Metabolomic Data Analysis with MetaboAnalyst 6.0

Name: guest1500149795325540422

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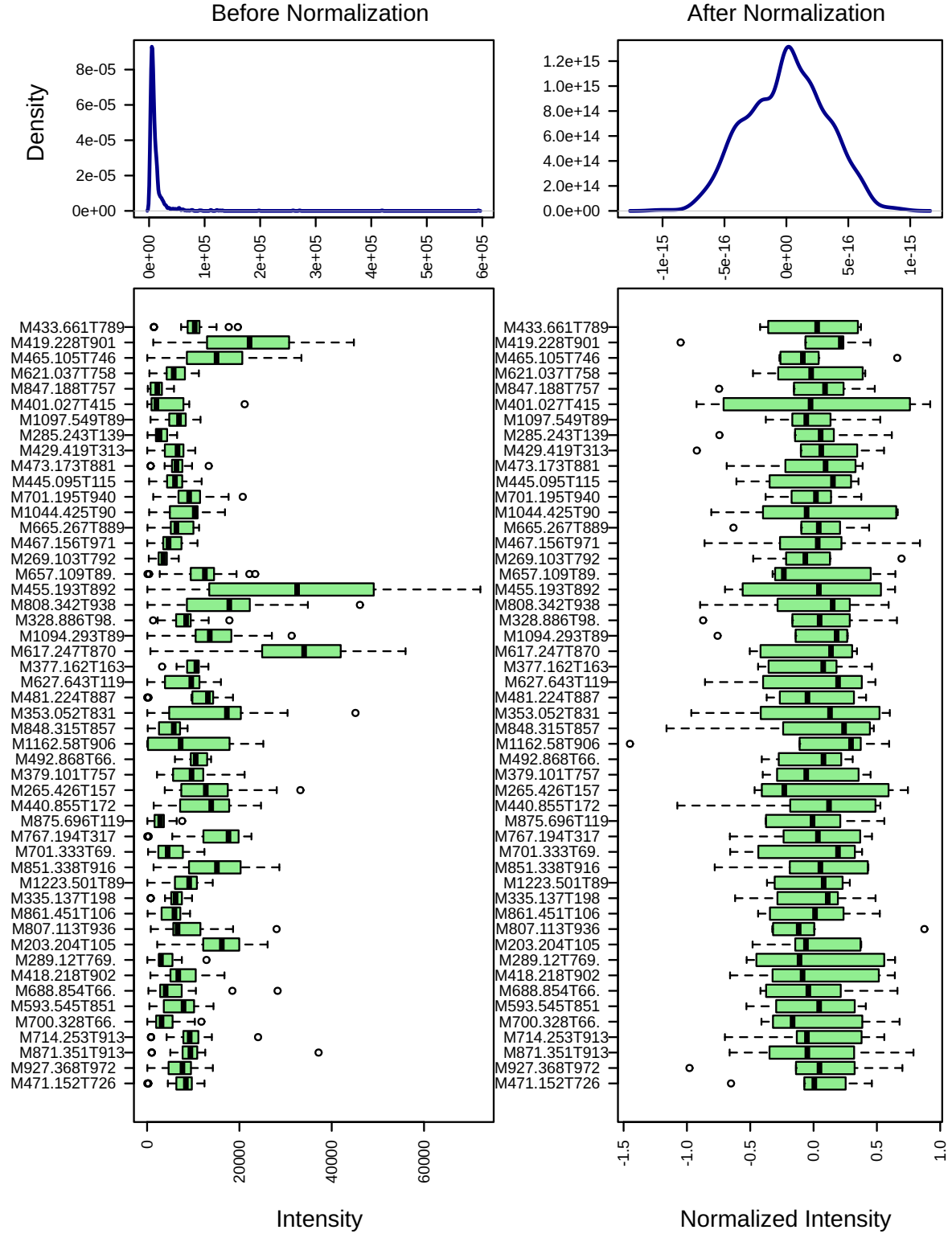
Figure 1: Box plots and kernel density plots before and after normalization. The boxplots show at most 50 features due to space limit. The density plots are based on all samples.

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.
