Metabolomic Data Analysis with MetaboAnalyst 6.0

Name: guest11701350932969246681

March 15, 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 807 (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)
X02.C12.2.neg	799	8	807
X18.C12.3.neg	782	25	807
X33.C12.4.neg	797	10	807
X12.D12.2.neg	804	3	807
X28.D12.3.neg	806	1	807
X44.D12.1.neg	287	520	807
X52.D12.4.neg	805	2	807
X05.F12.4.neg	806	1	807
X22.F12.1.neg	802	5	807
X38.F12.2.neg	806	1	807
X43.F12.3.neg	805	2	807
X53.Blank.neg	367	440	807
X10.QC1.neg	803	4	807
X24.QC.2.neg	807	0	807
X39.QC3.neg	797	10	807
X09.X12.3.neg	303	504	807
X32.X12.2.neg	798	9	807
X41.X12.1.neg	802	5	807

[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

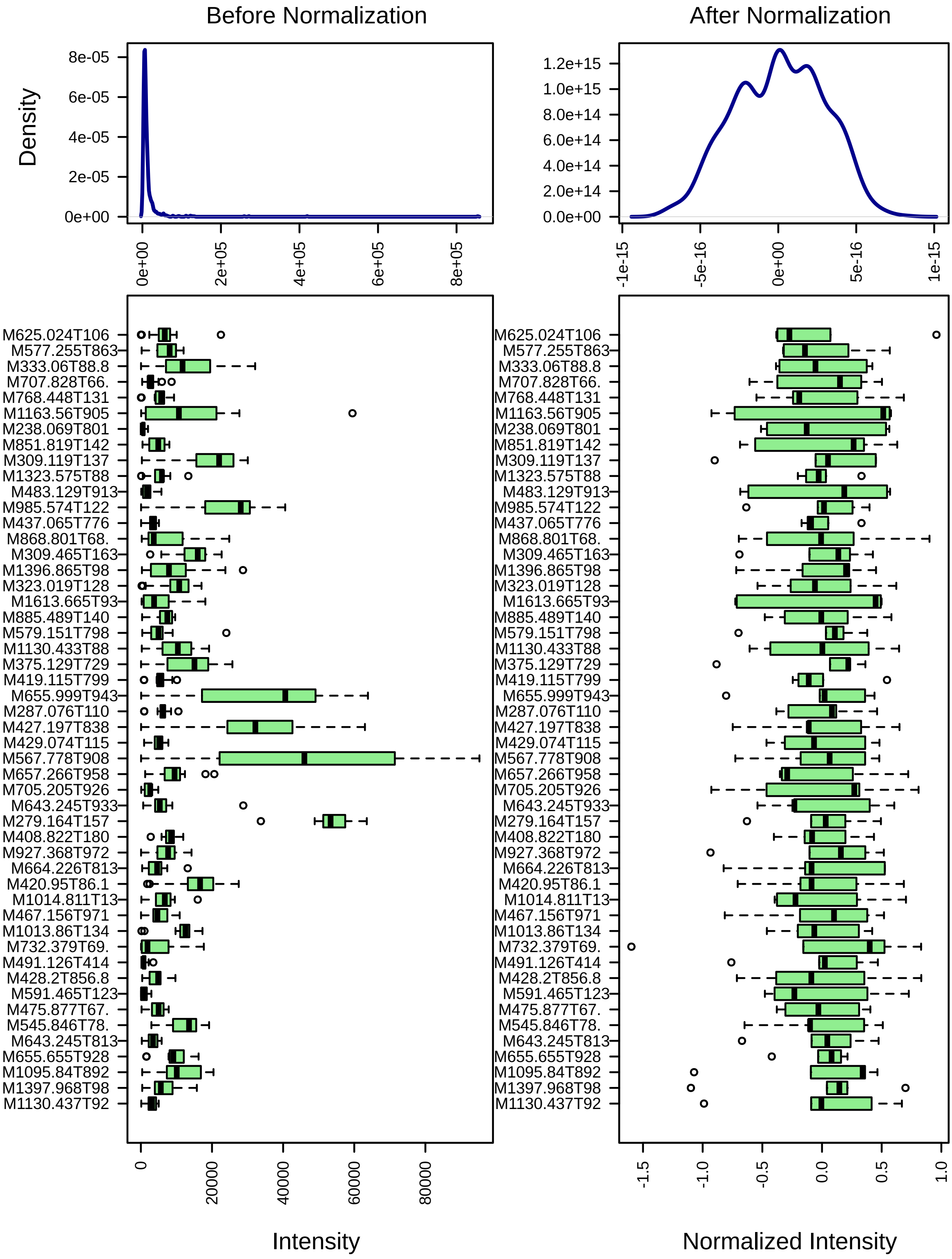

Figure 1: Box plots and kernel density plots before and after normalization. The boxplots show at most 50 features due to space limit. The density plots are based on all samples.

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.

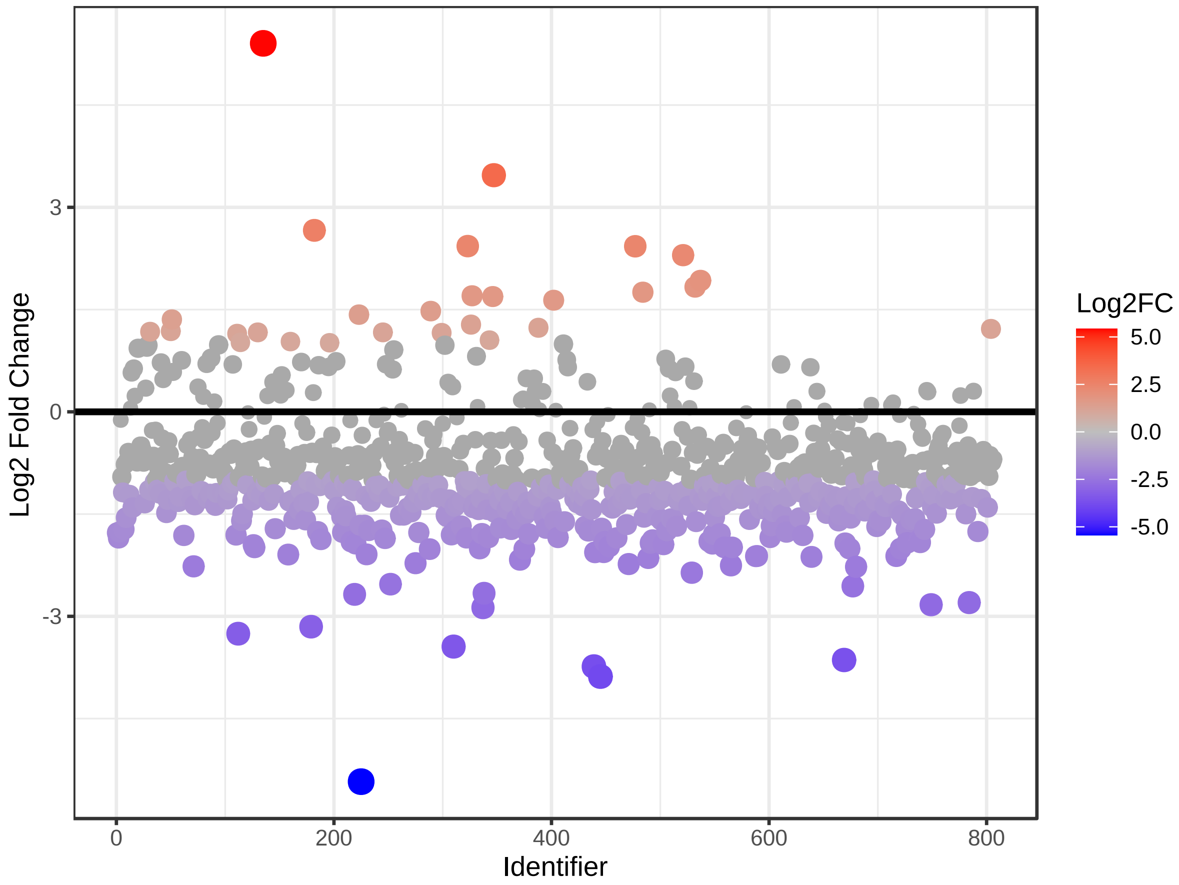

Figure 2: Important features selected by fold-change analysis with threshold 2. The red circles represent features above the threshold. Note the values are on log scale, so that both up-regulated and down-regulated features can be plotted in a symmetrical way

