Metabolomic Data Analysis with MetaboAnalyst 6.0

Name: guest290602822706745484

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

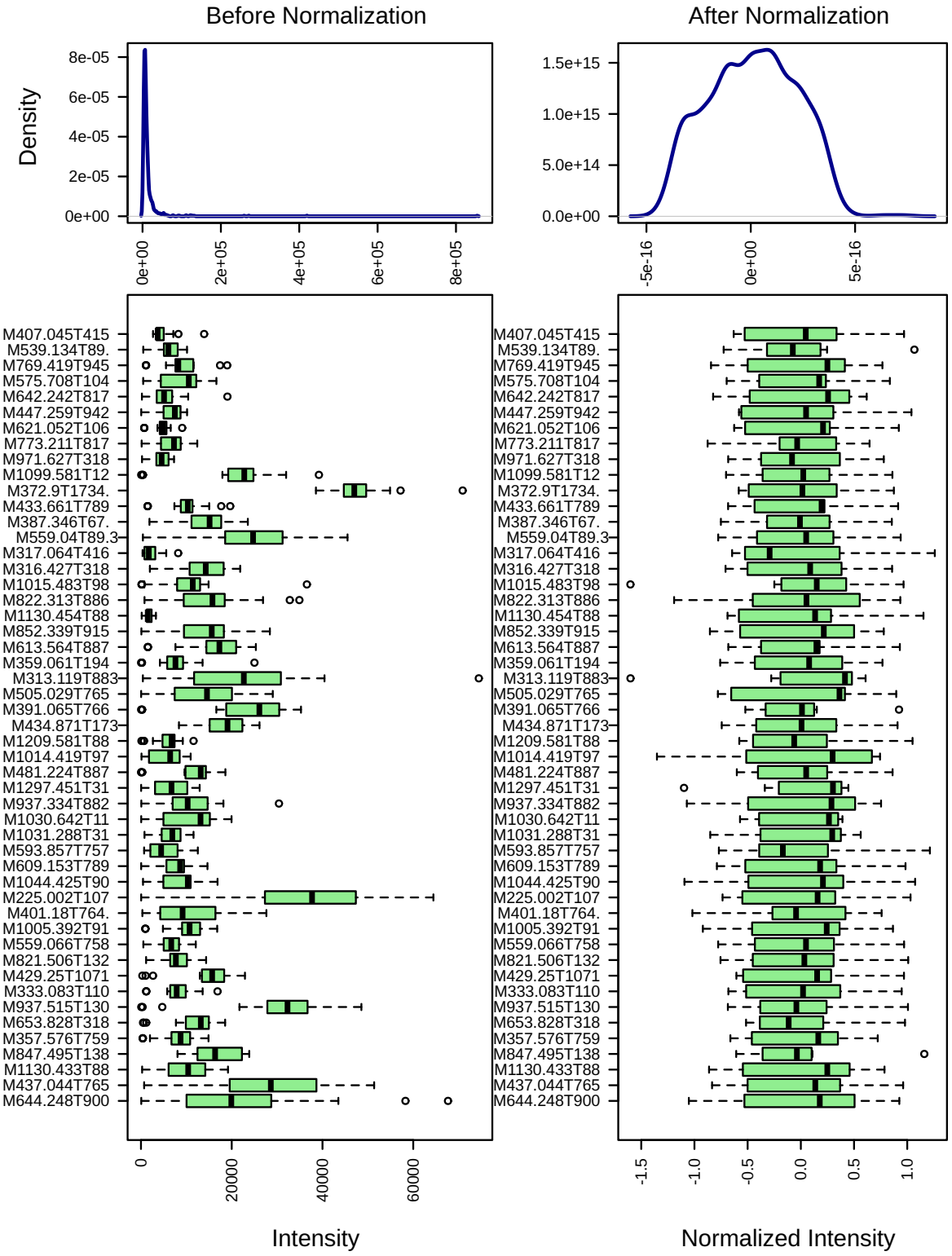

Figure 1: Box plots and kernel density plots before and after normalization. The boxplots show at most 50 features due to space limit. The density plots are based on all samples.

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.

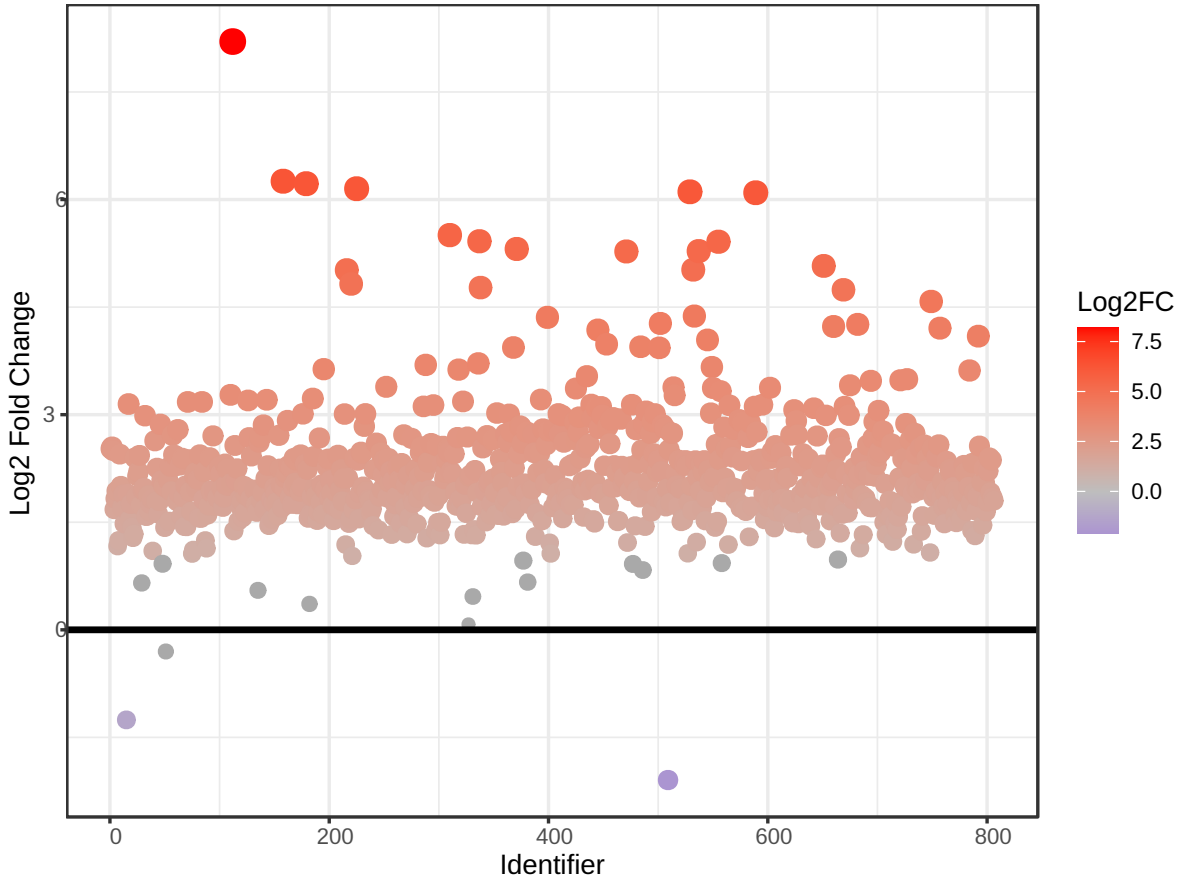

Figure 2: Important features selected by fold-change analysis with threshold 2. The red circles represent features above the threshold. Note the values are on log scale, so that both up-regulated and down-regulated features can be plotted in a symmetrical way