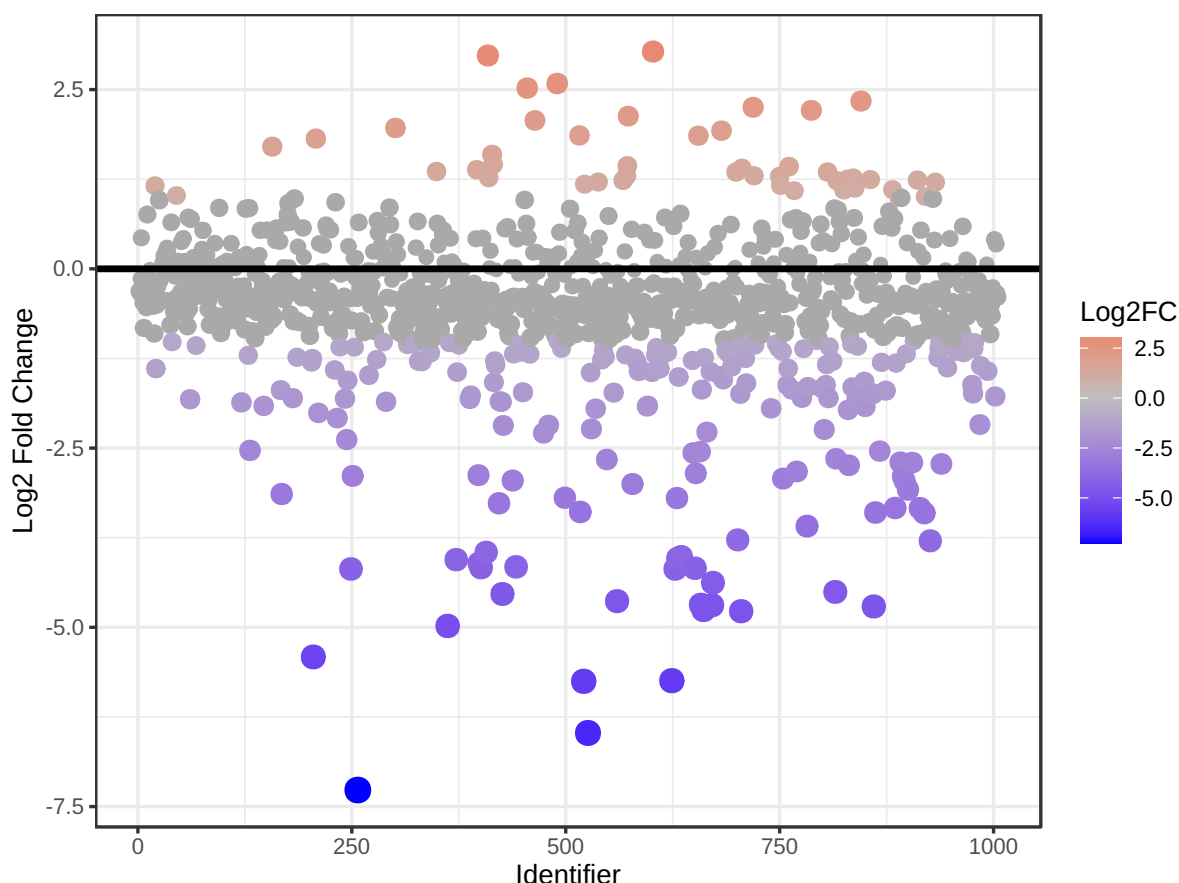

Figure 2: Important features selected by fold-change analysis with threshold 2. The red circles represent features above the threshold. Note the values are on log scale, so that both up-regulated and down-regulated features can be plotted in a symmetrical way

Table 2: Top 50 features identified by fold change analysis

|	Peaks(mz/rt)	Fold Change	log2(FC)
1	M204.085T413.719	0.0064812	-7.2695
2	M684.191T618.708	0.011252	-6.4736
3	M272.073T418.674	0.018539	-5.7533
4	M516.992T825.04	0.018654	-5.7444
5	M433.039T456.919	0.023468	-5.4132
6	M410.078T615.621	0.031664	-4.981
7	M469.036T876.272	0.036515	-4.7754
8	M299.07T414.239	0.036938	-4.7587
9	M433.122T384.02	0.038241	-4.7087
10	M333.041T419.397	0.038681	-4.6922
11	M365.528T823.98	0.038877	-4.685
12	M449.004T832.738	0.040252	-4.6348
13	M206.081T834.983	0.043133	-4.5351
14	M203.083T418.775	0.043999	-4.5064
15	M726.284T1051.577	0.048087	-4.3782
16	M379.565T877.71	0.054931	-4.1862
17	M354.082T204.227	0.054941	-4.186
18	M271.07T418.897	0.055325	-4.1759
19	M455.021T825.04	0.055676	-4.1668
20	M365.052T459.24	0.056075	-4.1565
21	M387.034T831.689	0.058055	-4.1064
22	M641.302T1183.734	0.060116	-4.0561
23	M433.038T831.69	0.060799	-4.0398
24	M531.009T876.272	0.061875	-4.0145
