26	M495.298T1036.333	5.8858	0.026762	1.5725	0.15152
27	M471.796T1446.028	5.2859	0.027949	1.5536	0.15152
28	M986.591T1195.638	4.2793	0.028414	1.5465	0.15152
29	M425.216T1401.37	3.8713	0.030563	1.5148	0.15152
30	M430.319T1557.453	-5.4451	0.030731	1.5124	0.15152
31	M808.52T1220.972	3.8782	0.031036	1.5081	0.15152
32	M677.438T1339.916	-3.9886	0.031085	1.5075	0.15152
33	M339.172T1490.329	-5.0149	0.031292	1.5046	0.15152
34	M628.514T1473.756	-4.358	0.03163	1.4999	0.15152
35	M441.224T1179.269	10.785	0.031645	1.4997	0.15152
36	M465.265T1237.073	3.9131	0.032542	1.4875	0.15152
37	M504.265T1099.137	5.3657	0.032936	1.4823	0.15152
38	M847.517T1163.004	5.3027	0.03367	1.4728	0.15152
39	M629.529T1601.79	-3.7891	0.035578	1.4488	0.15599
40	M605.427T1212.039	3.6493	0.036531	1.4373	0.15617
41	M523.599T1382.768	4.9632	0.037984	1.4204	0.15659
42	M374.303T1319.161	4.542	0.039369	1.4048	0.15659
43	M821.505T1179.269	4.335	0.039737	1.4008	0.15659
44	M669.562T1624.559	-3.6363	0.040499	1.3926	0.15659
45	M576.484T1557.453	-4.4501	0.041208	1.385	0.15659
46	M629.509T1624.418	-3.6773	0.043521	1.3613	0.16179
47	M841.471T1389.363	-4.3945	0.045535	1.3417	0.16567
48	M576.484T1473.551	-3.3108	0.047514	1.3232	0.16927
49	M295.151T1624.873	-3.5995	0.050734	1.2947	0.17418
50	M330.263T1439.913	3.8479	0.05093	1.293	0.17418

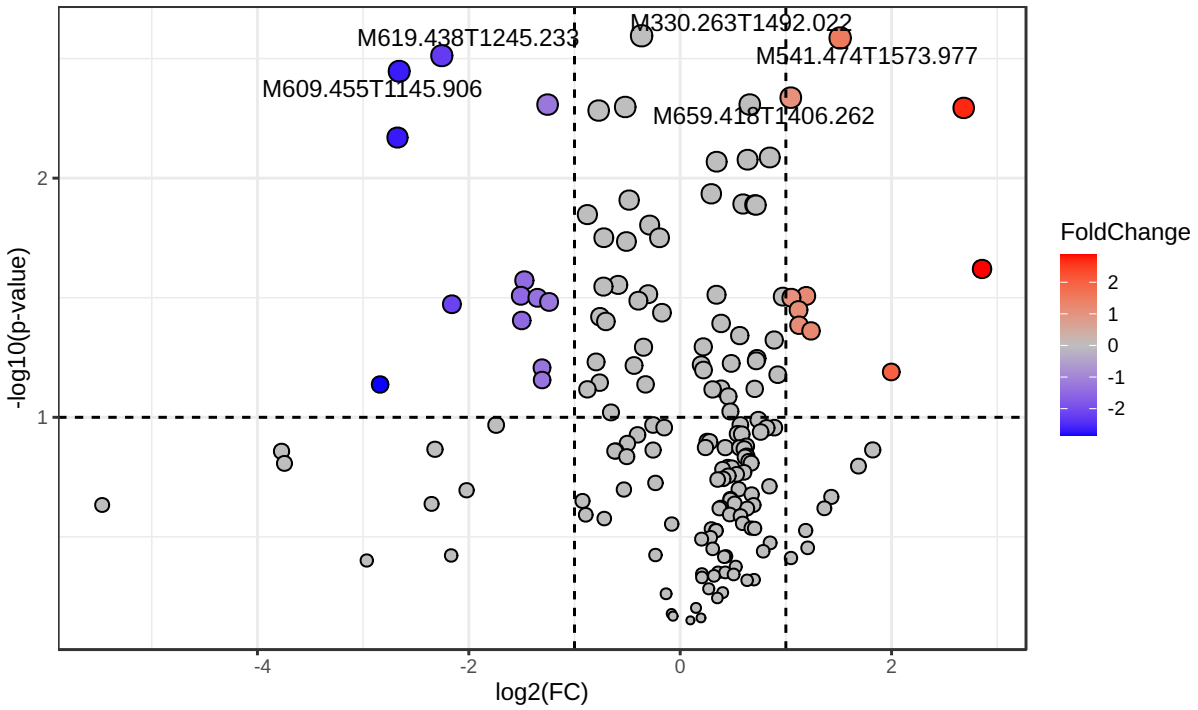

Figure 4: Important features selected by volcano plot with fold change threshold (x) 2 and t-tests threshold (y) 0.1. The red circles represent features above the threshold. Note both fold changes and p values are log transformed. The further its position away from the (0,0), the more significant the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