Table 2: Top 50 features identified by fold change analysis

	Peaks(mz/rt)	Fold Change	log2(FC)
1	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	M410.078T615.621	45.38	5.504
8	M387.034T831.689	42.759	5.4181
9	M365.528T823.98	42.562	5.4115
10	M365.052T459.24	39.688	5.3106
11	M531.009T876.272	38.888	5.2812
12	M449.004T832.738	38.744	5.2759
13	M463.02T875.074	33.695	5.0744
14	M379.565T877.71	32.476	5.0213
15	M1611.697T937.933	32.362	5.0162
16	M380.071T778.258	28.306	4.8231
17	M455.021T825.04	27.324	4.7721
18	M413.179T861.909	26.745	4.7412
19	M319.076T917.392	23.894	4.5786
20	M433.038T831.69	20.749	4.375
21	M379.067T781.128	20.495	4.3572
22	M1613.665T937.741	19.324	4.2723
23	M365.143T832.741	19.148	4.2591
24	M975.202T831.412	18.76	4.2296
25	M401.05T875.074	18.459	4.2063
26	M684.191T618.708	18.139	4.181
27	M447.05T875.673	17.087	4.0949
28	M471.117T721.976	16.474	4.0422
29	M895.29T938.858	15.808	3.9826
30	M402.053T876.272	15.425	3.9472
31	M857.325T1034.658	15.314	3.9368
32	M964.359T921.966	15.262	3.9319
33	M341.535T617.585	13.133	3.7151
34	M430.136T769.014	12.944	3.6942
35	M661.142T919.275	12.672	3.6636
36	M867.206T809.129	12.42	3.6346
37	M967.36T964.868	12.367	3.6284
38	M775.338T1039.594	12.261	3.616
39	M1462.085T939.347	11.603	3.5364
40	M759.145T875.232	11.273	3.4948
41	M333.062T954.304	11.155	3.4797
42	M1396.865T985.147	11.093	3.4716
43	M341.465T625.973	10.627	3.4097
44	M1014.419T974.744	10.472	3.3885
45	M781.183T974.609	10.421	3.3814
46	M957.274T935.63	10.378	3.3754
47	M409.075T617.585	10.357	3.3725
48	M483.129T913.248	10.3	3.3646
49	M942.309T937.647	10.042	3.328
50	M821.606T900.199	9.6883	3.2762

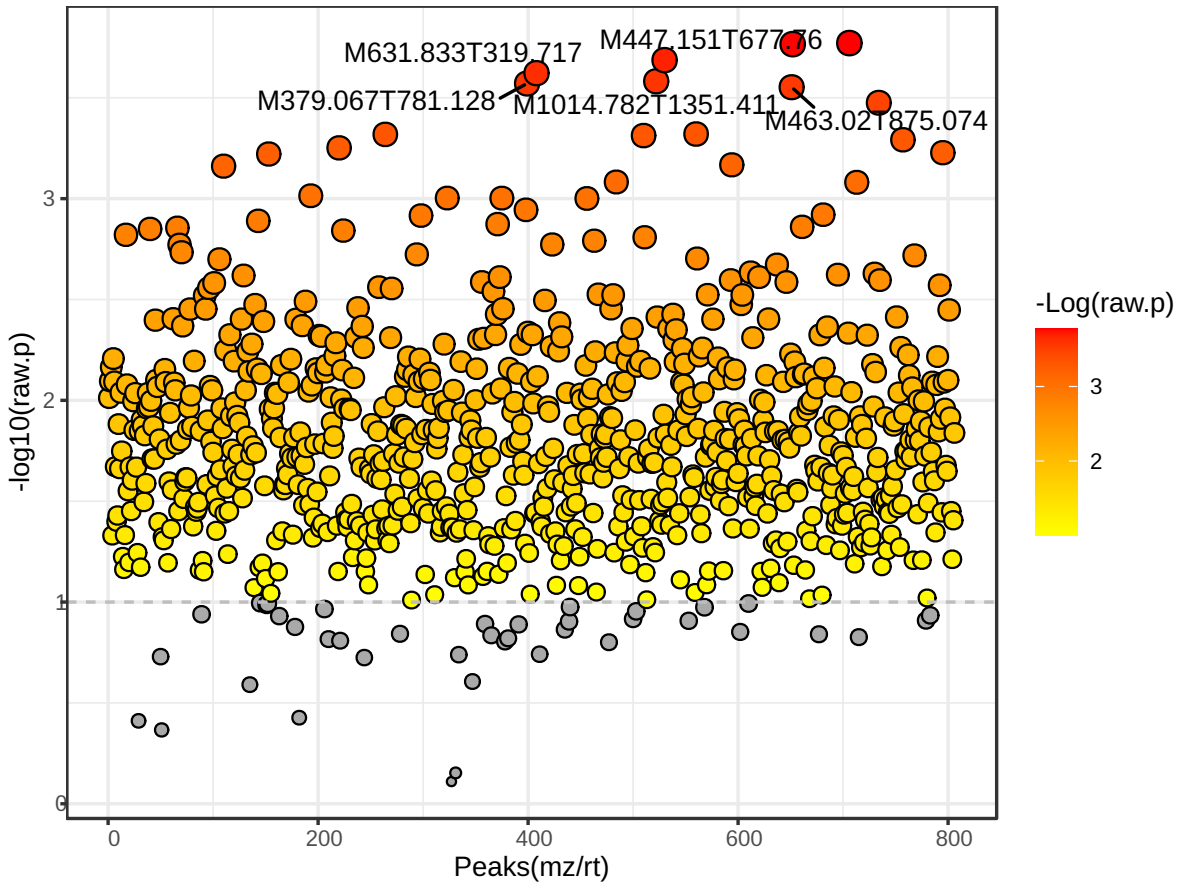

Figure 3: Important features selected by t-tests with threshold 0.1. The red circles represent features above the threshold. Note the p values are transformed by -log10 so that the more significant features (with smaller p values) will be plotted higher on the graph.

Table 3: Top 50 features identified by t-tests

	Peaks(mz/rt)	t.stat	p.value	-log10(p)	FDR
1	M1014.811T1344.847	-10.102	0.00016936	3.7712	0.032267
2	M289.12T769.482	-13.392	0.00017141	3.766	0.032267
3	M447.151T677.76	-9.685	0.00020574	3.6867	0.032267
4	M631.833T319.717	-9.3269	0.0002387	3.6222	0.032267
5	M1014.782T1351.411	-11.551	0.00026187	3.5819	0.032267
6	M379.067T781.128	-9.5814	0.00026864	3.5708	0.032267
7	M463.02T875.074	-9.031	0.00028023	3.5525	0.032267
8	M238.069T801.926	-9.8609	0.00033477	3.4753	0.032284
9	M365.136T768.138	-8.4084	0.0004782	3.3204	0.032284
10	M398.263T1317.762	-8.5793	0.00048032	3.3185	0.032284
11	M645.252T899.393	-8.5835	0.00048679	3.3127	0.032284
12	M401.05T875.074	-8.0252	0.0005107	3.2918	0.032284
13	M380.071T778.258	-7.7861	0.00056011	3.2517	0.032284
14	M615.195T761.983	-9.8803	0.00059179	3.2278	0.032284
15	M970.001T316.098	-9.2998	0.00060082	3.2213	0.032284
16	M621.037T758.225	-8.5845	0.00067978	3.1676	0.032719
17	M821.606T900.199	-7.6992	0.0006901	3.1611	0.032719
18	M402.053T876.272	-10.047	0.000827	3.0825	0.033537
19	M395.28T1470.273	-7.5166	0.00083001	3.0809	0.033537
20	M963.275T732.778	-6.9199	0.00096751	3.0143	0.033537
21	M240.933T64.89	-6.9419	0.0009933	3.0029	0.033537
22	M505.029T765.672	-9.0365	0.0009938	3.0027	0.033537
23	M852.339T915.443	-6.8731	0.00099796	3.0009	0.033537
24	M702.255T837.404	-6.6989	0.0011363	2.9445	0.033537
25	M373.167T930.605	-6.6351	0.0012009	2.9205	0.033537
26	M533.169T900.984	-8.6692	0.0012131	2.9161	0.033537
27	M481.183T769.795	-7.5219	0.0012886	2.8899	0.033537
28	M365.052T459.24	-7.7623	0.0013374	2.8737	0.033537
29	M341.088T439.204	-7.38	0.0013798	2.8602	0.033537
30	M729.273T847.264	-6.3939	0.0013945	2.8556	0.033537
31	M1053.272T316.465	-6.4693	0.001413	2.8499	0.033537
32	M221.066T891.624	-6.3877	0.00144	2.8416	0.033537
33	M180.067T785.272	-6.5319	0.0015111	2.8207	0.033537
34	M397.074T474.916	-6.4429	0.0015542	2.8085	0.033537
35	M544.284T936.329	-6.1885	0.0016165	2.7914	0.033537
36	M631.863T318.629	-6.4759	0.0016877	2.7727	0.033537
37	M653.752T320.843	-7.065	0.0016928	2.7714	0.033537
38	M653.783T316.318	-7.3614	0.0018439	2.7343	0.033537
39	M429.646T308.971	-6.0082	0.0018897	2.7236	0.033537
40	M549.174T847.493	-6.0554	0.0019117	2.7186	0.033537
41	M461.152T871.999	-6.1608	0.0019805	2.7032	0.033537
42	M642.242T817.768	-6.9179	0.0019977	2.6995	0.033537
43	M643.245T933.361	-6.4783	0.0021225	2.6732	0.033537
44	M617.247T870.852	-6.119	0.0023198	2.6346	0.033537
45	M824.341T916.725	-7.3707	0.0023513	2.6287	0.033537
46	M469.095T760.602	-6.4795	0.0023812	2.6232	0.033537
47	M703.247T844.715	-6.0799	0.0024043	2.619	0.033537
48	M968.386T889.852	-5.8329	0.0024484	2.6111	0.033537
49	M413.104T602.912	-6.2024	0.0024494	2.6109	0.033537
50	M473.173T881.847	-5.5955	0.0025321	2.5965	0.033537