25	M642.302T1183.207	0.06453	-3.9539
26	M319.076T917.392	0.072118	-3.7935
27	M471.046T618.546	0.072784	-3.7802
28	M463.02T875.074	0.083125	-3.5886
29	M339.058T415.891	0.094287	-3.4068
30	M401.027T415.826	0.094821	-3.3986
31	M679.277T1051.577	0.095344	-3.3907
32	M469.019T418.674	0.098825	-3.339
33	M333.062T954.304	0.099075	-3.3353
34	M687.305T1181.604	0.10368	-3.2698
35	M477.061T615.621	0.109	-3.1975
36	M263.13T940.795	0.10937	-3.1927
37	M407.173T414.983	0.11337	-3.1408
38	M824.341T916.725	0.1182	-3.0807
39	M614.901T67.494	8.1704	3.0304
40	M402.053T876.272	0.12485	-3.0017
41	M476.12T661.378	7.8641	2.9753
42	M759.145T875.232	0.12891	-2.9556
43	M857.325T1034.658	0.12916	-2.9527
44	M529.184T941.182	0.13165	-2.9253
45	M317.064T416.942	0.13456	-2.8937
46	M380.071T778.258	0.13505	-2.8884
47	M341.535T617.585	0.13604	-2.8779
48	M409.075T617.585	0.13853	-2.8517
49	M539.034T616.33	0.14097	-2.8266
50	M365.143T832.741	0.14971	-2.7397
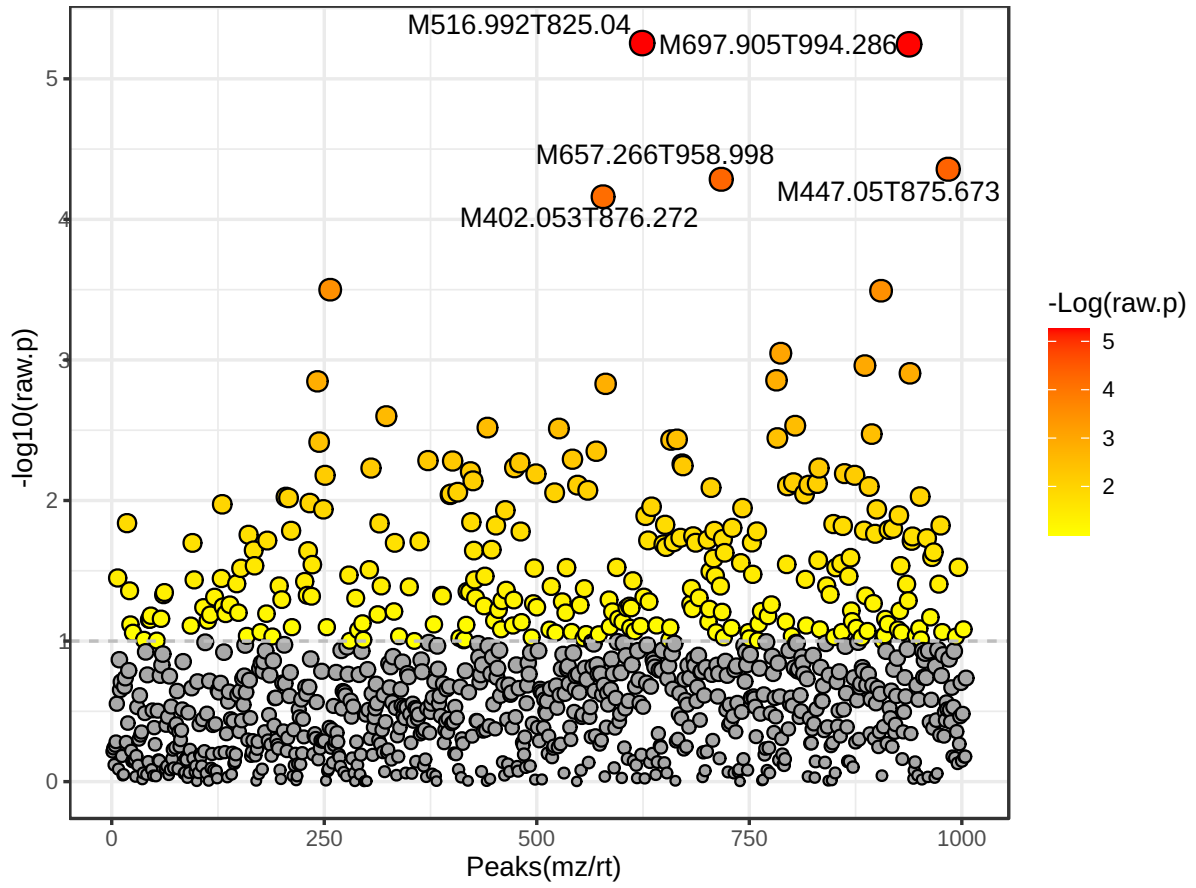
Figure 3: Important features selected by t-tests with threshold 0.1. The red circles represent features above the threshold. Note the p values are transformed by -log10 so that the more significant features (with smaller p values) will be plotted higher on the graph.