Table 2: Top 50 features identified by fold change analysis

	Peaks(mz/rt)	Fold Change	log2(FC)
1	M204.085T413.719	0.023274	-5.4252
2	M1163.085T907.707	42.386	5.4055
3	M684.191T618.708	0.067774	-3.8831
4	M272.073T418.674	0.075039	-3.7362
5	M413.179T861.909	0.080186	-3.6405
6	M1163.56T905.015	11.091	3.4714
7	M410.078T615.621	0.091904	-3.4437
8	M279.139T1150.344	0.10477	-3.2547
9	M433.039T456.919	0.1125	-3.152
10	M387.034T831.689	0.13662	-2.8717
11	M319.076T917.392	0.14068	-2.8296
12	M775.338T1039.594	0.14376	-2.7983
13	M311.104T1075.82	0.15628	-2.6778
14	M1134.038T907.684	6.3296	2.6621
15	M455.021T825.04	0.15803	-2.6617
16	M367.058T823.98	0.16997	-2.5566
17	M1014.419T974.744	0.17334	-2.5283
18	M240.933T64.89	5.3931	2.4311
19	M1193.041T904.79	5.3845	2.4288
20	M516.992T825.04	0.19482	-2.3598
21	M1192.04T901.764	4.9192	2.2984
22	M487.146T735.656	0.20683	-2.2735
23	M489.058T766.222	0.20781	-2.2667
24	M333.041T419.397	0.20992	-2.2521
25	M449.004T832.738	0.21272	-2.233
26	M1063.398T908.83	0.21458	-2.2204
27	M365.052T459.24	0.2223	-2.1694
28	M1044.425T908.907	0.22615	-2.1447
29	M850.805T68.208	0.22878	-2.128
30	M712.22T783.012	0.23037	-2.118
31	M343.103T489.112	0.23069	-2.116
32	M469.036T876.272	0.23087	-2.1148
33	M293.118T1042.937	0.23429	-2.0936
34	M469.208T948.545	0.23473	-2.0909
35	M237.04T858.99	0.23971	-2.0607
36	M1176.528T983.06	0.23994	-2.0592
37	M430.136T769.014	0.24757	-2.0141
38	M505.029T765.672	0.24777	-2.0129
39	M203.083T418.775	0.24891	-2.0063
40	M333.062T954.304	0.24893	-2.0062
41	M389.108T489.956	0.2491	-2.0052
42	M367.103T818.837	0.25197	-1.9887
43	M333.191T983.058	0.25198	-1.9886
44	M428.2T856.891	0.25201	-1.9885
45	M895.29T938.858	0.2542	-1.976
46	M923.402T931.508	0.25839	-1.9524
47	M420.999T106.905	0.25968	-1.9452
48	M361.323T326.759	0.26131	-1.9362
49	M380.071T778.258	0.26153	-1.935
50	M807.007T938.915	0.26293	-1.9272

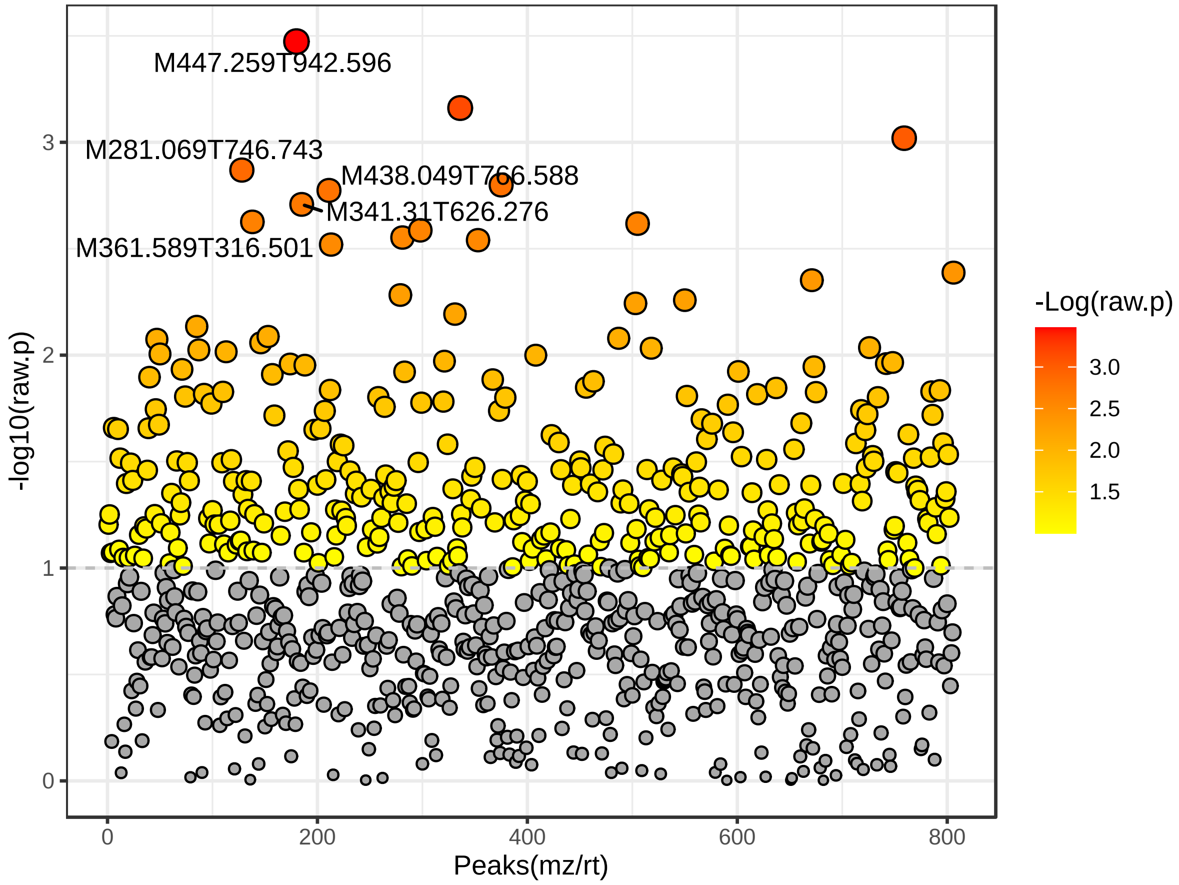

Figure 3: Important features selected by t-tests with threshold 0.1. The red circles represent features above the threshold. Note the p values are transformed by -log10 so that the more significant features (with smaller p values) will be plotted higher on the graph.

Table 3: Top 50 features identified by t-tests

	Peaks(mz/rt)	t.stat	p.value	-log10(p)	FDR
1	M447.259T942.596	-33.614	0.00033632	3.4732	0.18743
2	M341.535T617.585	-30.26	0.00069026	3.161	0.18743
3	M470.098T771.956	-13.144	0.00095659	3.0193	0.18743
4	M281.069T746.743	-11.701	0.0013505	2.8695	0.18743
5	M505.029T765.672	-17.624	0.0015871	2.7994	0.18743
6	M438.049T766.588	-21.844	0.0016824	2.7741	0.18743
7	M341.31T626.276	-11.739	0.0019575	2.7083	0.18743
8	M361.589T316.501	-15.864	0.0023652	2.6261	0.18743
9	M1136.99T908.053	9.866	0.0024105	2.6179	0.18743
10	M533.169T900.984	-11.47	0.0025914	2.5865	0.18743
11	M413.081T316.361	-9.1255	0.0028016	2.5526	0.18743
12	M341.259T618.637	-13.582	0.0028826	2.5402	0.18743
13	M407.045T415.442	-8.8691	0.0030231	2.5195	0.18743
14	M372.9T1734.569	-11.43	0.0040949	2.3878	0.22153
15	M359.022T90.408	-8.5501	0.0044447	2.3522	0.22153
16	M439.083T756.683	-10.135	0.0052169	2.2826	0.22153
17	M409.075T617.585	-10.274	0.0055164	2.2583	0.22153
18	M420.999T106.905	-7.3432	0.0057101	2.2434	0.22153
19	M1094.946T896.052	8.011	0.0064136	2.1929	0.22153
20	M405.08T802.267	-6.5542	0.0073322	2.1348	0.22153
21	M970.001T316.098	-7.8415	0.0081714	2.0877	0.22153
22	M475.366T1425.395	-9.4106	0.0083316	2.0793	0.22153
23	M653.891T318.024	-10.033	0.0084277	2.0743	0.22153
24	M727.283T864.823	-7.2308	0.0087463	2.0582	0.22153
25	M317.064T416.942	-8.5632	0.0092145	2.0355	0.22153
26	M545.298T995.647	-9.9017	0.009277	2.0326	0.22153
27	M385.077T746.611	7.0722	0.0094472	2.0247	0.22153
28	M579.301T894.416	-6.1833	0.0096411	2.0159	0.22153
29	M590.775T907.789	7.2775	0.0098716	2.0056	0.22153
30	M631.833T319.717	-6.4086	0.010004	1.9998	0.22153
31	M1097.514T1402.931	-7.3436	0.01067	1.9719	0.22153
32	M562.892T1151.046	-6.5194	0.010821	1.9657	0.22153
33	M265.261T1518.666	-7.142	0.010957	1.9603	0.22153
34	M437.044T765.875	-5.9525	0.011003	1.9585	0.22153
35	M323.187T1016.246	-7.8078	0.011144	1.953	0.22153
36	M385.113T791.317	-6.7983	0.011335	1.9456	0.22153
37	M489.058T766.222	-9.1434	0.011662	1.9332	0.22153
38	M508.215T930.69	-5.8524	0.011924	1.9236	0.22153
39	M225.002T107.866	-6.6617	0.011977	1.9216	0.22153
40	M565.23T844.341	-6.7718	0.012307	1.9098	0.22153
41	M1053.272T316.465	-7.3806	0.012687	1.8966	0.22153
42	M970.768T318.573	-6.0766	0.013022	1.8853	0.22153
43	M544.284T936.329	-5.2908	0.013273	1.877	0.22153
44	M852.339T915.443	-5.1594	0.014188	1.8481	0.22153
45	M643.245T933.361	-5.7626	0.014261	1.8459	0.22153
46	M1013.583T1347.533	-5.9443	0.014574	1.8364	0.22153
47	M795.198T1006.041	-5.8124	0.014628	1.8348	0.22153
48	M429.074T115.482	-5.3898	0.014825	1.829	0.22153
49	M821.606T900.199	-8.0523	0.014875	1.8275	0.22153
50	M341.465T625.973	-8.0433	0.014926	1.8261	0.22153