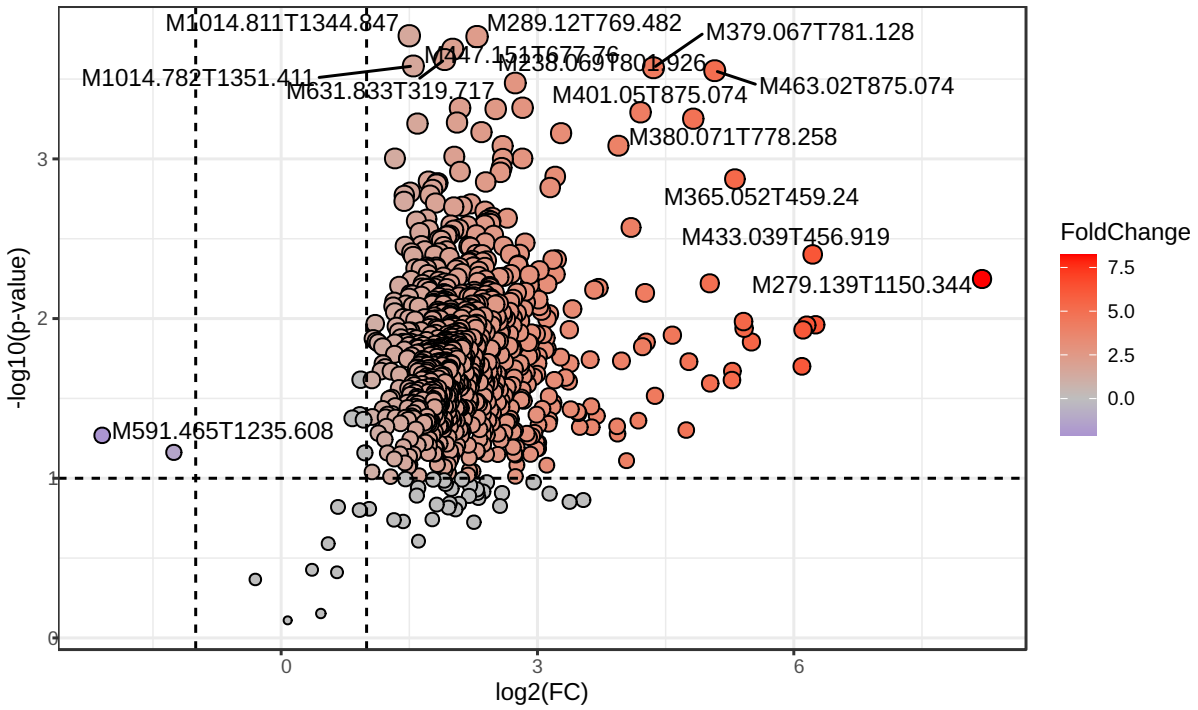

Figure 4: Important features selected by volcano plot with fold change threshold (x) 2 and t-tests threshold (y) 0.1. The red circles represent features above the threshold. Note both fold changes and p values are log transformed. The further its position away from the (0,0), the more significant the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

Table 4: Top 50 features identified by volcano plot

	Peaks(mz/rt)	FC	log2(FC)	raw.pval	-log10(p)
1	M1014.811T1344.847	2.8274	1.4995	0.00016936	3.7712
2	M289.12T769.482	4.8995	2.2926	0.00017141	3.766
3	M447.151T677.76	4.0222	2.008	0.00020574	3.6867
4	M631.833T319.717	3.7684	1.9139	0.0002387	3.6222
5	M1014.782T1351.411	2.9222	1.5471	0.00026187	3.5819
6	M379.067T781.128	20.495	4.3572	0.00026864	3.5708
7	M463.02T875.074	33.695	5.0744	0.00028023	3.5525
8	M238.069T801.926	6.6807	2.74	0.00033477	3.4753
9	M365.136T768.138	7.0893	2.8256	0.0004782	3.3204
10	M398.263T1317.762	4.2681	2.0936	0.00048032	3.3185
11	M645.252T899.393	5.7006	2.5111	0.00048679	3.3127
12	M401.05T875.074	18.459	4.2063	0.0005107	3.2918
13	M380.071T778.258	28.306	4.8231	0.00056011	3.2517
14	M615.195T761.983	4.1651	2.0583	0.00059179	3.2278
15	M970.001T316.098	3.0216	1.5953	0.00060082	3.2213
16	M621.037T758.225	5.0775	2.3441	0.00067978	3.1676
17	M821.606T900.199	9.6883	3.2762	0.0006901	3.1611
18	M402.053T876.272	15.425	3.9472	0.000827	3.0825
19	M395.28T1470.273	6.0291	2.5919	0.00083001	3.0809
20	M963.275T732.778	4.0739	2.0264	0.00096751	3.0143
21	M240.933T64.89	2.5165	1.3314	0.0009933	3.0029
22	M505.029T765.672	7.0782	2.8234	0.0009938	3.0027
23	M852.339T915.443	6.0406	2.5947	0.00099796	3.0009
24	M702.255T837.404	5.9809	2.5804	0.0011363	2.9445
25	M373.167T930.605	4.2653	2.0927	0.0012009	2.9205
26	M533.169T900.984	5.9218	2.566	0.0012131	2.9161
27	M481.183T769.795	9.2388	3.2077	0.0012886	2.8899
28	M365.052T459.24	39.688	5.3106	0.0013374	2.8737
29	M341.088T439.204	3.3051	1.7247	0.0013798	2.8602
30	M729.273T847.264	5.2557	2.3939	0.0013945	2.8556
31	M1053.272T316.465	3.56	1.8319	0.001413	2.8499
32	M221.066T891.624	3.5281	1.8189	0.00144	2.8416
33	M180.067T785.272	8.8676	3.1485	0.0015111	2.8207
34	M397.074T474.916	3.4159	1.7723	0.0015542	2.8085
35	M544.284T936.329	2.8479	1.5099	0.0016165	2.7914
36	M631.863T318.629	3.3459	1.7424	0.0016877	2.7727
37	M653.752T320.843	2.7156	1.4413	0.0016928	2.7714
38	M653.783T316.318	2.7111	1.4389	0.0018439	2.7343
39	M429.646T308.971	3.4994	1.8071	0.0018897	2.7236
40	M549.174T847.493	4.6592	2.2201	0.0019117	2.7186
41	M461.152T871.999	4.345	2.1193	0.0019805	2.7032
42	M642.242T817.768	4.0552	2.0198	0.0019977	2.6995
43	M643.245T933.361	5.2261	2.3857	0.0021225	2.6732
44	M617.247T870.852	5.5369	2.4691	0.0023198	2.6346
45	M824.341T916.725	6.264	2.6471	0.0023513	2.6287
46	M469.095T760.602	3.26	1.7049	0.0023812	2.6232
47	M703.247T844.715	5.3585	2.4218	0.0024043	2.619
48	M968.386T889.852	5.4818	2.4547	0.0024484	2.6111
49	M413.104T602.912	2.9951	1.5826	0.0024494	2.6109
50	M473.173T881.847	5.2183	2.3836	0.0025321	2.5965

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.