Table 4: Important features identified by volcano plot

	Peaks(mz/rt)	FC	log2(FC)	raw.pval	-log10(p)
1	M541.474T1573.977	2.8551	1.5136	0.0025937	2.5861
2	M619.438T1245.233	0.20941	-2.2556	0.0030763	2.512
3	M609.455T1145.906	0.15835	-2.6588	0.0035742	2.4468
4	M659.418T1406.262	2.0654	1.0465	0.0046079	2.3365
5	M828.538T1289.688	0.41908	-1.2547	0.0049255	2.3076
6	M951.603T1424.253	6.4202	2.6826	0.0050858	2.2936
7	M537.533T1295.888	0.15658	-2.675	0.0067652	2.1697
8	M479.188T913.185	7.2402	2.856	0.023975	1.6203
9	M495.298T1036.333	0.3599	-1.4744	0.026762	1.5725
10	M808.52T1220.972	0.3517	-1.5076	0.031036	1.5081
11	M677.438T1339.916	2.2849	1.1921	0.031085	1.5075
12	M628.514T1473.756	2.0742	1.0525	0.03163	1.4999
13	M441.224T1179.269	0.39209	-1.3507	0.031645	1.4997
14	M504.265T1099.137	0.42327	-1.2403	0.032936	1.4823
15	M847.517T1163.004	0.22375	-2.1601	0.03367	1.4728
16	M629.529T1601.79	2.1732	1.1199	0.035578	1.4488
17	M374.303T1319.161	0.35373	-1.4993	0.039369	1.4048
18	M576.484T1557.453	2.1801	1.1244	0.041208	1.385
19	M629.509T1624.418	2.3588	1.238	0.043521	1.3613
20	M392.284T1436.378	0.40403	-1.3075	0.062039	1.2073
21	M576.533T1641.124	3.9926	1.9973	0.064628	1.1896
22	M772.55T1170.538	0.40431	-1.3065	0.069835	1.1559
23	M722.581T1518.016	0.1397	-2.8396	0.072912	1.1372

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

PLS is a supervised method that uses multivariate regression techniques to extract via linear combination of original variables (X) the information that can predict the class membership (Y). The PLS regression is performed using the `plsr` function provided by R `pls` package[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 5 shows the overview of scores plots; Figure 6 shows the 2-D scores plot between selected components; Figure 7 shows the 3-D scores plot between selected components; Figure 8 shows the loading plot between the selected components;Figure 9 shows the classification performance with different number of components; Figure 10 shows the results of permutation test for model validation; Figure 11 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