Table 3: Top 50 features identified by t-tests

|	Peaks(mz/rt)	t.stat	p.value	-log10(p)	FDR
1	M516.992T825.04	32.164	5.57e-06	5.2541	0.0028552
2	M697.905T994.286	32.004	5.682e-06	5.2455	0.0028552
3	M447.05T875.673	19.133	4.3972e-05	4.3568	0.01303
4	M657.266T958.998	18.352	5.186e-05	4.2852	0.01303
5	M402.053T876.272	17.076	6.8987e-05	4.1612	0.013866
6	M204.085T413.719	11.592	0.00031644	3.4997	0.046206
7	M238.069T801.926	11.542	0.00032184	3.4924	0.046206
8	M404.916T66.832	-8.8609	0.00089584	3.0478	0.11254
9	M543.282T933.008	8.4043	0.0010971	2.9598	0.11438
10	M401.05T875.074	8.13	0.0012451	2.9048	0.11438
11	M463.02T875.074	7.8893	0.0013959	2.8551	0.11438
12	M407.045T415.442	7.8547	0.0014194	2.8479	0.11438
13	M475.366T1425.395	7.7693	0.0014795	2.8299	0.11438
14	M439.083T756.683	6.7498	0.0025118	2.6	0.17548
15	M341.088T304.64	6.4713	0.0029378	2.532	0.17548
16	M365.052T459.24	6.4185	0.0030285	2.5188	0.17548
17	M684.191T618.708	6.3919	0.0030753	2.5121	0.17548
18	M317.064T416.942	6.2361	0.0033689	2.4725	0.17548
19	M289.12T769.482	6.129	0.0035909	2.4448	0.17548
20	M365.136T768.138	6.0921	0.0036715	2.4352	0.17548
21	M365.528T823.98	6.0729	0.0037142	2.4301	0.17548
22	M384.092T841.195	6.0175	0.0038413	2.4155	0.17548
23	M564.929T1201.416	5.776	0.0044614	2.3505	0.1784
24	M852.339T915.443	5.5716	0.0050848	2.2937	0.1784
25	M641.302T1183.734	5.5374	0.0051994	2.284	0.1784
26	M455.021T825.04	5.5268	0.0052356	2.281	0.1784
27	M260.967T89.826	5.4821	0.0053911	2.2683	0.1784