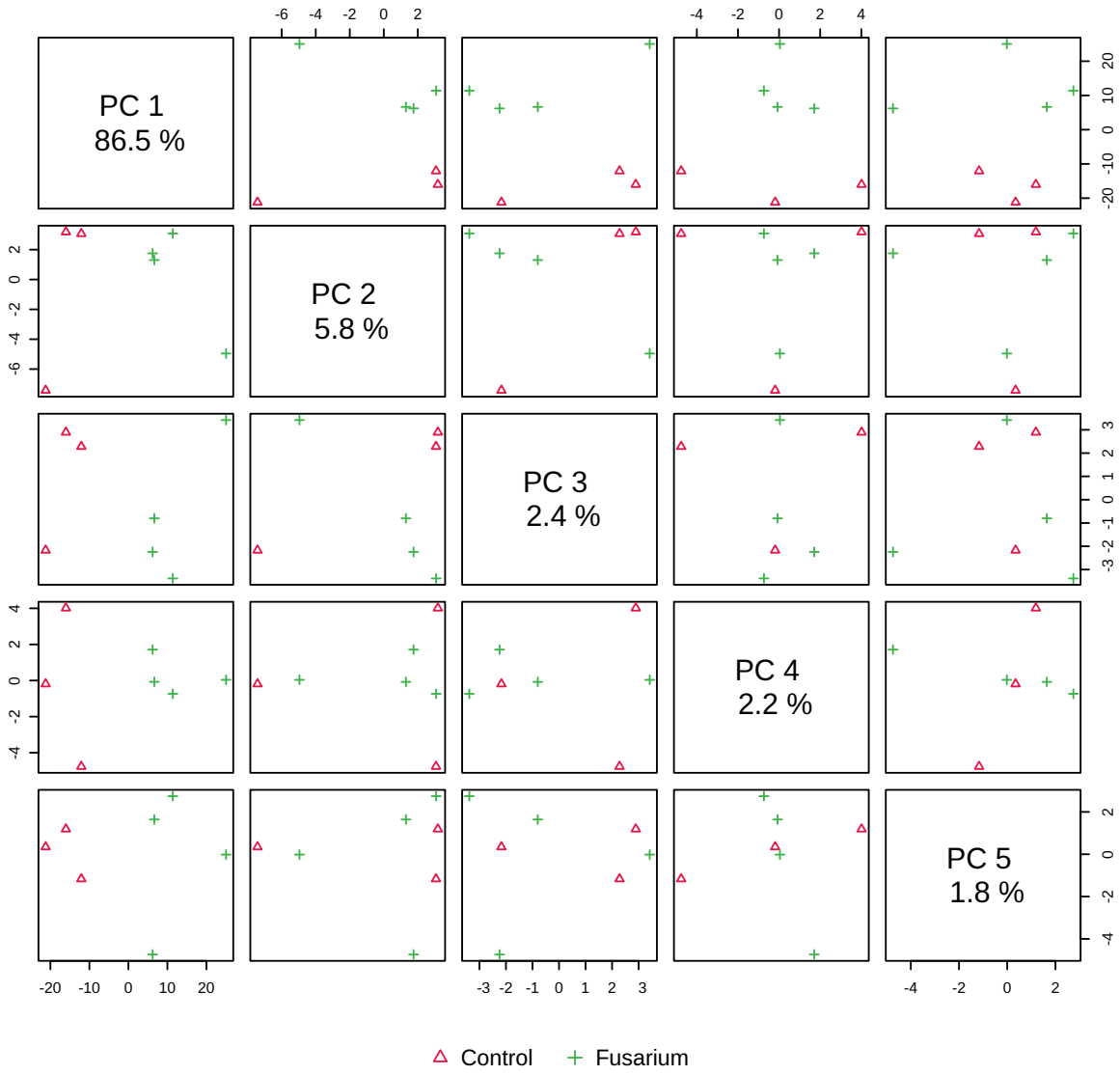

Figure 5: Pairwise score plots between the selected PCs. The explained variance of each PC is shown in the corresponding diagonal cell.

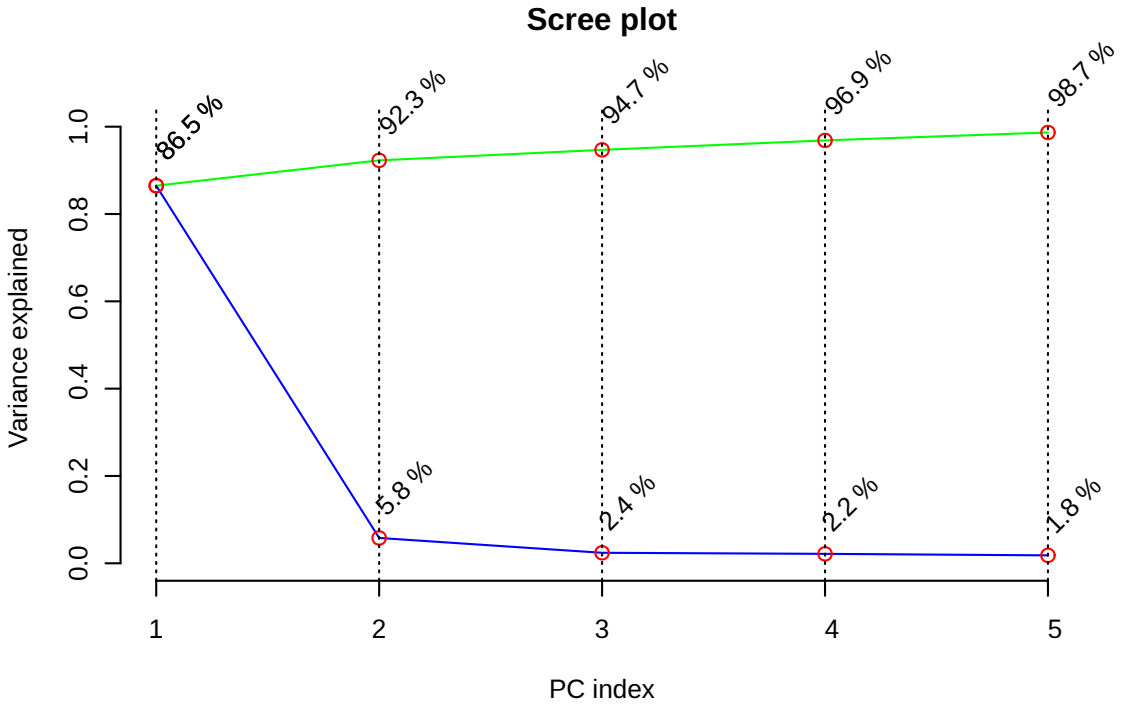

Figure 6: Scree plot shows the variance explained by PCs. The green line on top shows the accumulated variance explained; the blue line underneath shows the variance explained by individual PC.