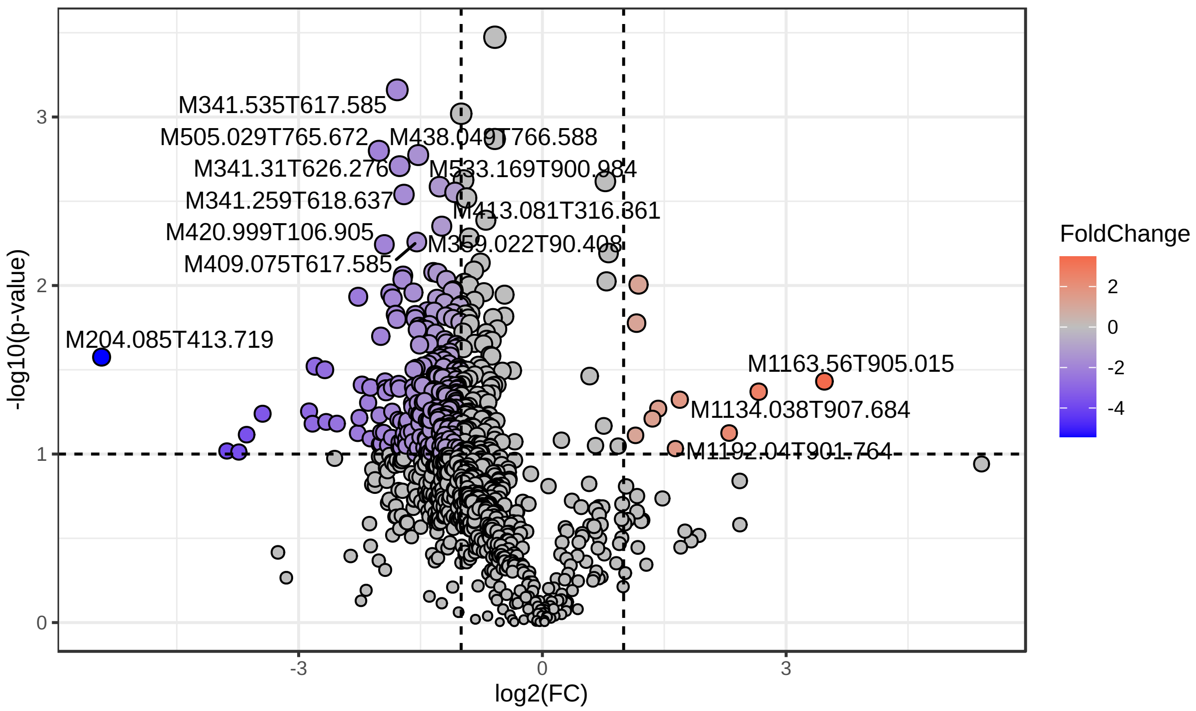

Figure 4: Important features selected by volcano plot with fold change threshold (x) 2 and t-tests threshold (y) 0.1. The red circles represent features above the threshold. Note both fold changes and p values are log transformed. The further its position away from the (0,0), the more significant the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

Table 4: Top 50 features identified by volcano plot

	Peaks(mz/rt)	FC	log2(FC)	raw.pval	-log10(p)
1	M341.535T617.585	0.28992	-1.7863	0.00069026	3.161
2	M505.029T765.672	0.24777	-2.0129	0.0015871	2.7994
3	M438.049T766.588	0.34666	-1.5284	0.0016824	2.7741
4	M341.31T626.276	0.29574	-1.7576	0.0019575	2.7083
5	M533.169T900.984	0.41571	-1.2664	0.0025914	2.5865
6	M413.081T316.361	0.47455	-1.0754	0.0028016	2.5526
7	M341.259T618.637	0.30699	-1.7037	0.0028826	2.5402
8	M359.022T90.408	0.4238	-1.2385	0.0044447	2.3522
9	M409.075T617.585	0.34252	-1.5457	0.0055164	2.2583
10	M420.999T106.905	0.25968	-1.9452	0.0057101	2.2434
11	M475.366T1425.395	0.39469	-1.3412	0.0083316	2.0793
12	M653.891T318.024	0.40894	-1.29	0.0084277	2.0743
13	M727.283T864.823	0.30472	-1.7145	0.0087463	2.0582
14	M317.064T416.942	0.30322	-1.7216	0.0092145	2.0355
15	M545.298T995.647	0.4404	-1.1831	0.009277	2.0326
16	M590.775T907.789	2.2713	1.1835	0.0098716	2.0056
17	M1097.514T1402.931	0.46865	-1.0934	0.01067	1.9719
18	M562.892T1151.046	0.46395	-1.108	0.010821	1.9657
19	M437.044T765.875	0.33286	-1.587	0.011003	1.9585
20	M323.187T1016.246	0.27324	-1.8718	0.011144	1.953
21	M489.058T766.222	0.20781	-2.2667	0.011662	1.9332
22	M508.215T930.69	0.2789	-1.8422	0.011924	1.9236
23	M225.002T107.866	0.40696	-1.297	0.011977	1.9216
24	M1053.272T316.465	0.43462	-1.2022	0.012687	1.8966
25	M544.284T936.329	0.49298	-1.0204	0.013273	1.877
26	M852.339T915.443	0.37447	-1.4171	0.014188	1.8481
27	M643.245T933.361	0.3975	-1.331	0.014261	1.8459
28	M1013.583T1347.533	0.46709	-1.0982	0.014574	1.8364
29	M821.606T900.199	0.2859	-1.8064	0.014875	1.8275
30	M341.465T625.973	0.33879	-1.5615	0.014926	1.8261
31	M1069.501T1402.821	0.43853	-1.1892	0.015224	1.8175
32	M269.103T792.774	0.46581	-1.1022	0.015751	1.8027
33	M238.069T801.926	0.33817	-1.5642	0.015765	1.8023
34	M728.286T863.521	0.28902	-1.7908	0.015831	1.8005
35	M669.182T316.584	0.49573	-1.0124	0.016525	1.7819
36	M1095.01T894.541	2.233	1.159	0.016719	1.7768
37	M489.187T785.097	0.38132	-1.3909	0.017098	1.767
38	M398.263T1317.762	0.34909	-1.5183	0.017493	1.7571
39	M650.407T1378.088	0.35808	-1.4816	0.017955	1.7458
40	M968.386T889.852	0.37167	-1.4279	0.018258	1.7385
41	M529.154T821.794	0.34434	-1.5381	0.018275	1.7381
42	M483.209T897.853	0.40196	-1.3149	0.019203	1.7166
43	M367.103T818.837	0.25197	-1.9887	0.02	1.699
44	M871.351T913.248	0.4349	-1.2013	0.020988	1.678
45	M511.24T944.134	0.44076	-1.1819	0.021947	1.6586
46	M999.268T316.316	0.38003	-1.3958	0.022148	1.6547
47	M836.522T1479.692	0.47964	-1.06	0.022407	1.6496
48	M415.161T821.344	0.35063	-1.512	0.022519	1.6475
49	M567.171T792.05	0.49186	-1.0237	0.023029	1.6377
50	M401.049T776.54	0.47757	-1.0662	0.023574	1.6276

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.