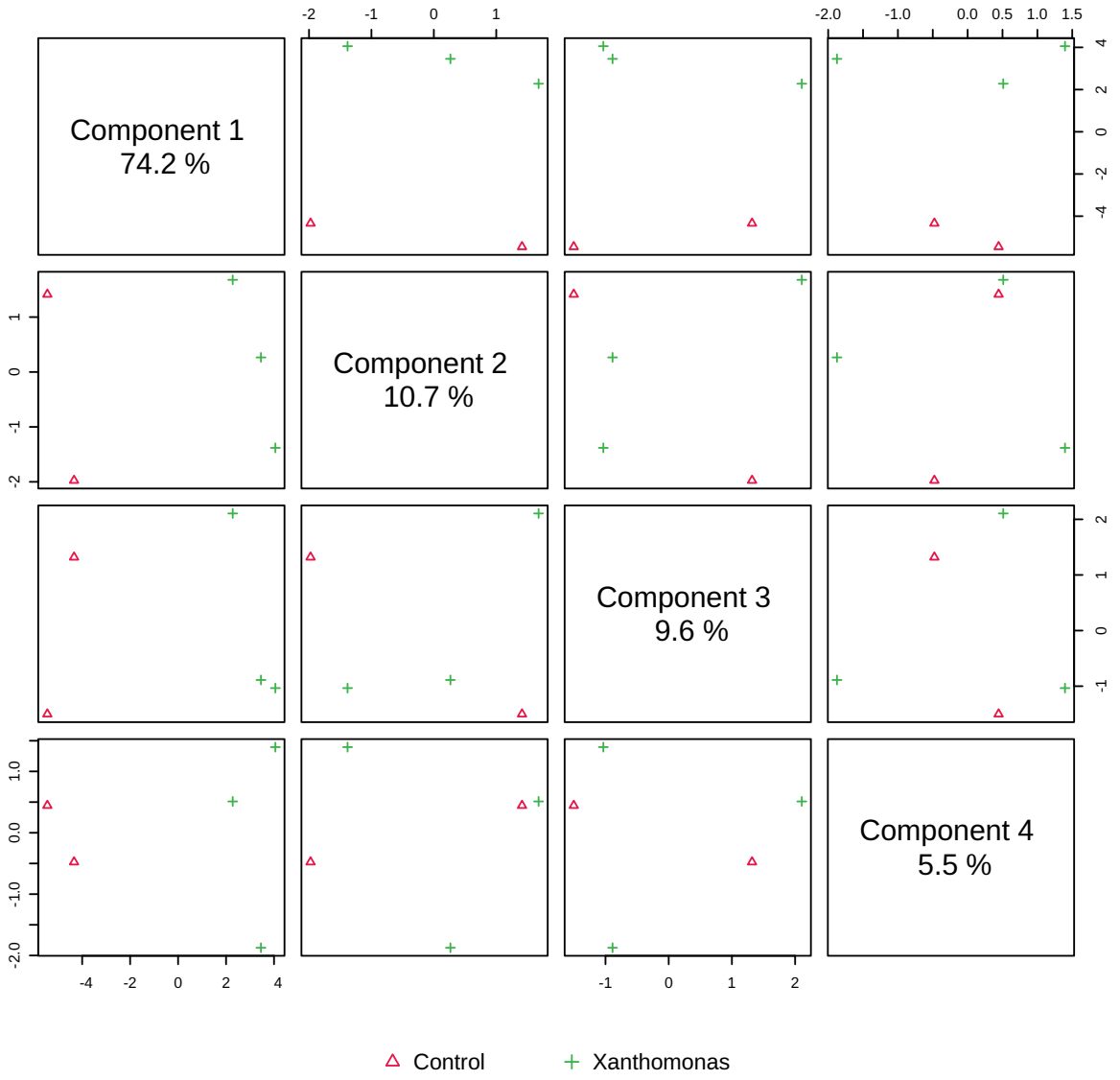

Figure 5: Pairwise scores plots between the selected components. The explained variance of each component is shown in the corresponding diagonal cell.