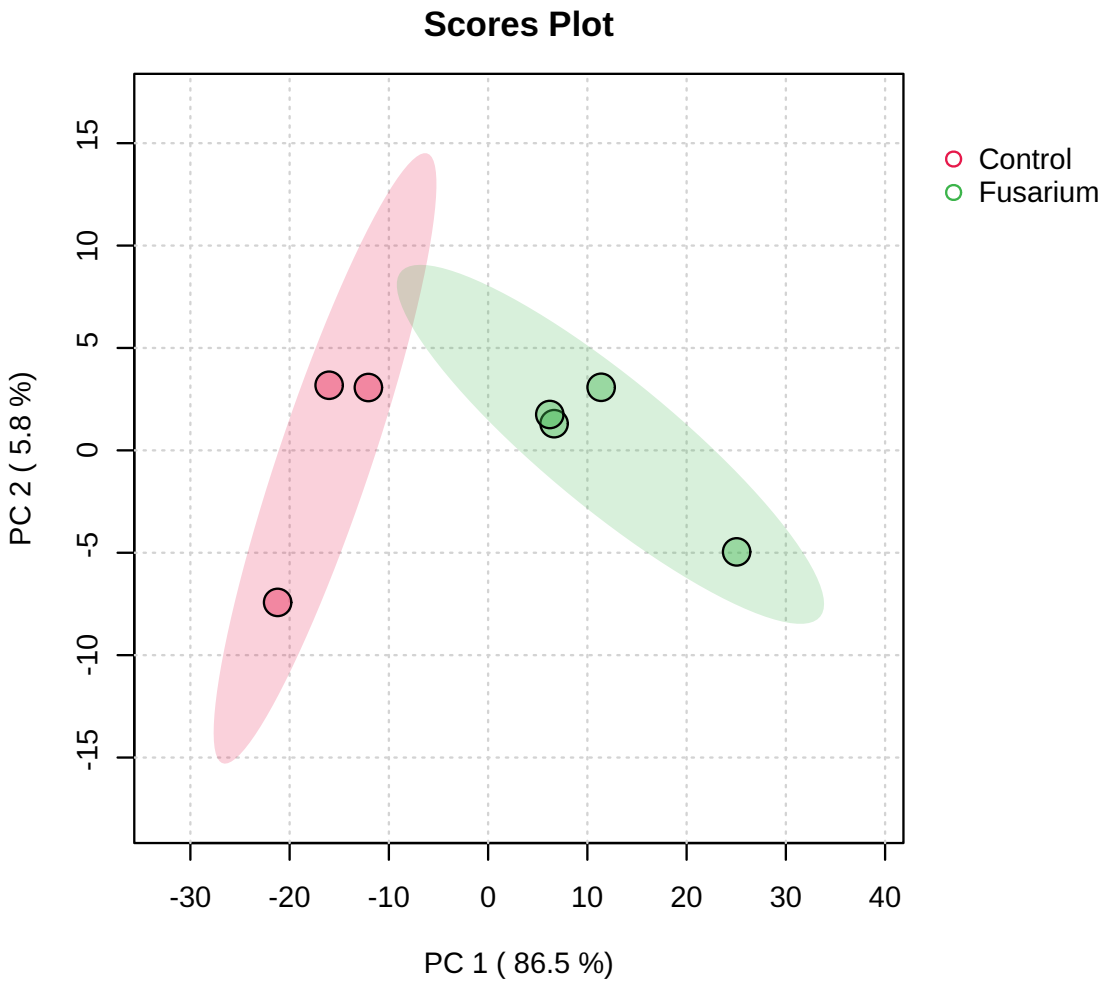

Figure 7: Scores plot between the selected PCs. The explained variances are shown in brackets.

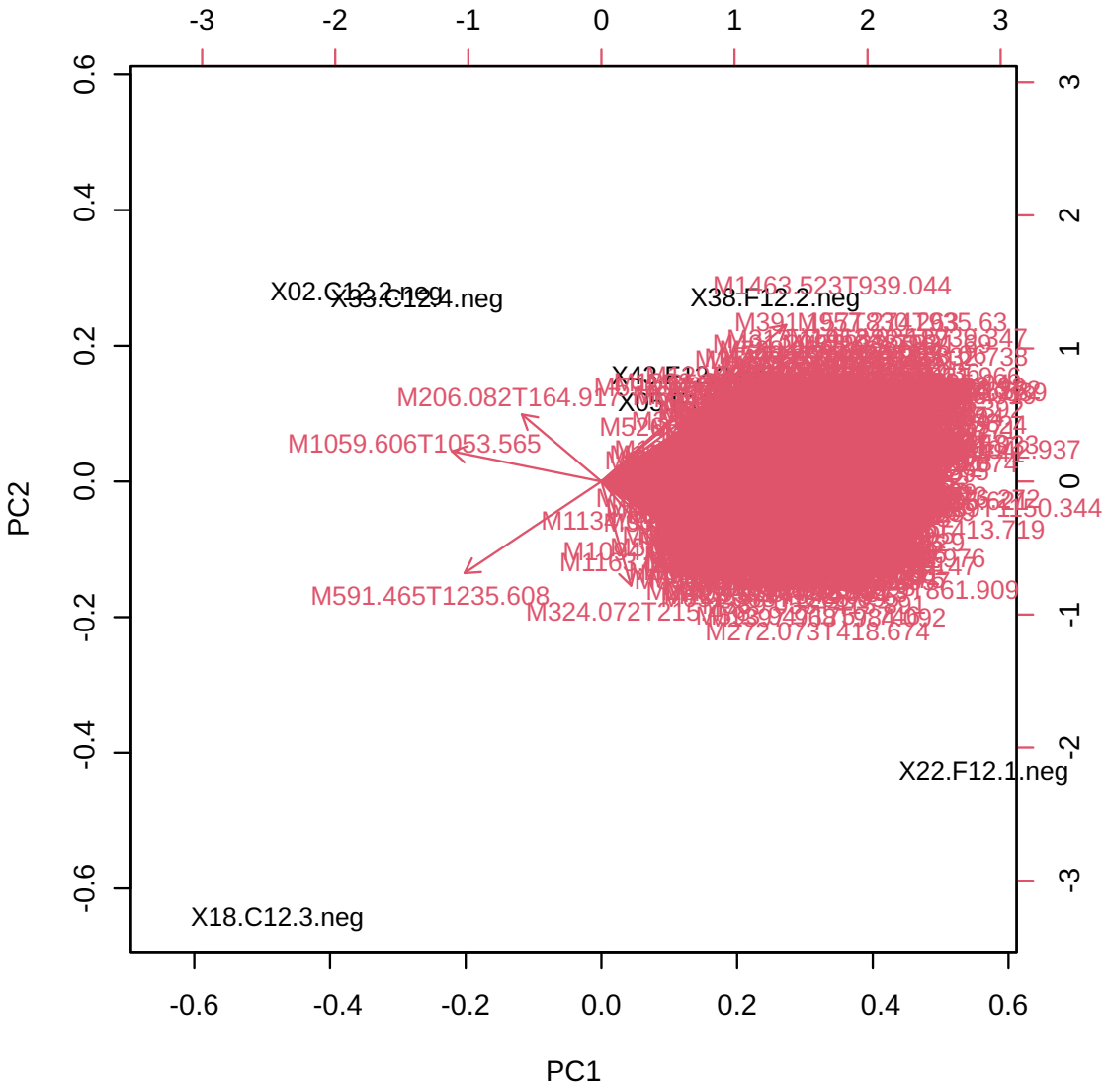

Figure 8: PCA biplot between the selected PCs. Note, you may want to test different centering and scaling normalization methods for the biplot to be displayed properly.