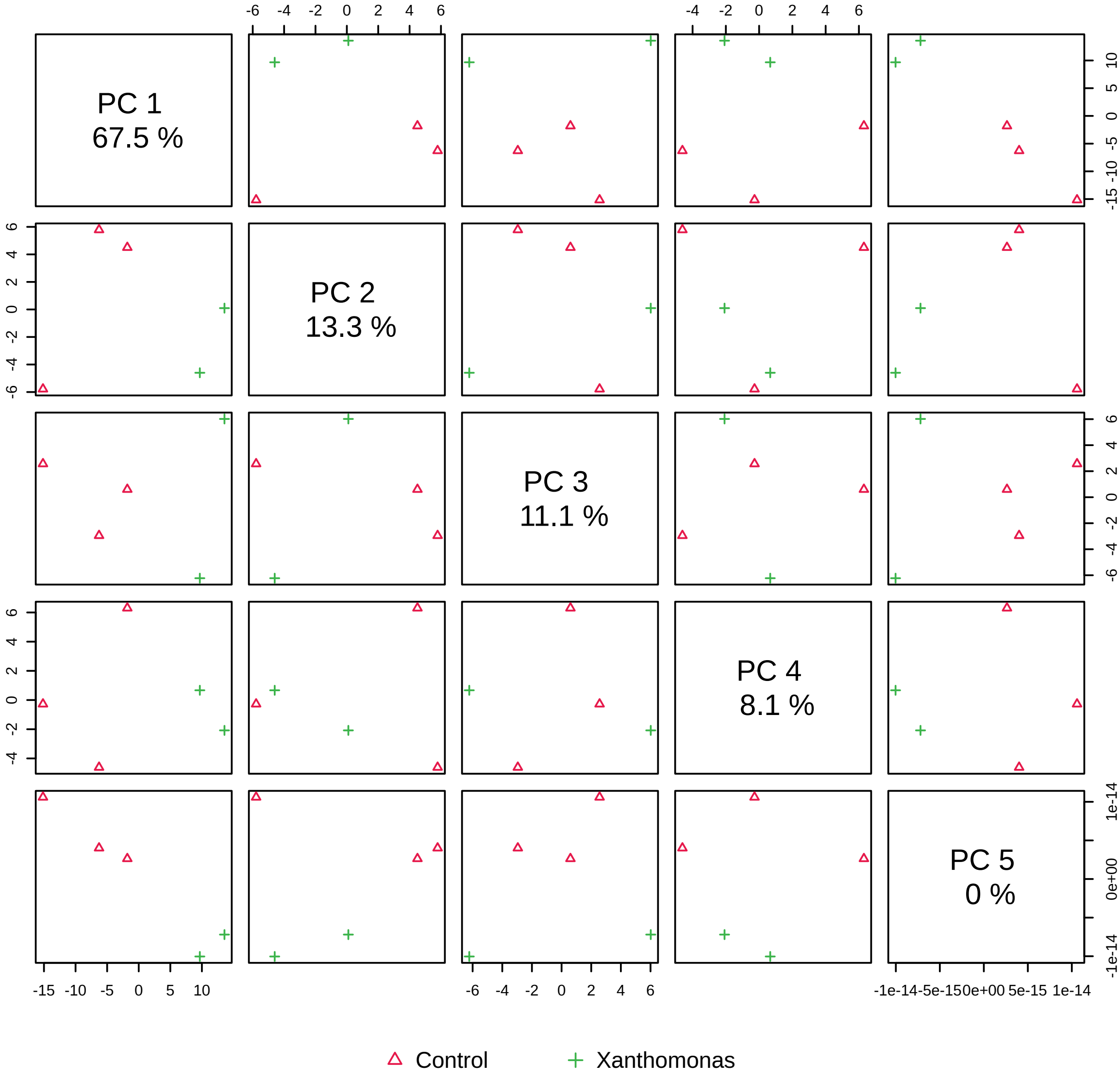

Figure 5: Pairwise score plots between the selected PCs. The explained variance of each PC is shown in the corresponding diagonal cell.

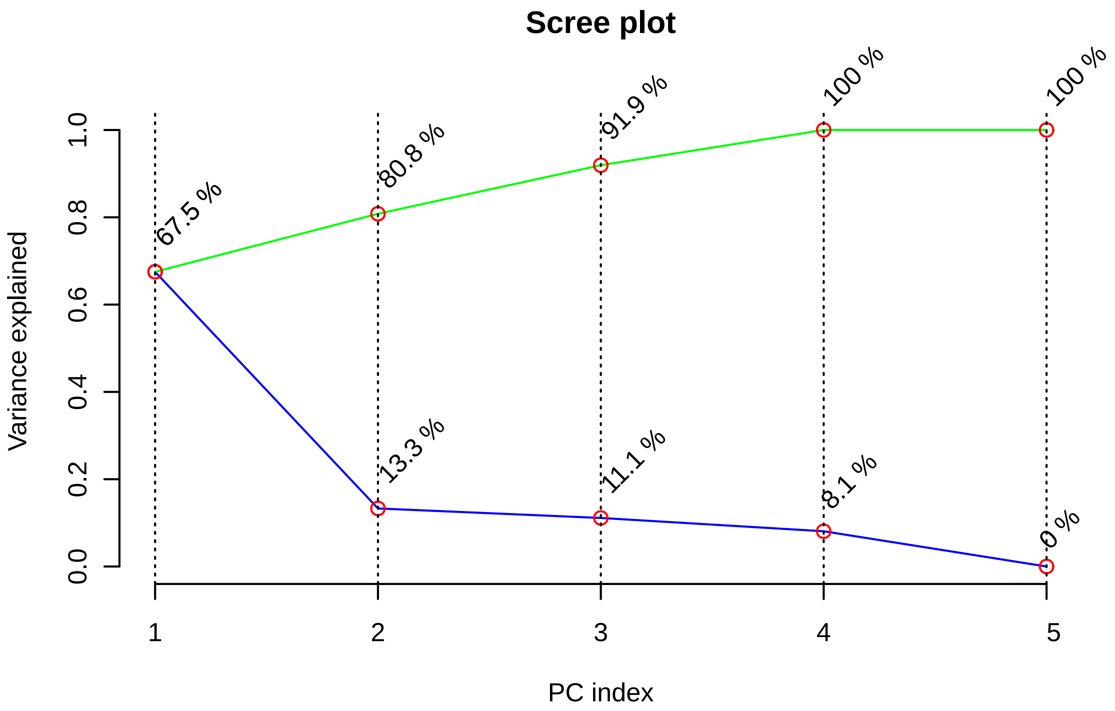

Figure 6: Scree plot shows the variance explained by PCs. The green line on top shows the accumulated variance explained; the blue line underneath shows the variance explained by individual PC.

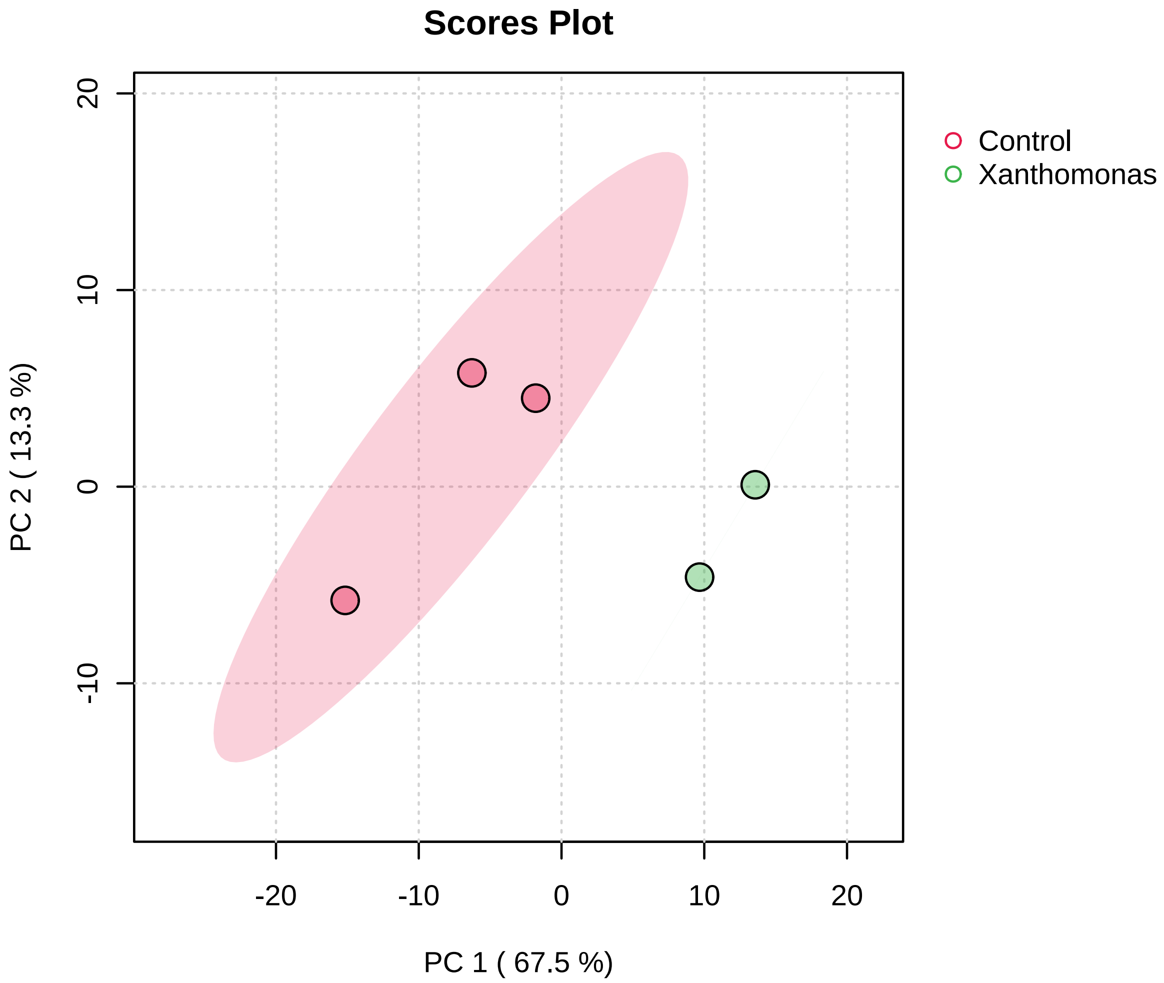

Figure 7: Scores plot between the selected PCs. The explained variances are shown in brackets.