28	M333.041T419.397	5.4521	0.0054989	2.2597	0.1784
29	M726.284T1051.577	5.4096	0.005656	2.2475	0.1784
30	M379.067T781.128	5.3601	0.005846	2.2331	0.1784
31	M555.17T790.902	5.353	0.0058737	2.2311	0.1784
32	M398.263T1317.762	5.3519	0.0058781	2.2308	0.1784
33	M687.305T1181.604	5.2634	0.00624	2.2048	0.1784
34	M401.027T415.826	5.2179	0.0064369	2.1913	0.1784
35	M263.13T940.795	5.2111	0.0064668	2.1893	0.1784
36	M395.28T1470.273	5.1774	0.006618	2.1793	0.1784
37	M380.071T778.258	5.1772	0.0066189	2.1792	0.1784
38	M919.334T939.971	5.047	0.0072465	2.1399	0.1784
39	M429.141T889.195	5.0043	0.0074676	2.1268	0.1784
40	M373.167T930.605	4.9851	0.0075699	2.1209	0.1784
41	M472.886T66.482	-4.9473	0.0077762	2.1092	0.1784
42	M721.31T972.44	4.9472	0.0077769	2.1092	0.1784
43	M341.088T439.204	4.9305	0.0078698	2.104	0.1784
44	M292.14T116.338	4.9164	0.0079497	2.0996	0.1784
45	M469.036T876.272	4.8911	0.0080955	2.0918	0.1784
46	M449.004T832.738	4.8296	0.0084636	2.0724	0.1784
47	M642.302T1183.207	4.7881	0.0087238	2.0593	0.1784
48	M272.073T418.674	4.7756	0.008804	2.0553	0.1784
49	M387.034T831.689	4.7493	0.0089752	2.047	0.1784
50	M203.083T418.775	4.7468	0.008992	2.0461	0.1784
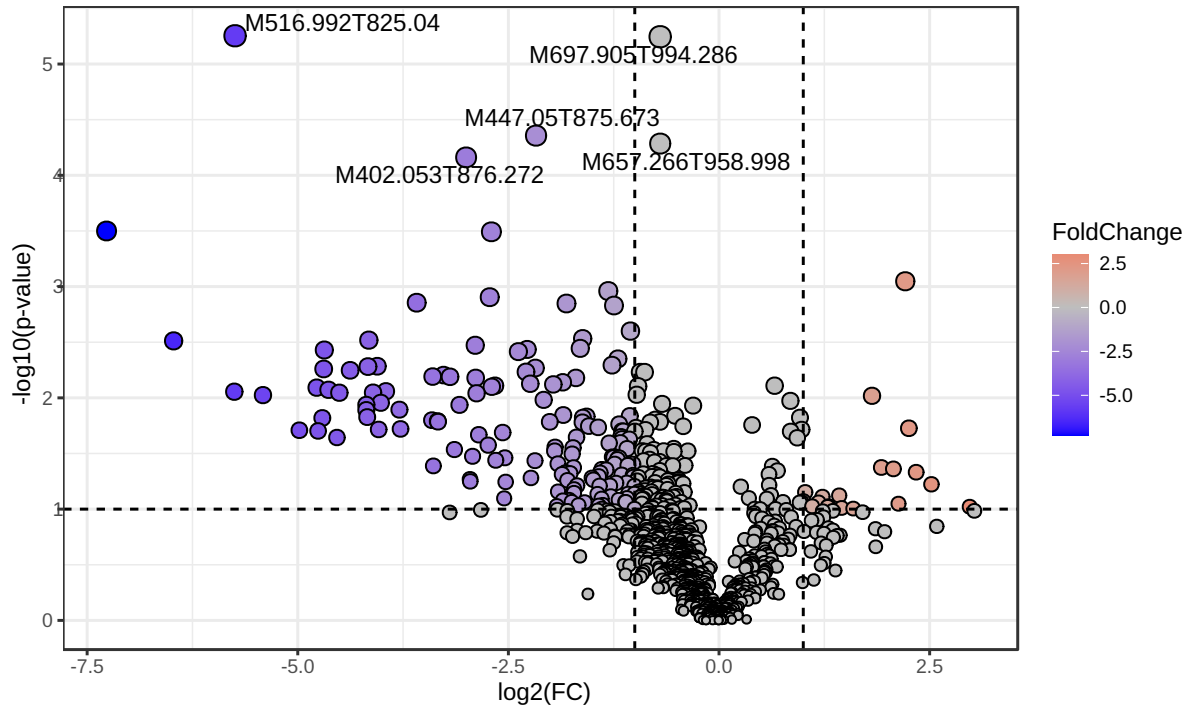
Figure 4: Important features selected by volcano plot with fold change threshold (x) 2 and t-tests threshold (y) 0.1. The red circles represent features above the threshold. Note both fold changes and p values are log transformed. The further its position away from the (0,0), the more significant the feature is.