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

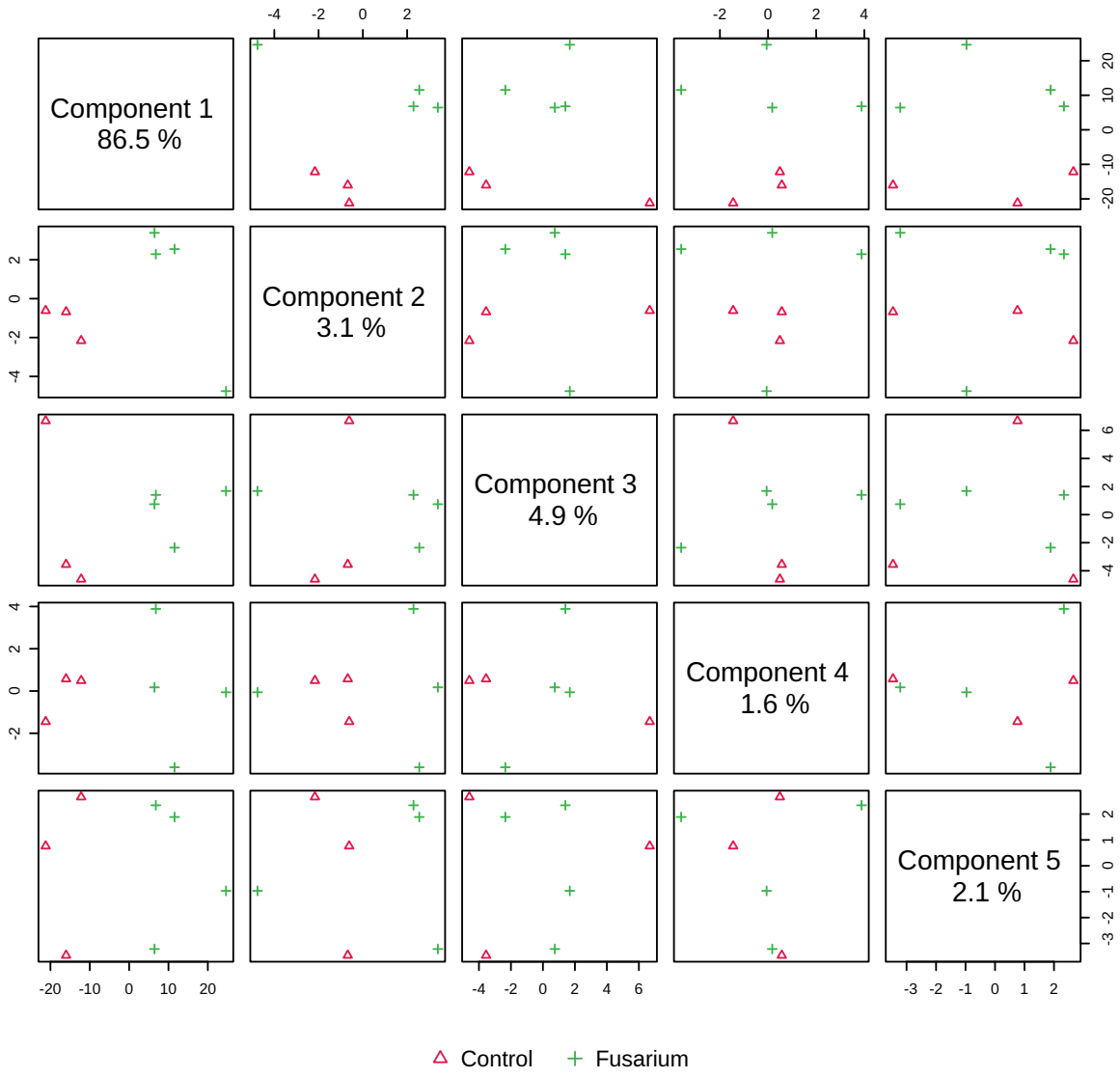

Figure 9: Pairwise scores plots between the selected components. The explained variance of each component is shown in the corresponding diagonal cell.

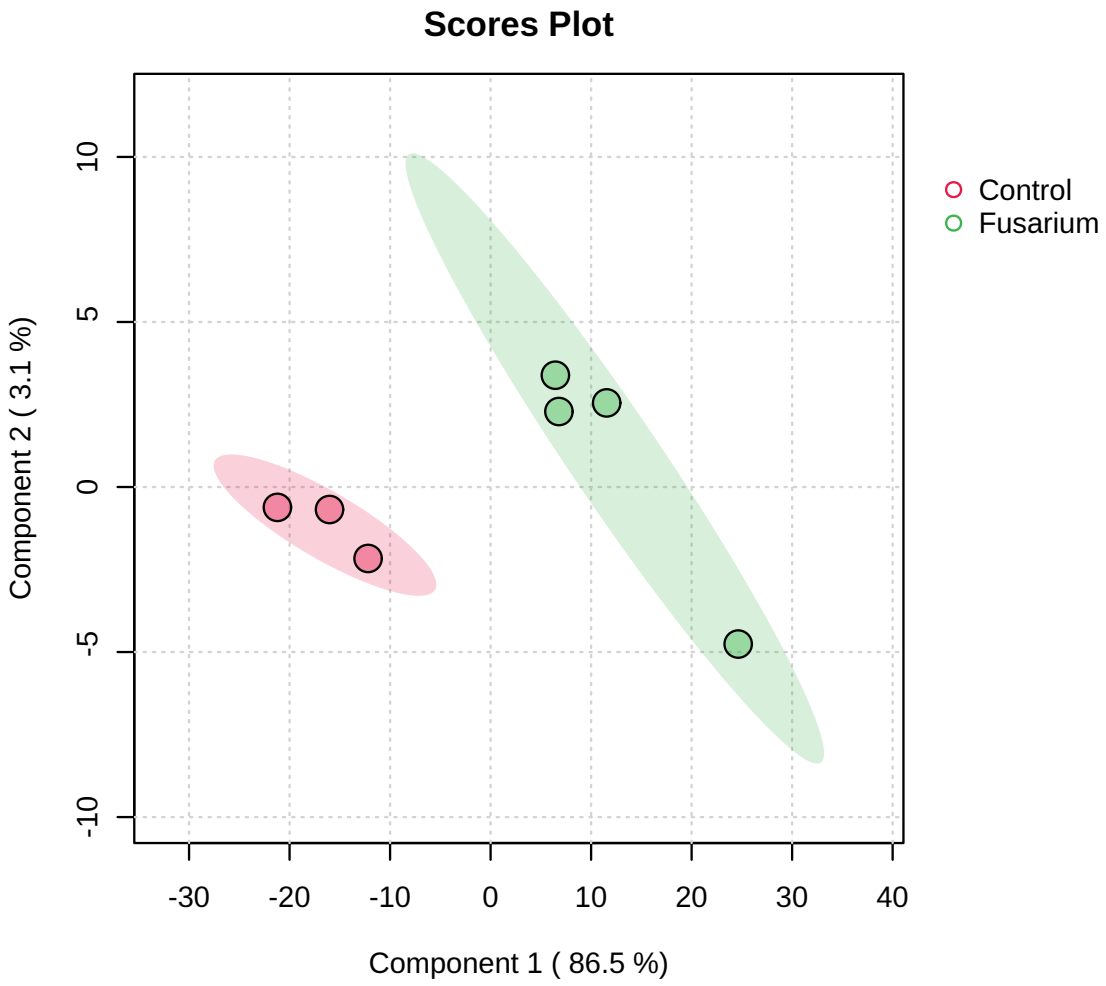

Figure 10: Scores plot between the selected PCs. The explained variances are shown in brackets.

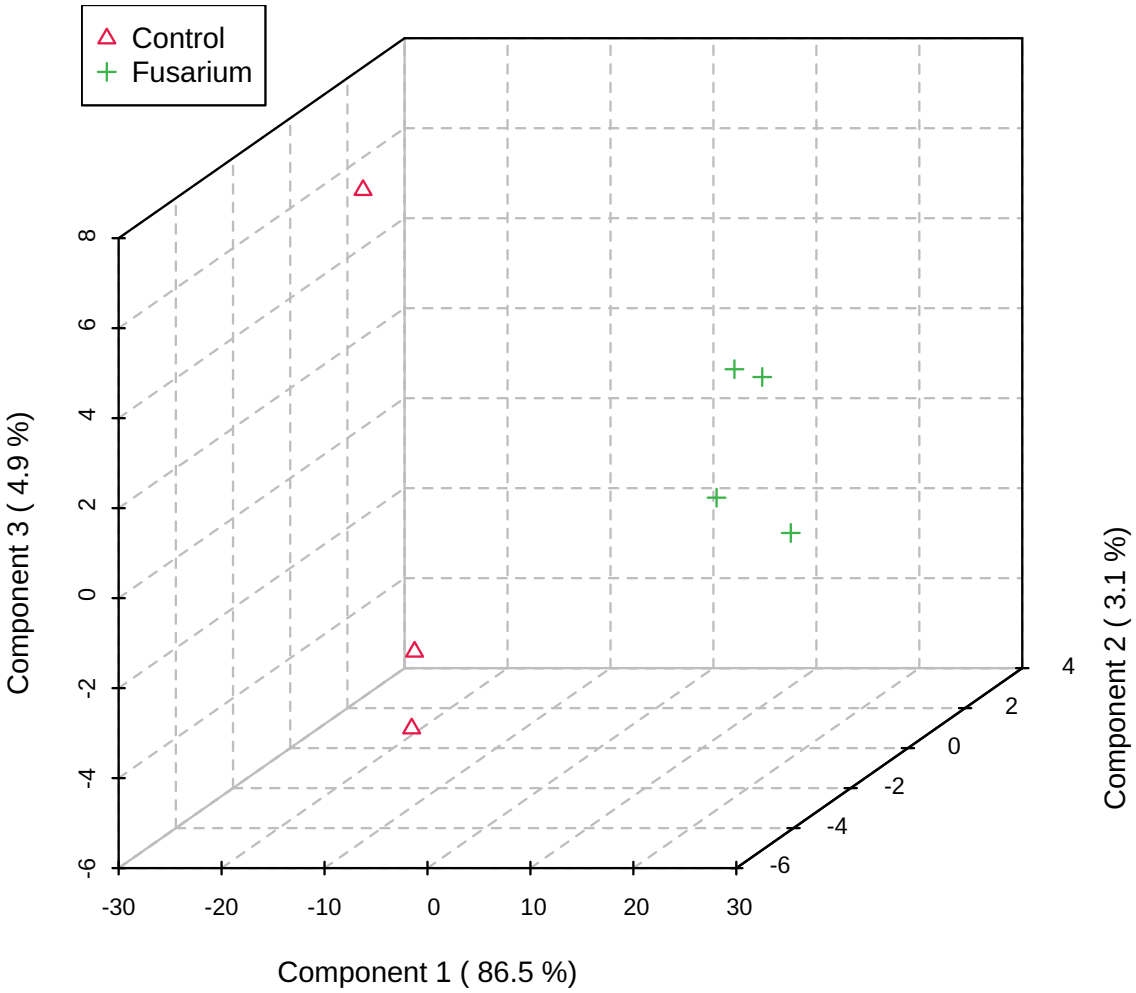

Figure 11: 3D scores plot between the selected PCs. The explained variances are shown in brackets.