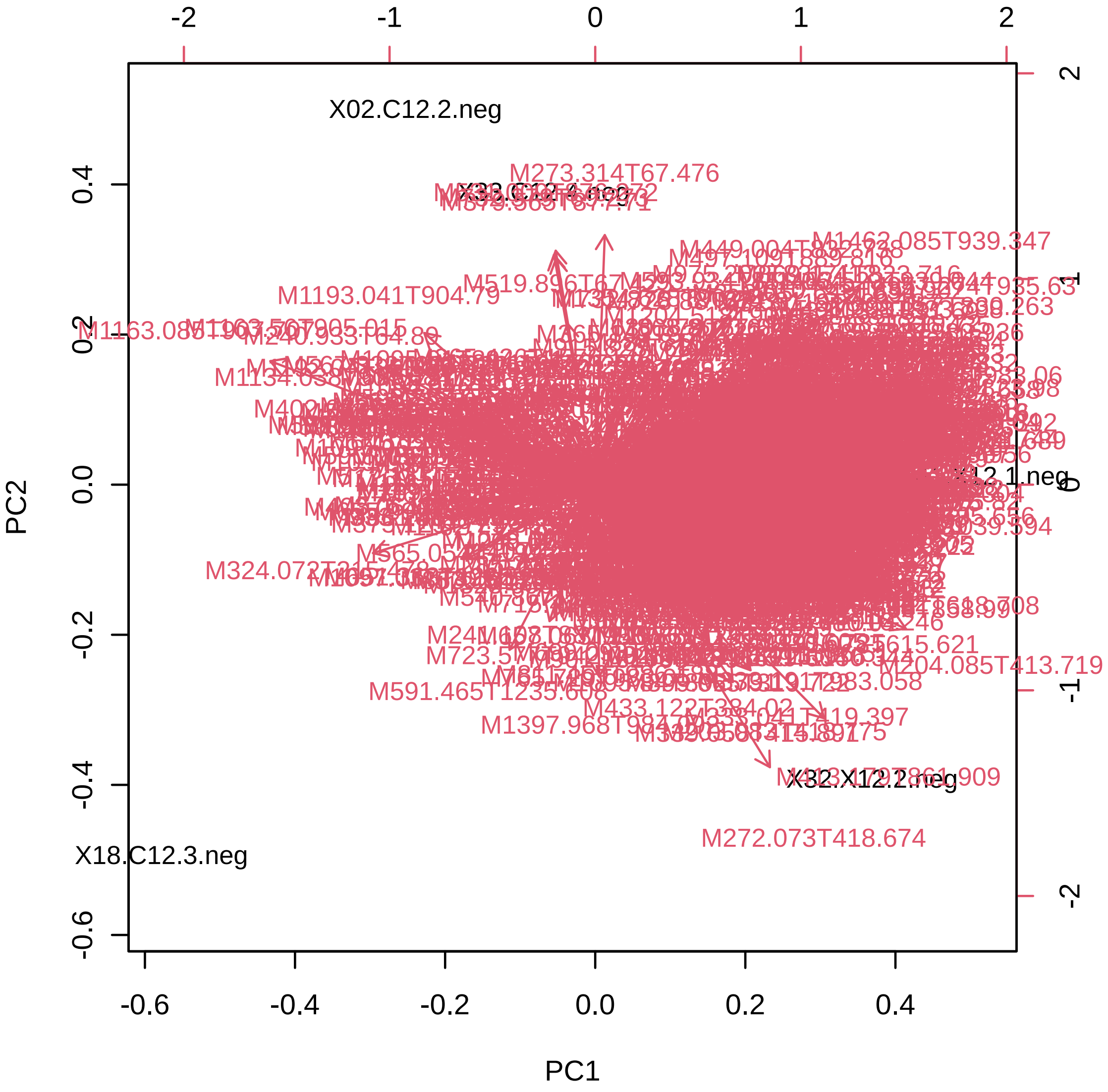

Figure 8: PCA biplot between the selected PCs. Note, you may want to test different centering and scaling normalization methods for the biplot to be displayed properly.

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

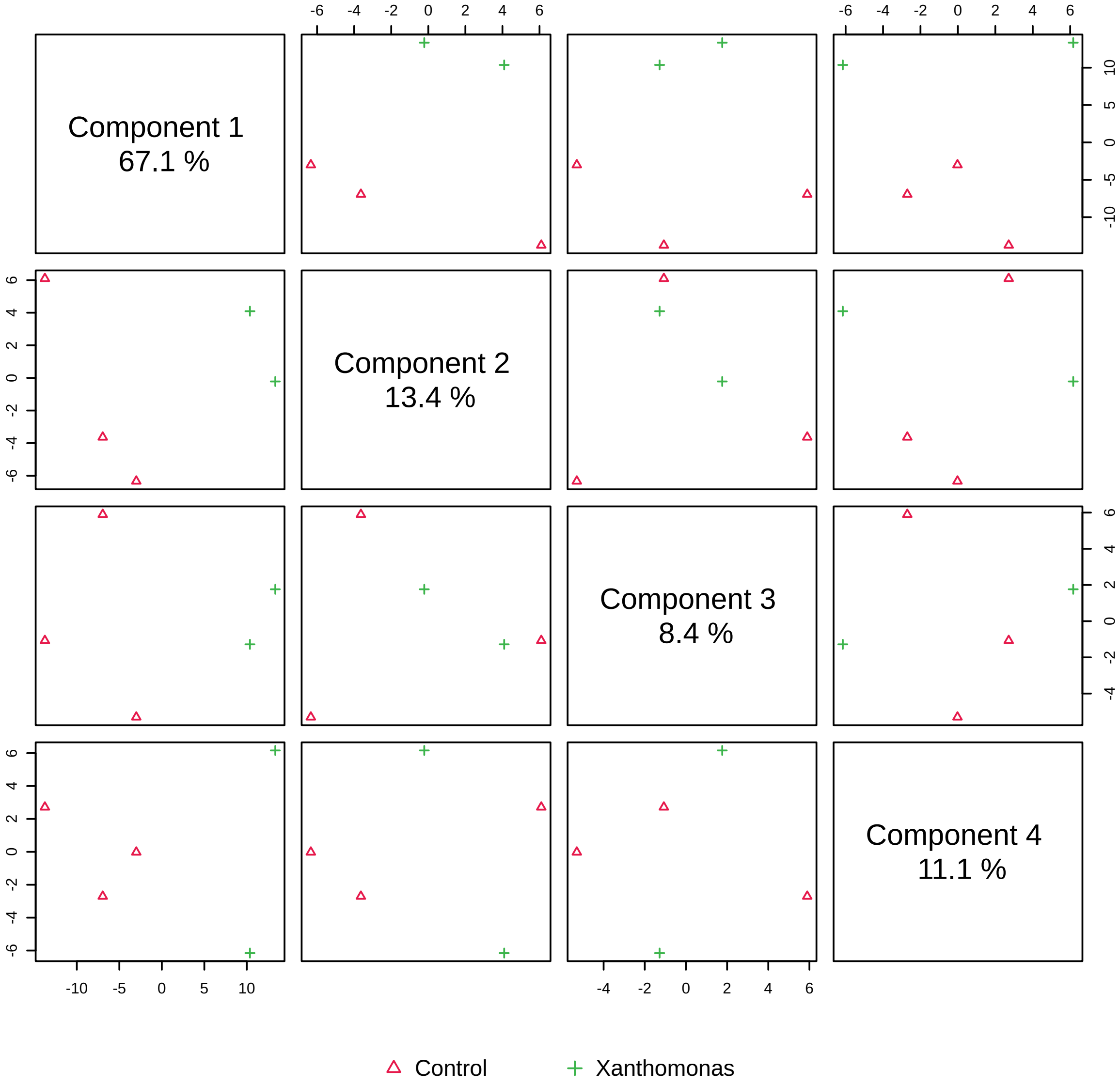

Figure 9: Pairwise scores plots between the selected components. The explained variance of each component is shown in the corresponding diagonal cell.