Table 4: Top 50 features identified by volcano plot

	Peaks(mz/rt)	FC	log2(FC)	raw.pval	-log10(p)
1	M516.992T825.04	0.018654	-5.7444	5.57e-06	5.2541
2	M447.05T875.673	0.22193	-2.1718	4.3972e-05	4.3568
3	M402.053T876.272	0.12485	-3.0017	6.8987e-05	4.1612
4	M204.085T413.719	0.0064812	-7.2695	0.00031644	3.4997
5	M238.069T801.926	0.1537	-2.7018	0.00032184	3.4924
6	M404.916T66.832	4.6286	2.2106	0.00089584	3.0478
7	M543.282T933.008	0.40212	-1.3143	0.0010971	2.9598
8	M401.05T875.074	0.15159	-2.7217	0.0012451	2.9048
9	M463.02T875.074	0.083125	-3.5886	0.0013959	2.8551
10	M407.045T415.442	0.28507	-1.8106	0.0014194	2.8479
11	M475.366T1425.395	0.42137	-1.2469	0.0014795	2.8299
12	M439.083T756.683	0.48246	-1.0515	0.0025118	2.6
13	M341.088T304.64	0.32568	-1.6185	0.0029378	2.532
14	M365.052T459.24	0.056075	-4.1565	0.0030285	2.5188
15	M684.191T618.708	0.011252	-6.4736	0.0030753	2.5121
16	M317.064T416.942	0.13456	-2.8937	0.0033689	2.4725
17	M289.12T769.482	0.31925	-1.6472	0.0035909	2.4448
18	M365.136T768.138	0.20632	-2.277	0.0036715	2.4352
19	M365.528T823.98	0.038877	-4.685	0.0037142	2.4301
20	M384.092T841.195	0.19189	-2.3817	0.0038413	2.4155
21	M564.929T1201.416	0.43529	-1.2	0.0044614	2.3505
22	M852.339T915.443	0.41459	-1.2702	0.0050848	2.2937
23	M641.302T1183.734	0.060116	-4.0561	0.0051994	2.284
24	M455.021T825.04	0.055676	-4.1668	0.0052356	2.281
25	M260.967T89.826	0.22075	-2.1795	0.0053911	2.2683
26	M333.041T419.397	0.038681	-4.6922	0.0054989	2.2597
27	M726.284T1051.577	0.048087	-4.3782	0.005656	2.2475
28	M379.067T781.128	0.20432	-2.2911	0.005846	2.2331
29	M687.305T1181.604	0.10368	-3.2698	0.00624	2.2048