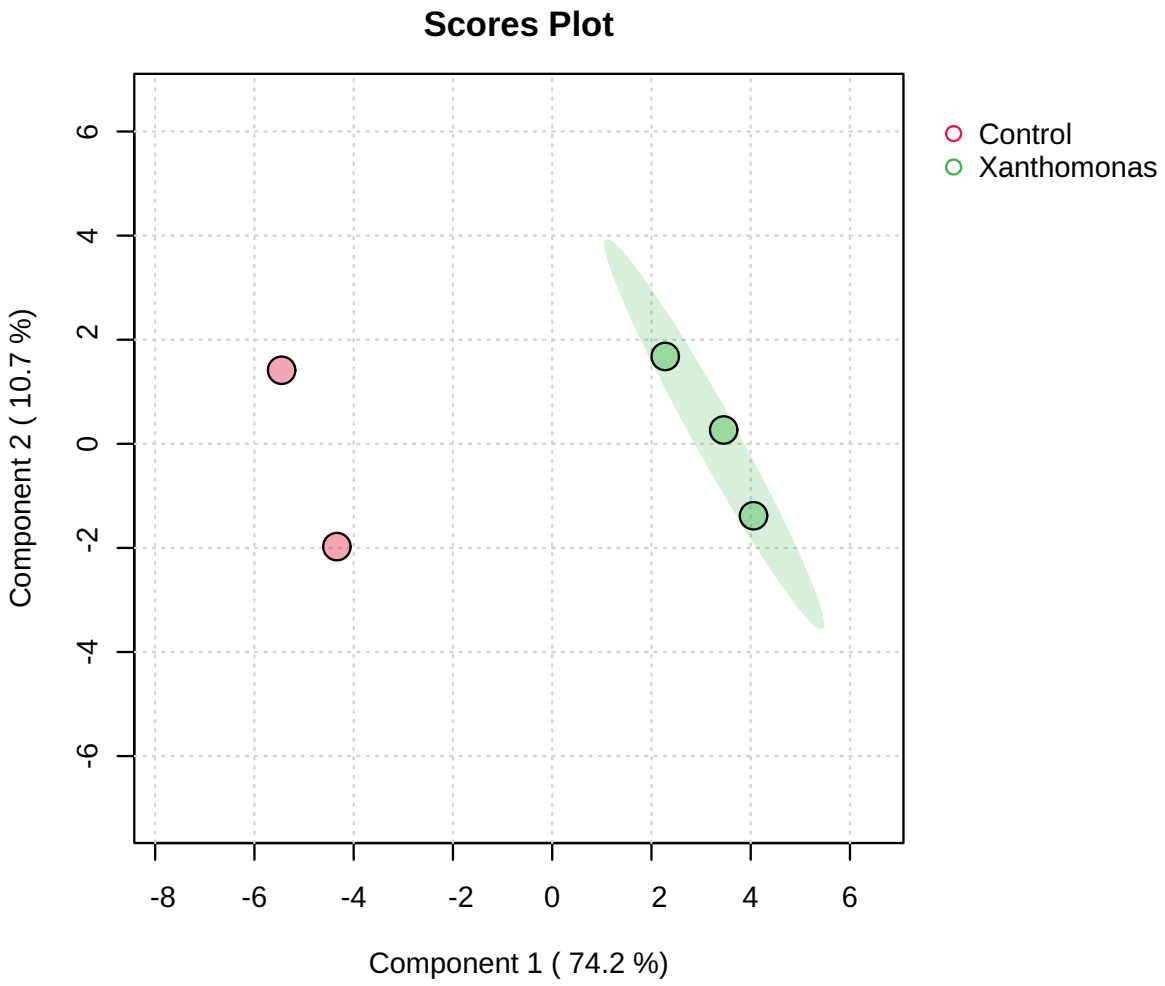

Figure 6: Scores plot between the selected PCs. The explained variances are shown in brackets.