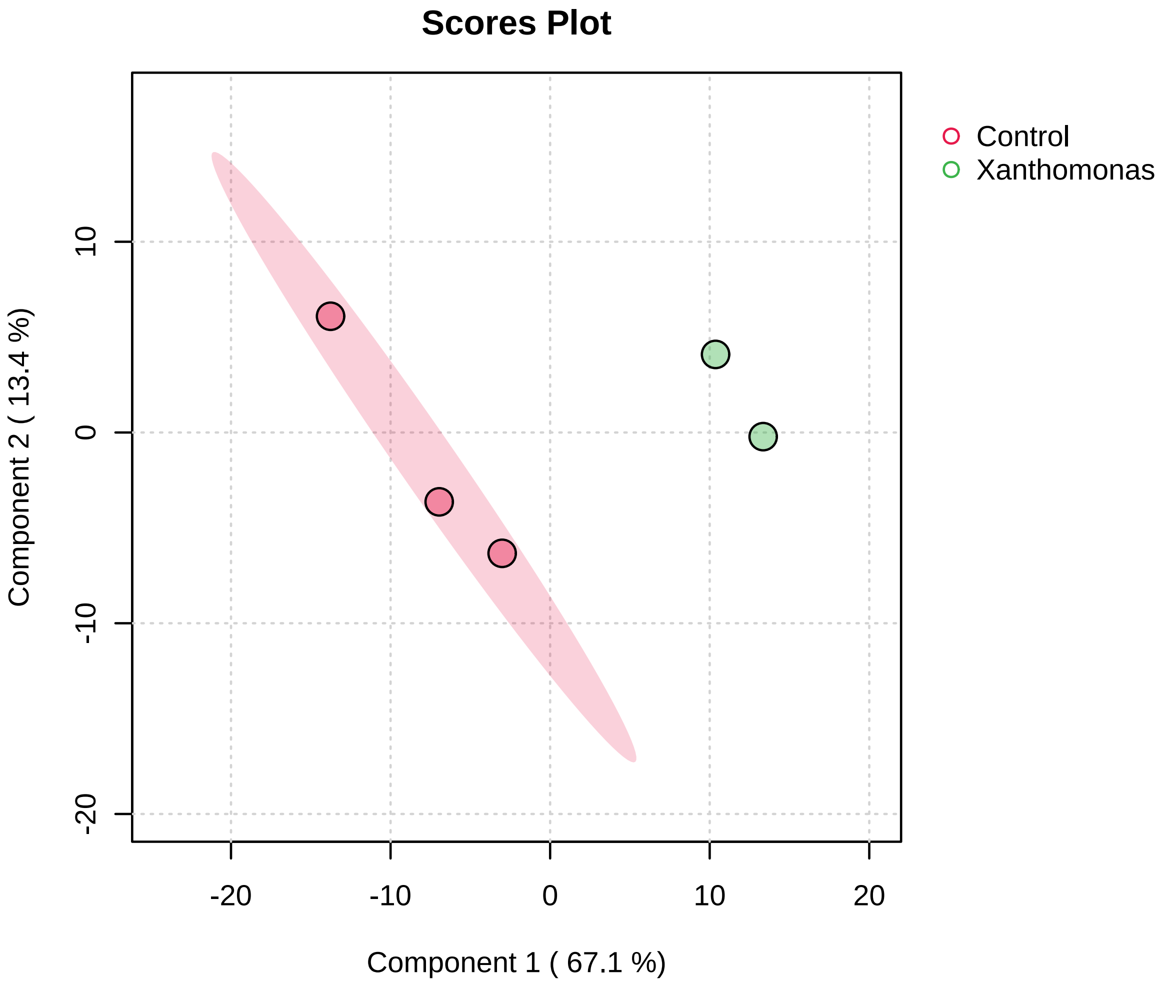

Figure 10: Scores plot between the selected PCs. The explained variances are shown in brackets.

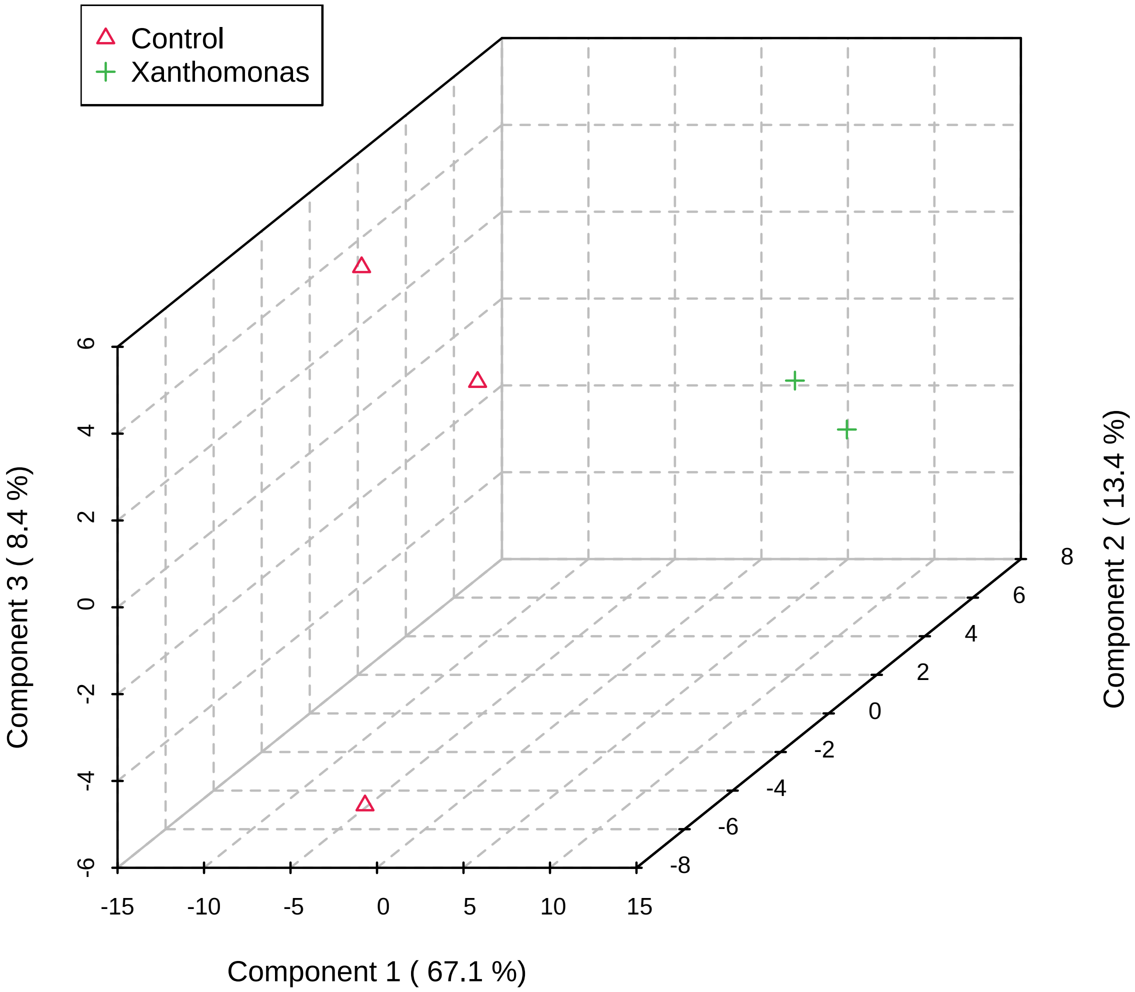

Figure 11: 3D scores plot between the selected PCs. The explained variances are shown in brackets.