30	M401.027T415.826	0.094821	-3.3986	0.0064369	2.1913
31	M263.13T940.795	0.10937	-3.1927	0.0064668	2.1893
32	M395.28T1470.273	0.30799	-1.6991	0.006618	2.1793
33	M380.071T778.258	0.13505	-2.8884	0.0066189	2.1792
34	M919.334T939.971	0.27649	-1.8547	0.0072465	2.1399
35	M429.141T889.195	0.21161	-2.2405	0.0074676	2.1268
36	M373.167T930.605	0.2558	-1.9669	0.0075699	2.1209
37	M721.31T972.44	0.1583	-2.6592	0.0077769	2.1092
38	M292.14T116.338	0.15418	-2.6973	0.0079497	2.0996
39	M469.036T876.272	0.036515	-4.7754	0.0080955	2.0918
40	M449.004T832.738	0.040252	-4.6348	0.0084636	2.0724
41	M642.302T1183.207	0.06453	-3.9539	0.0087238	2.0593
42	M272.073T418.674	0.018539	-5.7533	0.008804	2.0553
43	M387.034T831.689	0.058055	-4.1064	0.0089752	2.047
44	M203.083T418.775	0.043999	-4.5064	0.008992	2.0461
45	M341.535T617.585	0.13604	-2.8779	0.0090987	2.041
46	M433.039T456.919	0.023468	-5.4132	0.0094462	2.0247
47	M1134.038T907.684	3.5206	1.8158	0.0095858	2.0184
48	M393.081T756.691	0.2363	-2.0813	0.010425	1.9819
49	M531.009T876.272	0.061875	-4.0145	0.011076	1.9556
50	M354.082T204.227	0.054941	-4.186	0.011552	1.9374

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.
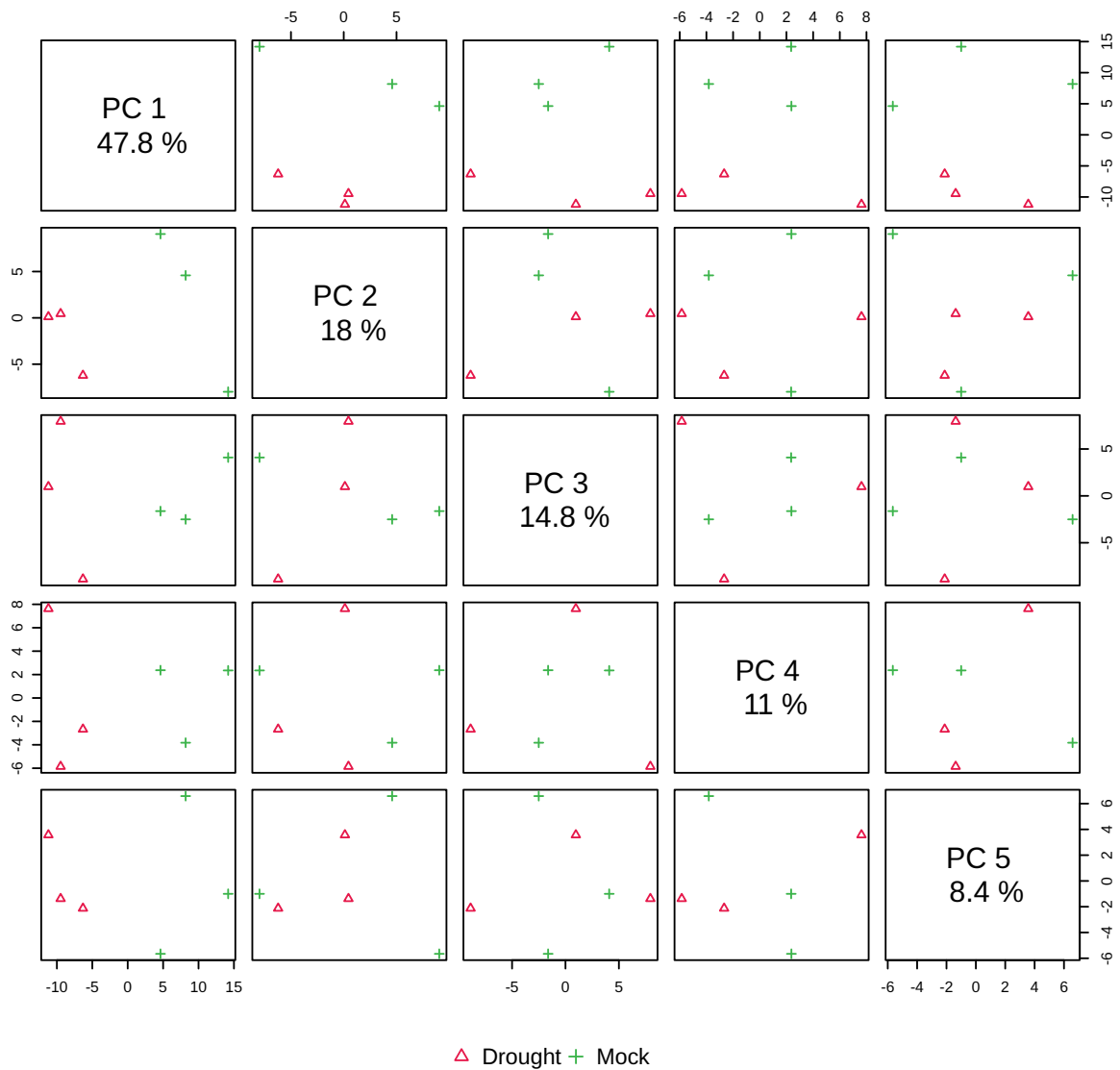
Figure 5: Pairwise score plots between the selected PCs. The explained variance of each PC is shown in the corresponding diagonal cell.
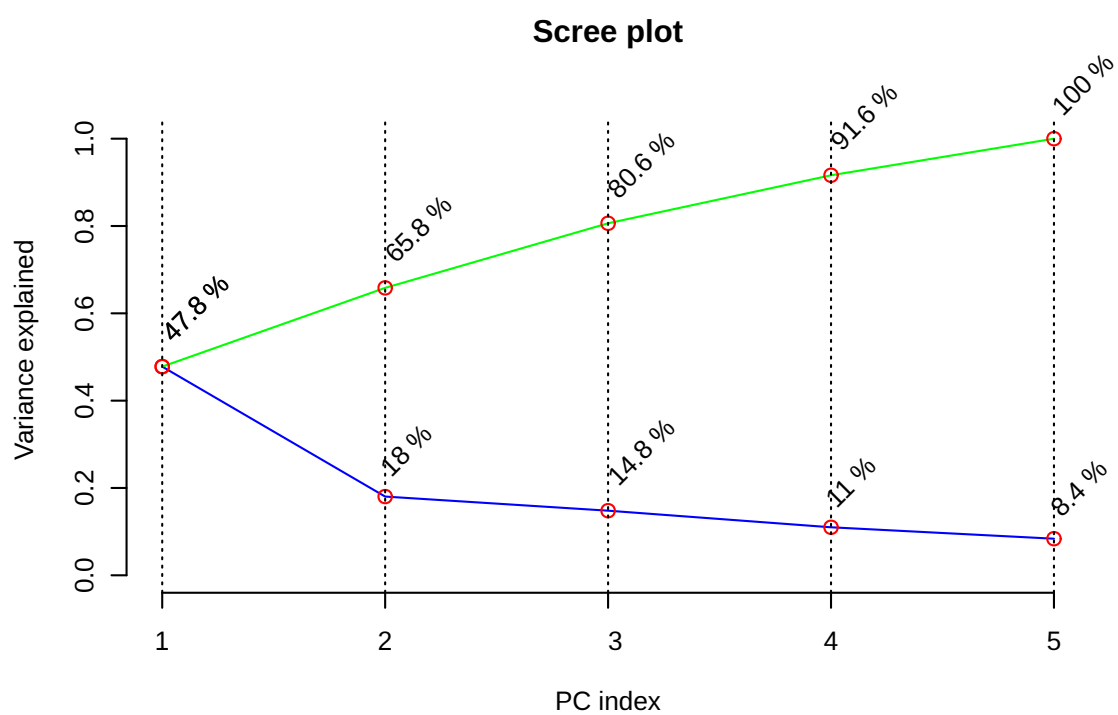
Figure 6: Scree plot shows the variance explained by PCs. The green line on top shows the accumulated variance explained; the blue line underneath shows the variance explained by individual PC.