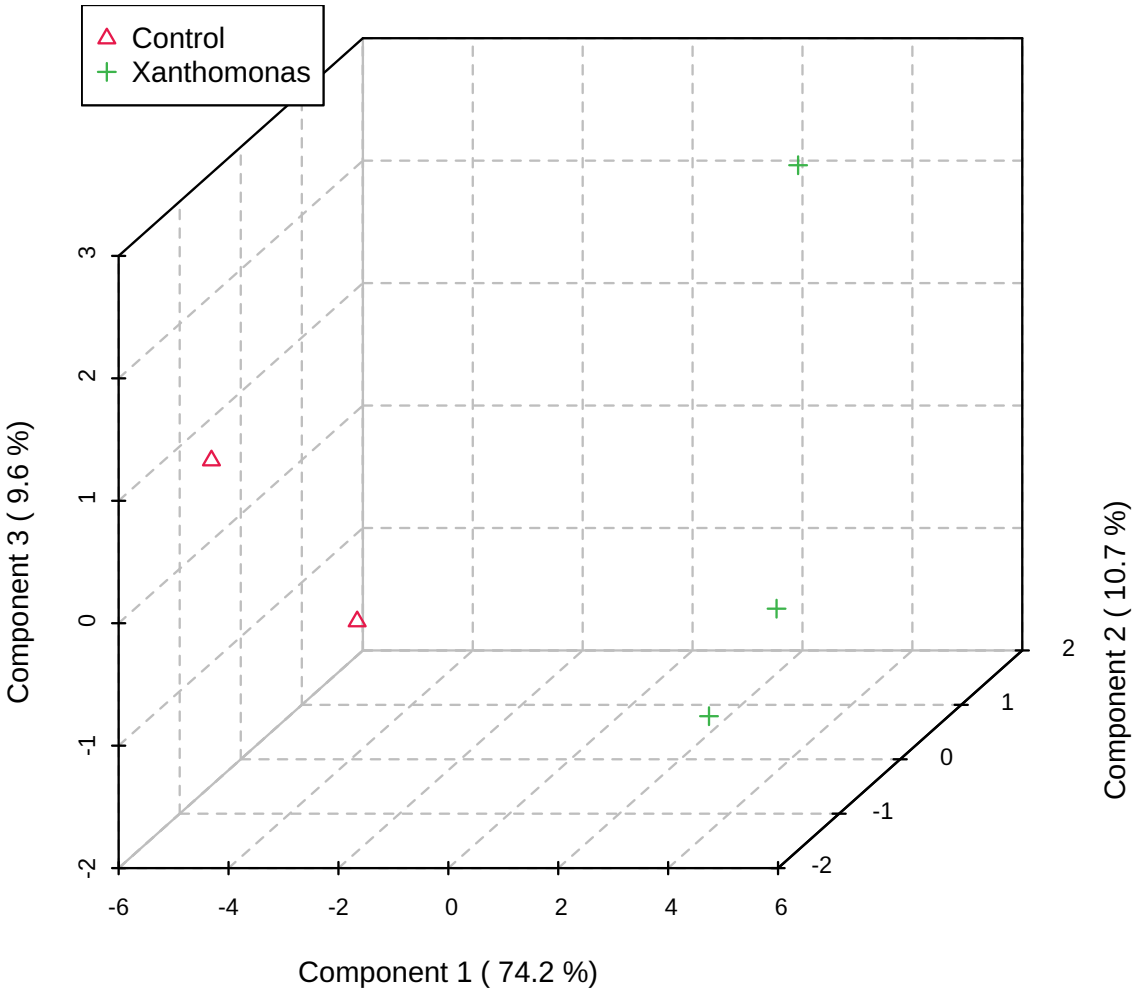

Figure 7: 3D scores plot between the selected PCs. The explained variances are shown in brackets.

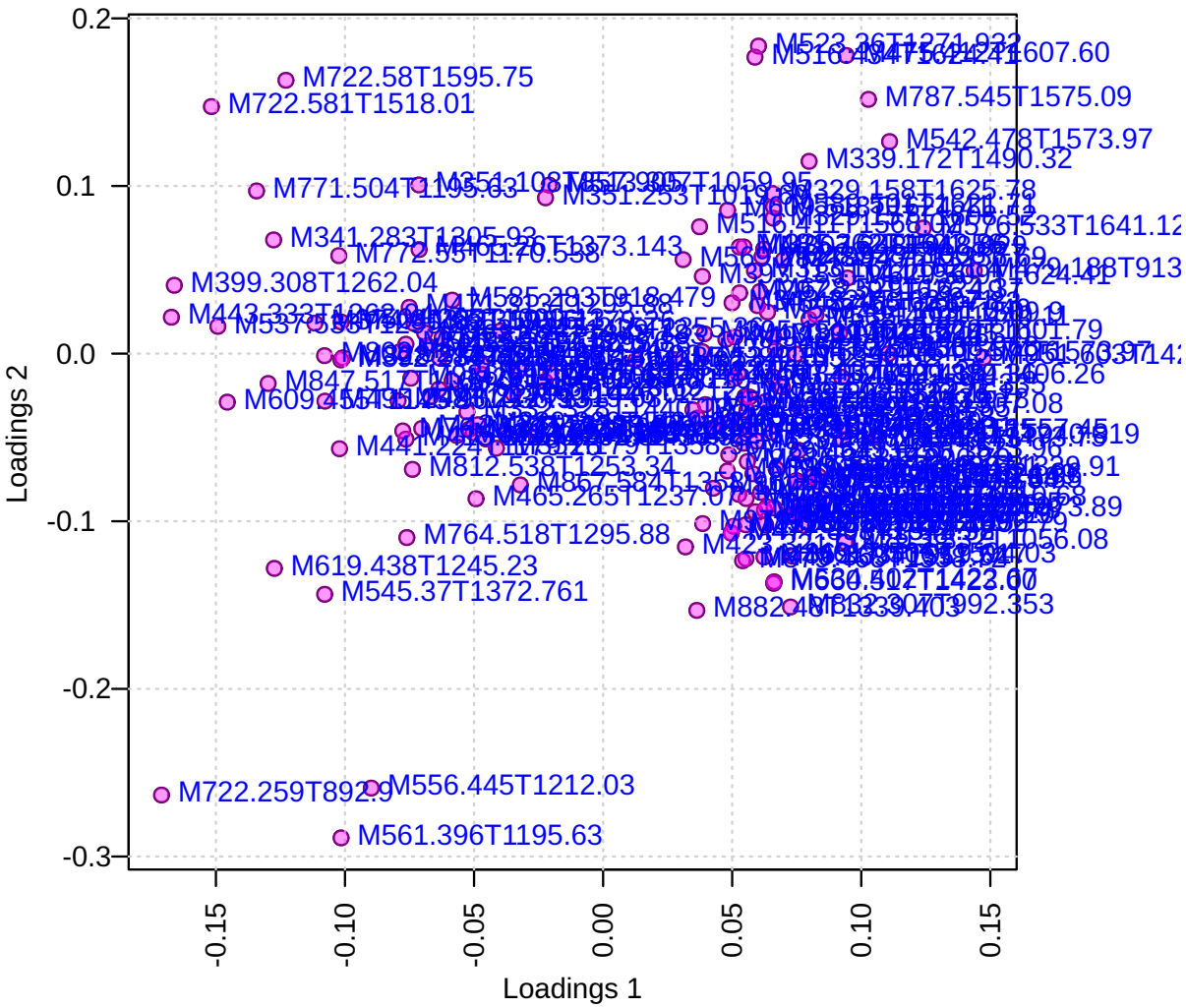

Figure 8: Loadings plot between the selected PCs.