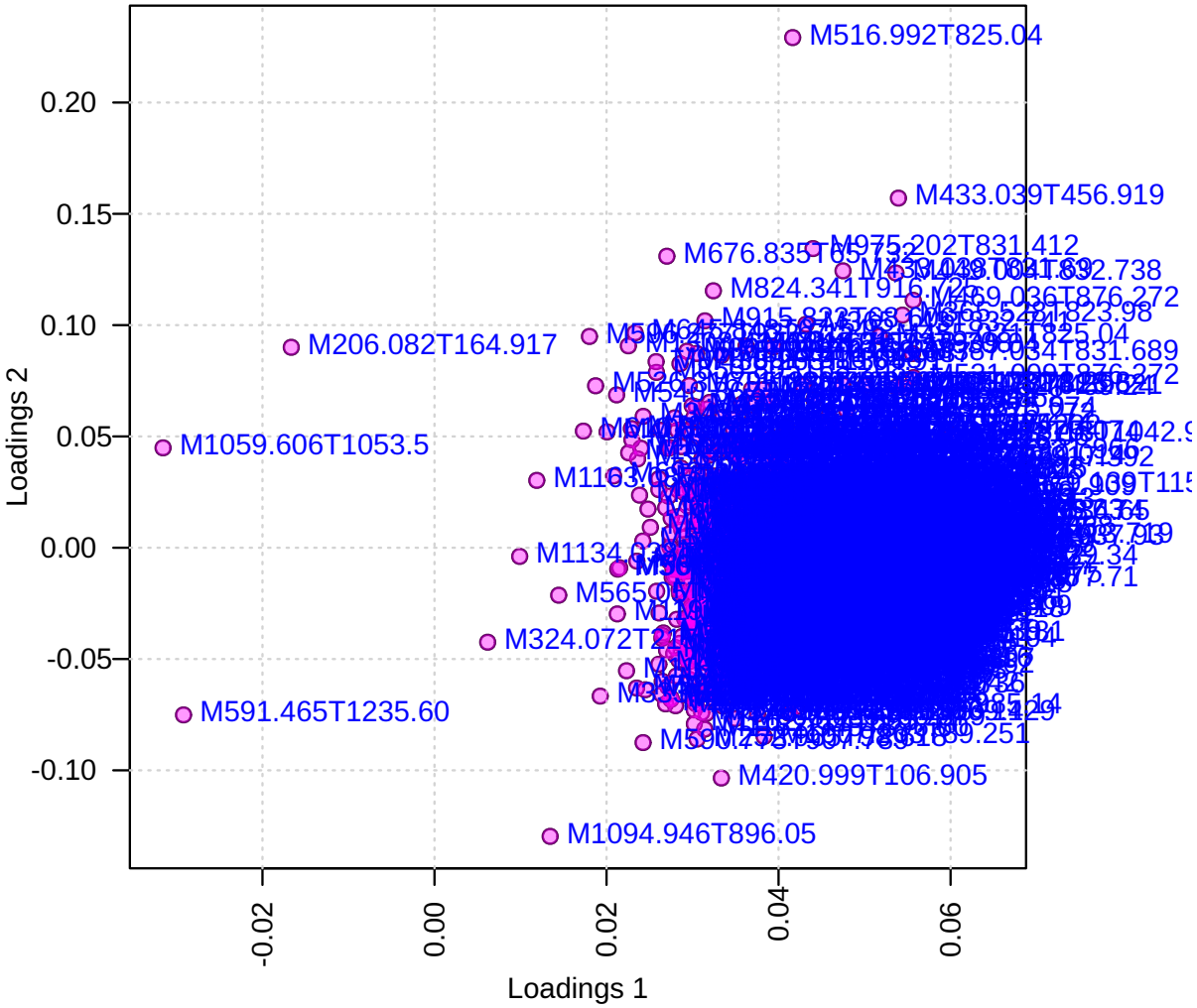

Figure 12: Loadings plot between the selected PCs.

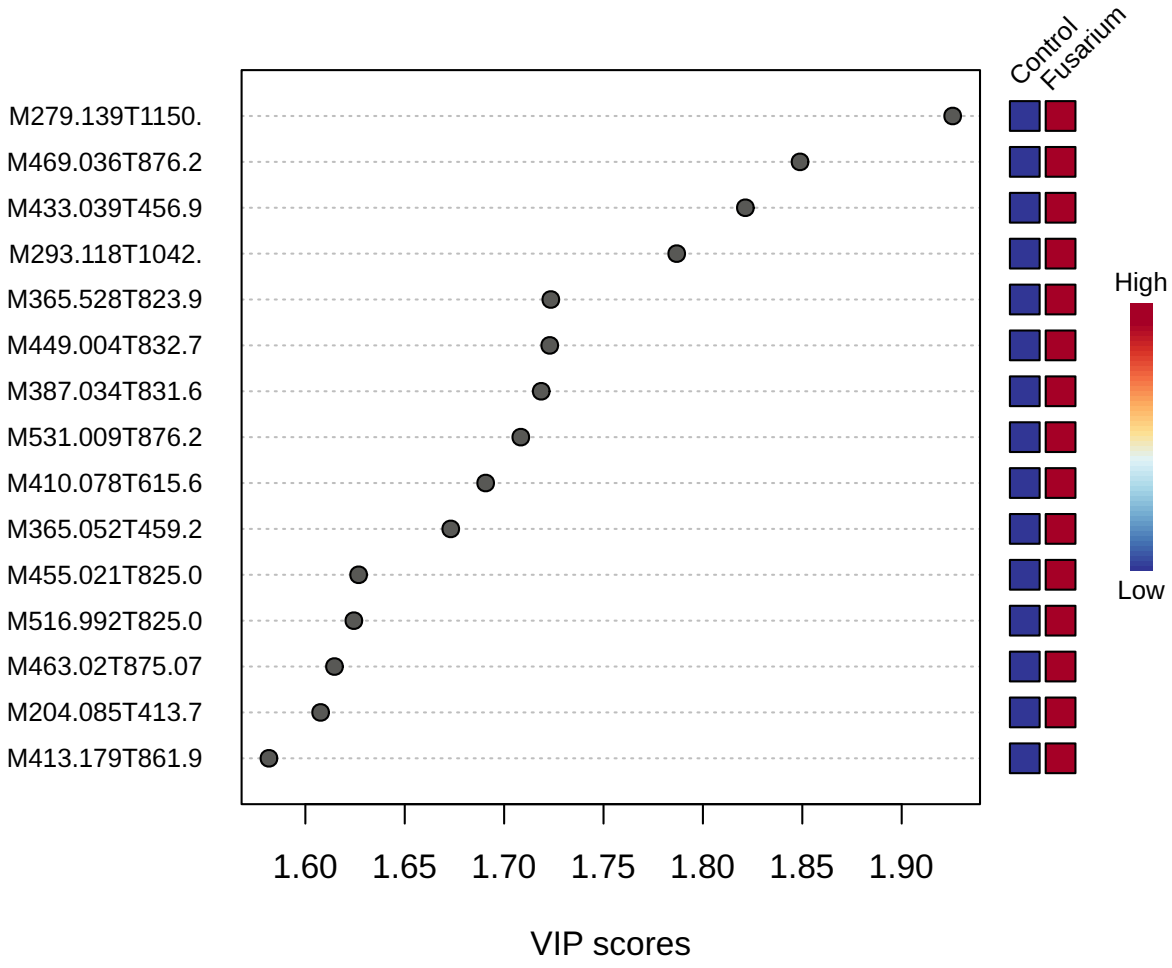

Figure 13: Important features identified by PLS-DA. The colored boxes on the right indicate the relative concentrations of the corresponding metabolite in each group under study.

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.