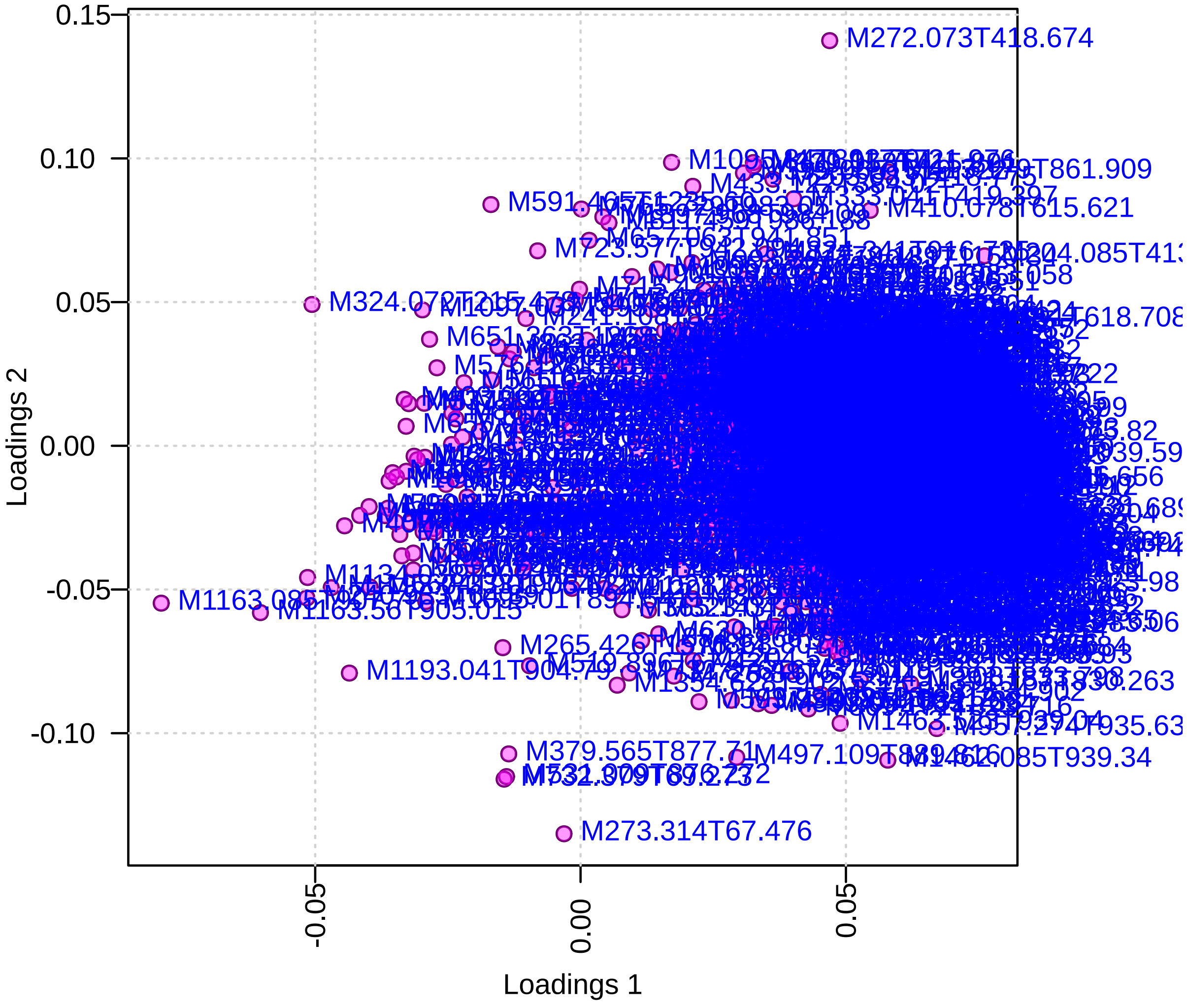

Figure 12: Loadings plot between the selected PCs.

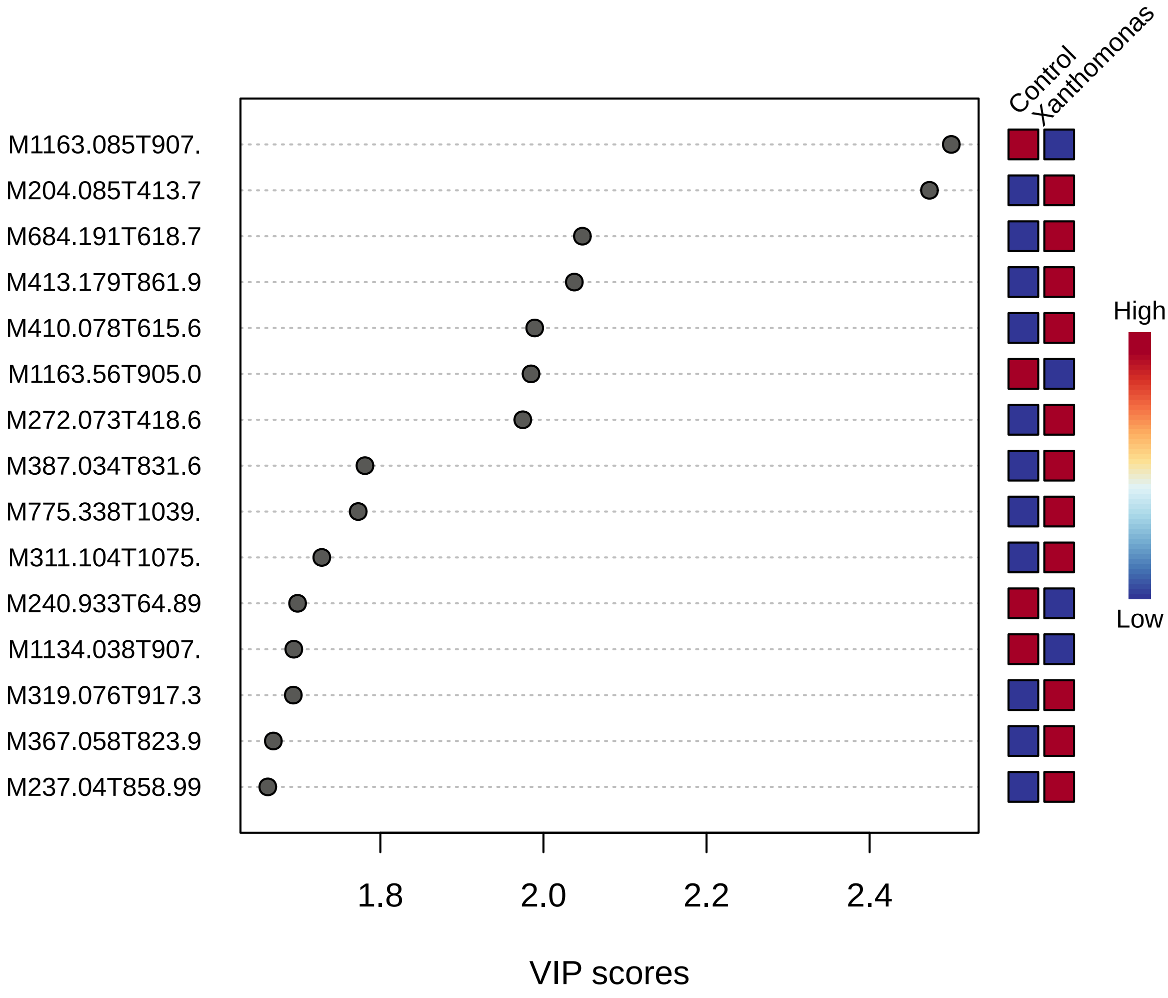

Figure 13: Important features identified by PLS-DA. The colored boxes on the right indicate the relative concentrations of the corresponding metabolite in each group under study.

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.

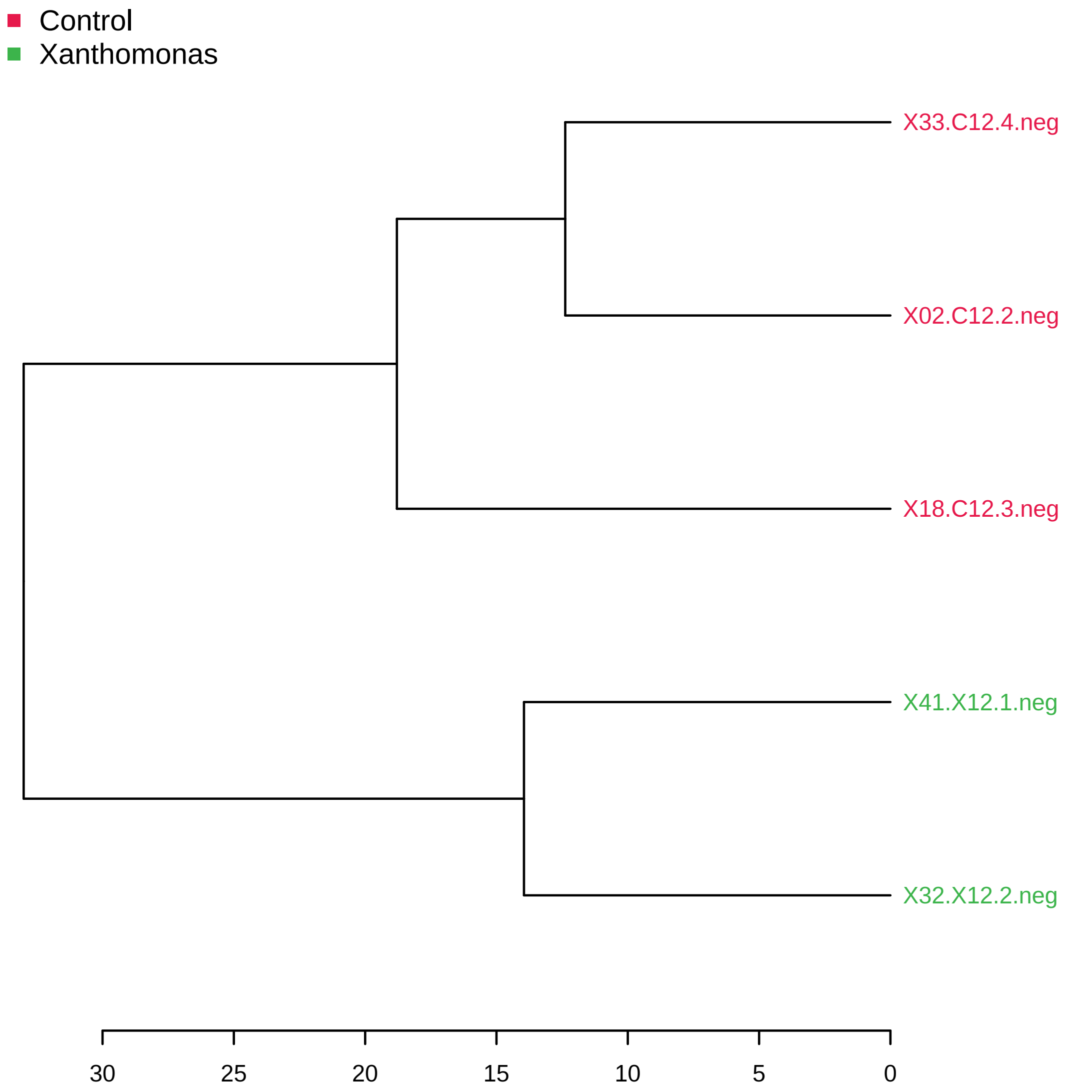

Figure 14: Clustering result shown as dendrogram (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