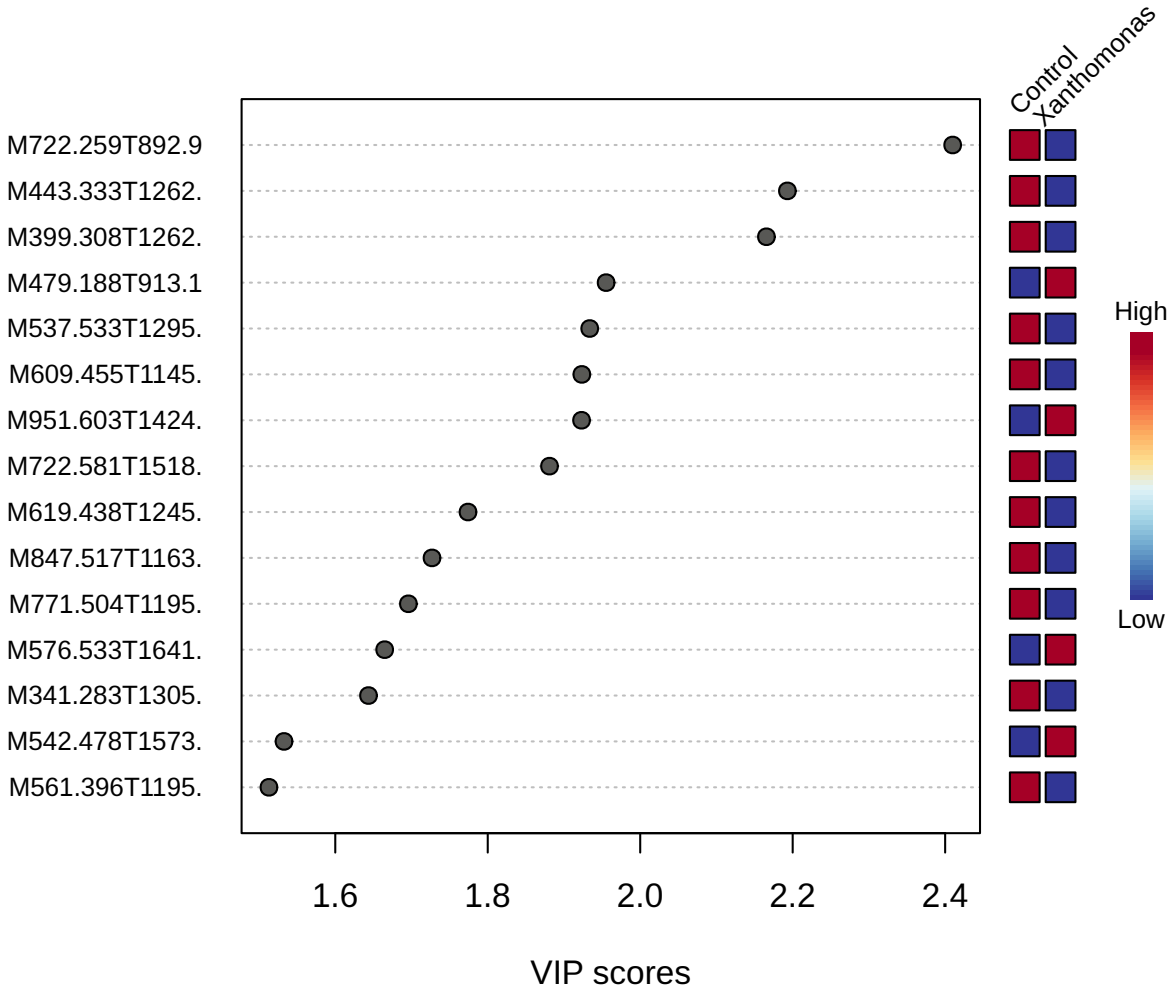

Figure 9: Important features identified by PLS-DA. The colored boxes on the right indicate the relative concentrations of the corresponding metabolite in each group under study.

2.3 Hierarchical Clustering

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

Hierachical clustering is performed with the `hclust` function in package `stat`. Figure 12 shows the clustering result in the form of a dendrogram. Figure 13 shows the clustering result in the form of a heatmap.