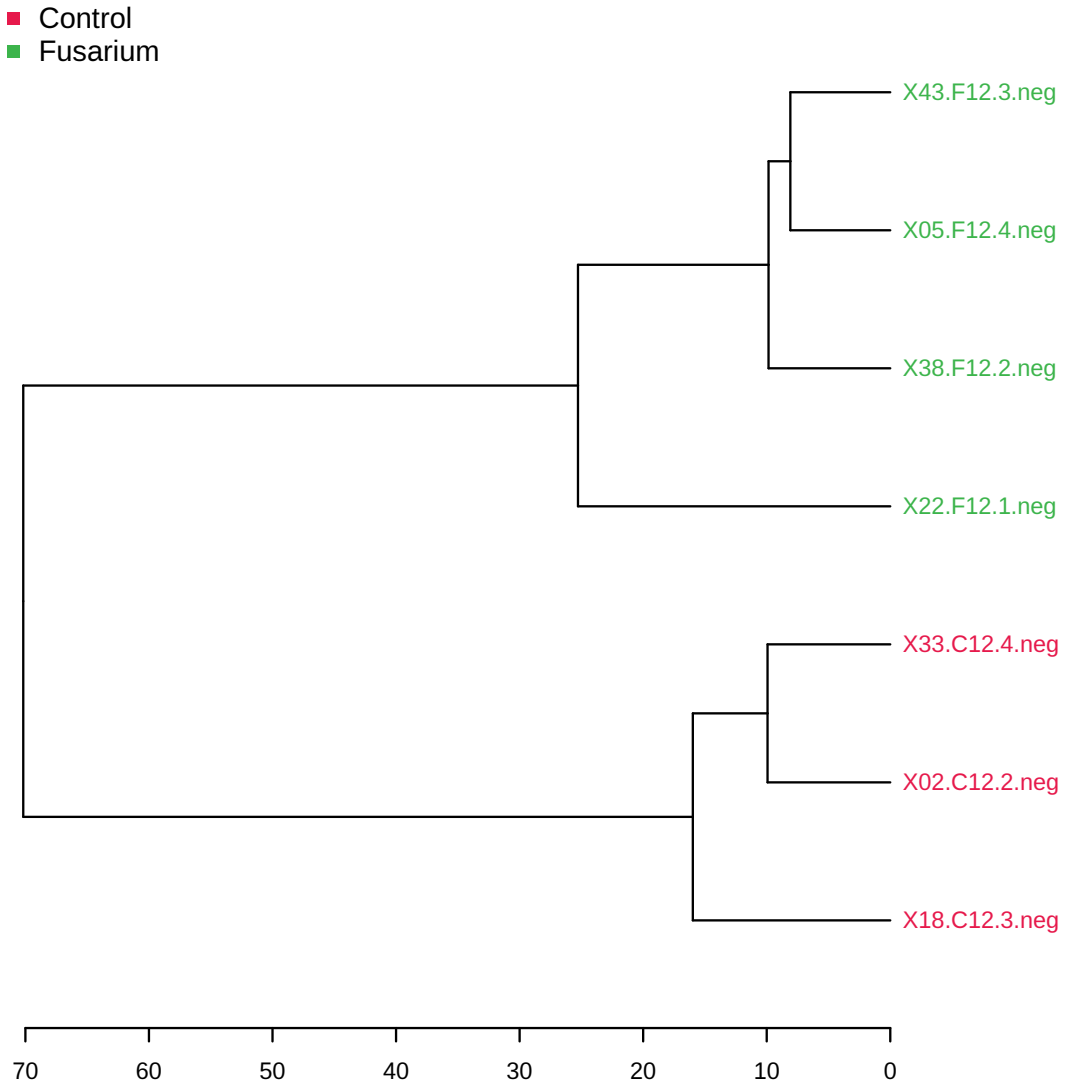

Figure 14: Clustering result shown as dendrogram (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

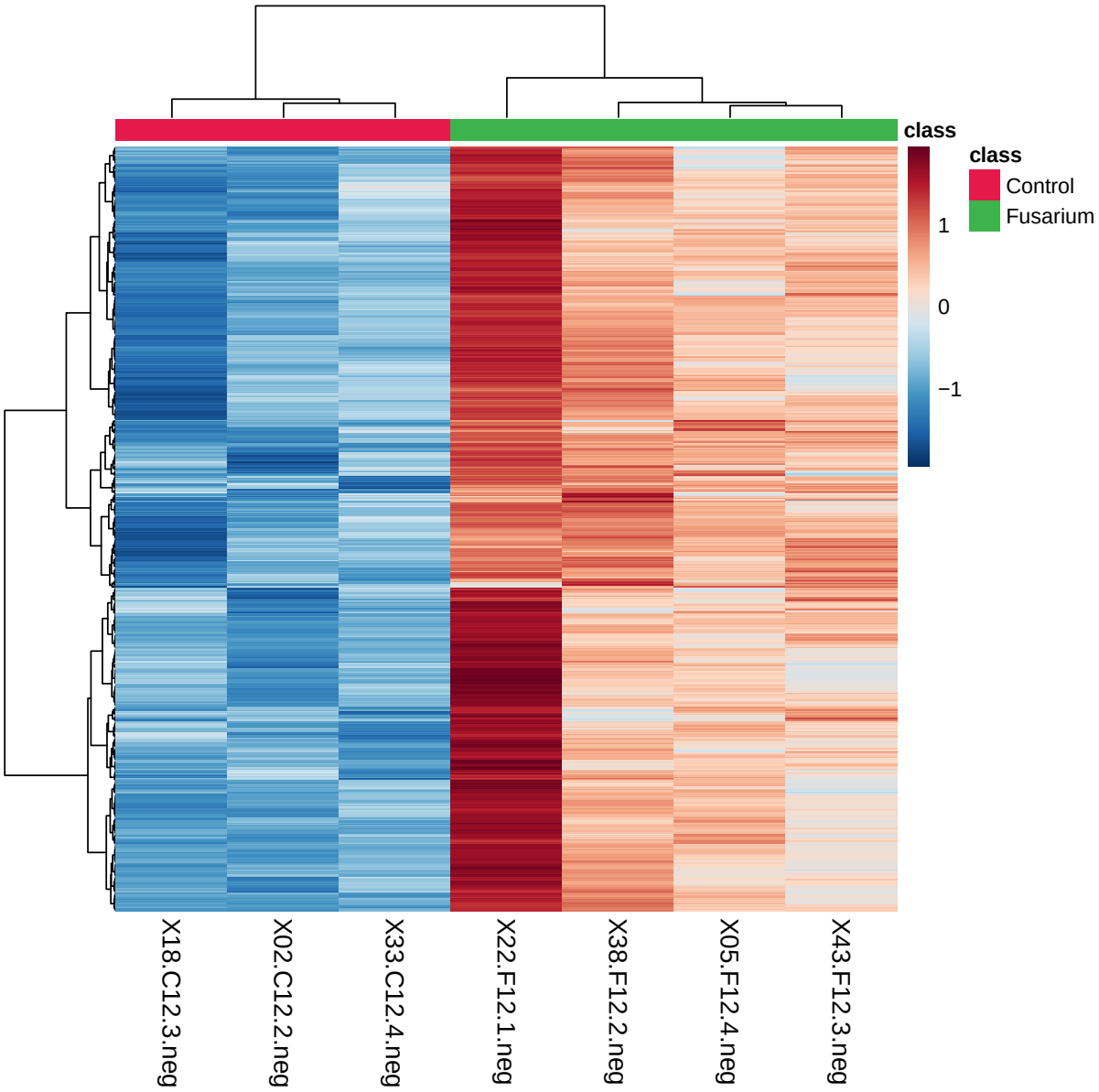

Figure 15: Clustering result shown as heatmap (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

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(\"\")"
[11] "grp.nm.vec <- c(\"Control\",\"Fusarium\")"
[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<-Volcano.Anal(mSet, FALSE, 2.0, 0, F, 0.1, FALSE, \"raw\")"
[28] "mSet<-PlotVolcano(mSet, \"volcano_1_\",1, 0, \"png\", 72, width=NA, -1)"
[29] "mSet<-Volcano.Anal(mSet, FALSE, 2.0, 1, F, 0.1, FALSE, \"raw\")"
[30] "mSet<-PlotVolcano(mSet, \"volcano_2_\",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, 5)"
[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_2_\", \"pdf\", 72, width=NA, \"norm\", \"row\", \"euclide
[51] "mSet<-SaveTransformedData(mSet)"
[52] "mSet<-PreparePDFReport(mSet, \"guest290602822706745484\")\n"
```

The report was generated on Fri Mar 15 06:39:31 2024 with R version 4.3.2 (2023-10-31), OS system: Linux, version: -Ubuntu SMP Tue Jan 9 15:25:40 UTC 2024 .