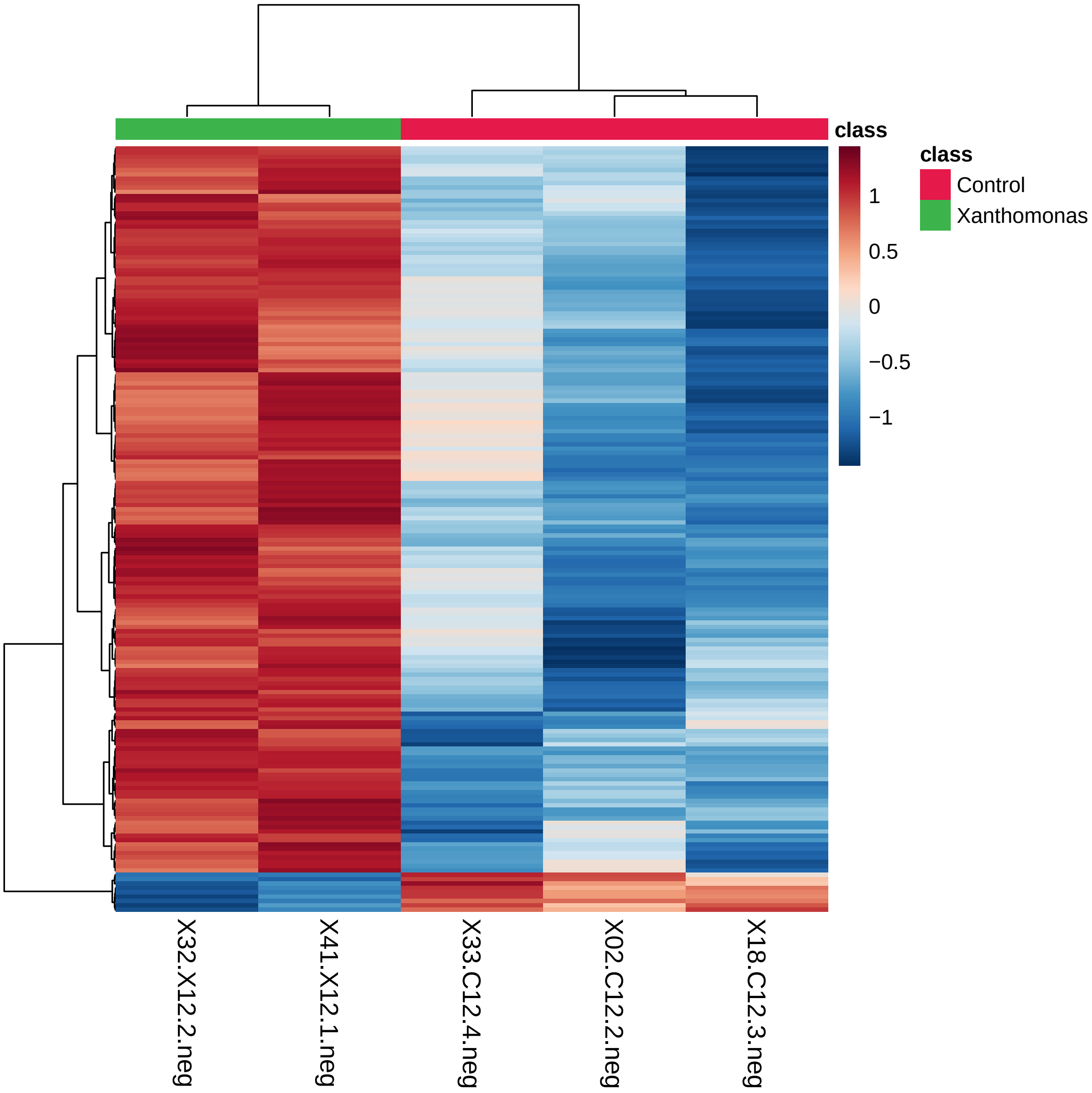

Figure 15: Clustering result shown as heatmap (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

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(\"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<-Ttests.Anal(mSet, F, 0.05, FALSE, TRUE, \"fdr\", FALSE)"
[20] "mSet<-PlotTT(mSet, \"tt_0_\", \"png\", 72, width=NA)"
[21] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, FALSE, \"raw\", FALSE)"
[22] "mSet<-PlotTT(mSet, \"tt_1_\", \"png\", 72, width=NA)"
[23] "mSet<-Ttests.Anal(mSet, T, 0.05, FALSE, FALSE, \"raw\", FALSE)"
[24] "mSet<-PlotTT(mSet, \"tt_2_\", \"png\", 72, width=NA)"
[25] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, FALSE, \"raw\", FALSE)"
[26] "mSet<-PlotTT(mSet, \"tt_3_\", \"png\", 72, width=NA)"
[27] "mSet<-Volcano.Anal(mSet, FALSE, 2.0, 0, F, 0.1, TRUE, \"raw\")"
[28] "mSet<-PlotVolcano(mSet, \"volcano_0_\",1, 0, \"png\", 72, width=NA, -1)"
[29] "mSet<-Volcano.Anal(mSet, FALSE, 2.0, 0, F, 0.1, FALSE, \"raw\")"
[30] "mSet<-PlotVolcano(mSet, \"volcano_1_\",1, 0, \"png\", 72, width=NA, -1)"
[31] "mSet<-PCA.Anal(mSet)"
[32] "mSet<-PlotPCAPairSummary(mSet, \"pca_pair_0_\", \"png\", 72, width=NA, 5)"
[33] "mSet<-PlotPCAScree(mSet, \"pca_scree_0_\", \"png\", 72, width=NA, 5)"
[34] "mSet<-PlotPCA2DScore(mSet, \"pca_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[35] "mSet<-PlotPCALoading(mSet, \"pca_loading_0_\", \"png\", 72, width=NA, 1,2);"
[36] "mSet<-PlotPCABiplot(mSet, \"pca_biplot_0_\", \"png\", 72, width=NA, 1,2)"
[37] "mSet<-PlotPCA3DLoading(mSet, \"pca_loading3d_0_\", \"json\", 1,2,3)"
[38] "mSet<-PLSR.Anal(mSet, reg=TRUE)"
[39] "mSet<-PlotPLSPairSummary(mSet, \"pls_pair_0_\", \"png\", 72, width=NA, 4)"
[40] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[41] "mSet<-PlotPLS3DScoreImg(mSet, \"pls_score3d_0_\", \"png\", 72, width=NA, 1,2,3, 40)"
[42] "mSet<-PlotPLSLoading(mSet, \"pls_loading_0_\", \"png\", 72, width=NA, 1, 2);"
[43] "mSet<-PlotPLS3DLoading(mSet, \"pls_loading3d_0_\", \"json\", 1,2,3)"
[44] "mSet<-PlotPLS.Imp(mSet, \"pls_imp_0_\", \"png\", 72, width=NA, \"vip\", \"Comp. 1\", 15,FALSE)
[45] "mSet<-PlotHCTree(mSet, \"tree_0_\", \"png\", 72, width=NA, \"euclidean\", \"ward.D\")"
[46] "mSet<-PlotHeatMap(mSet, \"heatmap_1_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\
[47] "mSet<-GetGroupNames(mSet, \"null\")"
[48] "mSet<-PlotStaticHeatMap(mSet, \"heatmap_1_\", \"pdf\", 72, width=NA, \"norm\", \"row\", \"eucl
[49] "mSet<-PlotSubHeatMap(mSet, \"heatmap_2_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclide
[50] "mSet<-PlotSubHeatMap(mSet, \"heatmap_3_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclide
[51] "mSet<-PlotSubHeatMap(mSet, \"heatmap_3_\", \"pdf\", 72, width=NA, \"norm\", \"row\", \"euclide
[52] "mSet<-SaveTransformedData(mSet)"
[53] "mSet<-PreparePDFReport(mSet, \"guest11701350932969246681\")\n"
```

The report was generated on Fri Mar 15 06:47:00 2024 with R version 4.3.2 (2023-10-31), OS system: Linux, version: -Ubuntu SMP Tue Jan 9 15:25:40 UTC 2024 .