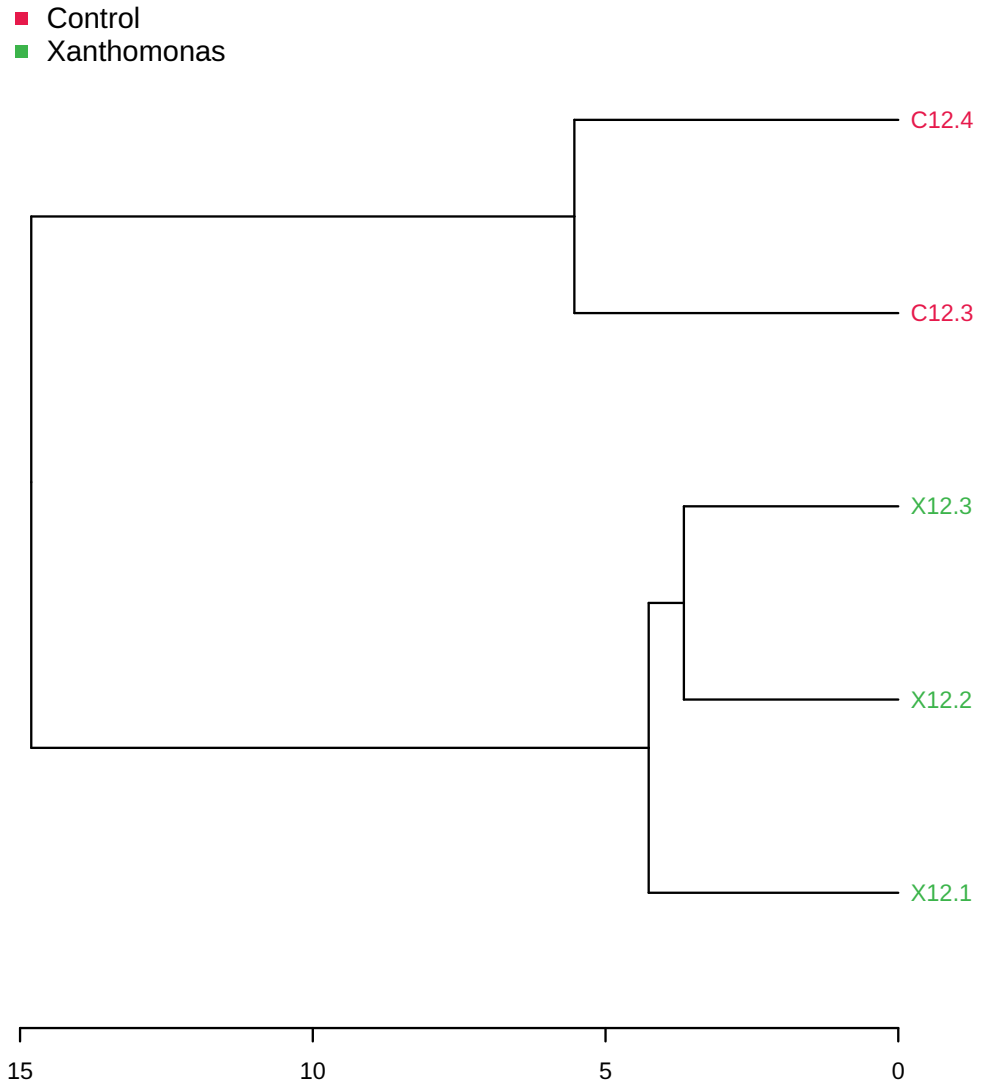

Figure 10: Clustering result shown as dendrogram (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

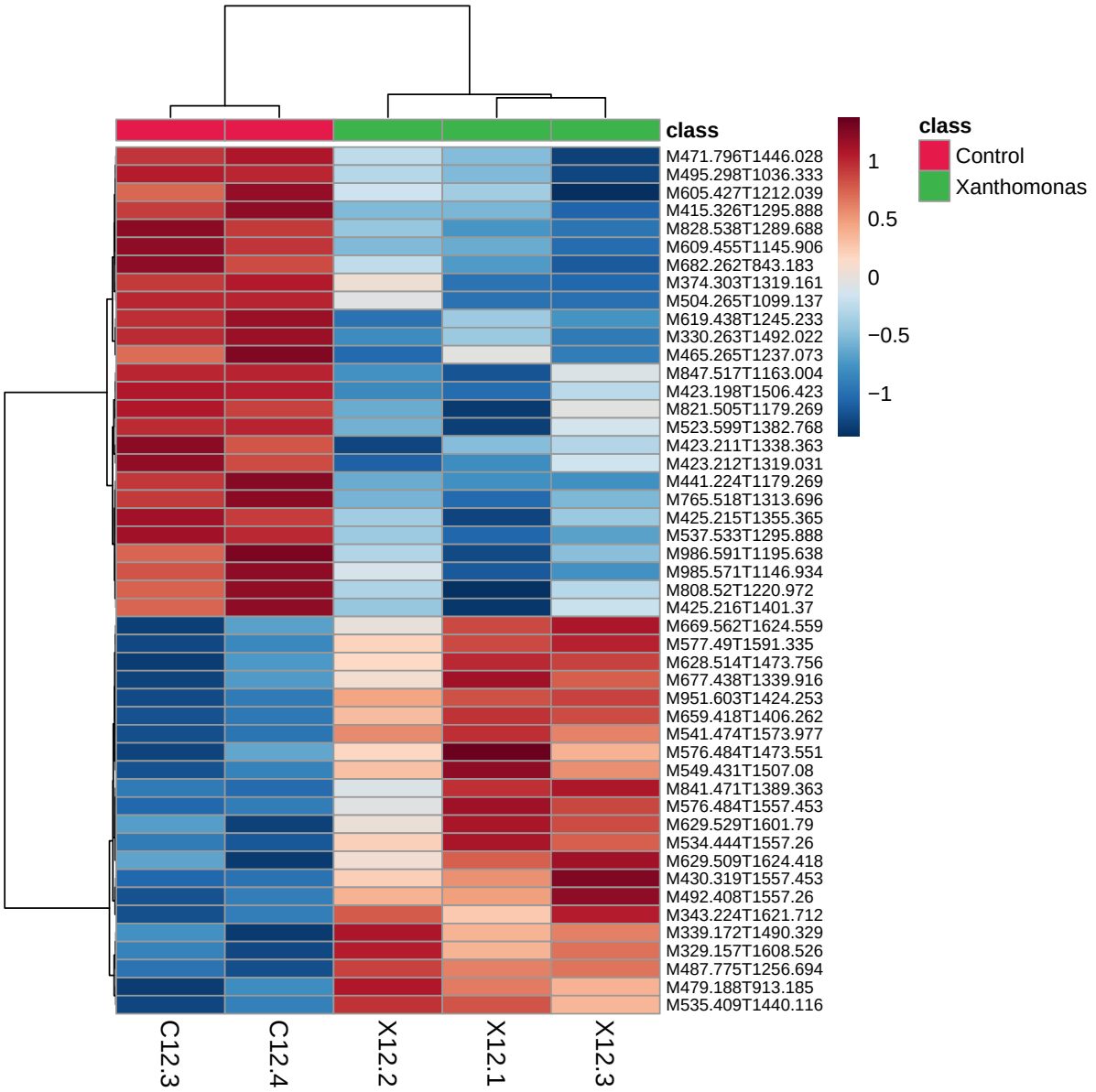

Figure 11: Clustering result shown as heatmap (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

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(\"C12.1\")"
[11] "grp.nm.vec <- c(\"Control\",\"Xanthomonas\")"
[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)"
[27] "mSet<-Volcano.Anal(mSet, FALSE, 2.0, 1, F, 0.1, FALSE, \"raw\")"
[28] "mSet<-PlotVolcano(mSet, \"volcano_1_\",1, 0, \"png\", 72, width=NA)"
[29] "mSet<-PLSR.Anal(mSet, reg=TRUE)"
[30] "mSet<-PlotPLSPairSummary(mSet, \"pls_pair_0_\", \"png\", 72, width=NA, 4)"
[31] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[32] "mSet<-PlotPLS3DScoreImg(mSet, \"pls_score3d_0_\", \"png\", 72, width=NA, 1,2,3, 40)"
[33] "mSet<-PlotPLSLoading(mSet, \"pls_loading_0_\", \"png\", 72, width=NA, 1, 2);"
[34] "mSet<-PlotPLS3DLoading(mSet, \"pls_loading3d_0_\", \"json\", 1,2,3)"
[35] "mSet<-PlotPLS.Imp(mSet, \"pls_imp_0_\", \"png\", 72, width=NA, \"vip\", \"Comp. 1\", 15,FALSE)
[36] "mSet<-PlotHCTree(mSet, \"tree_0_\", \"png\", 72, width=NA, \"euclidean\", \"ward.D\")"
[37] "mSet<-PlotHeatMap(mSet, \"heatmap_1_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\
[38] "mSet<-PlotHeatMap(mSet, \"heatmap_2_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\
[39] "mSet<-PlotSubHeatMap(mSet, \"heatmap_3_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclide
[40] "mSet<-SaveTransformedData(mSet)"
[41] "mSet<-PreparePDFReport(mSet, \"guest92591215451044746l3\")\n"
```

The report was generated on Wed Mar 6 07:40:24 2024 with R version 4.2.2 (2022-10-31), OS system: Linux, version: -Ubuntu SMP Mon Oct 2 21:09:21 UTC 2023 .