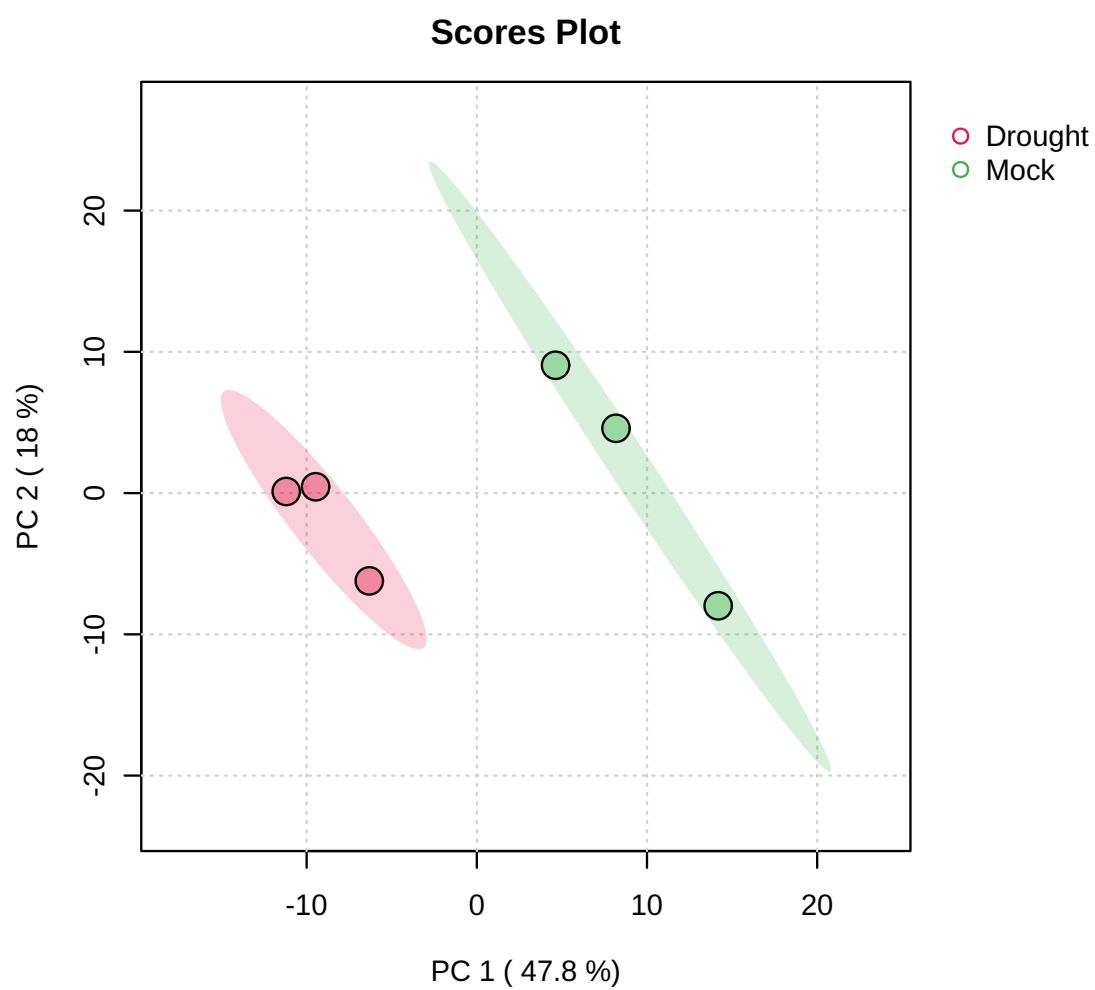
Figure 7: Scores plot between the selected PCs. The explained variances are shown in brackets.
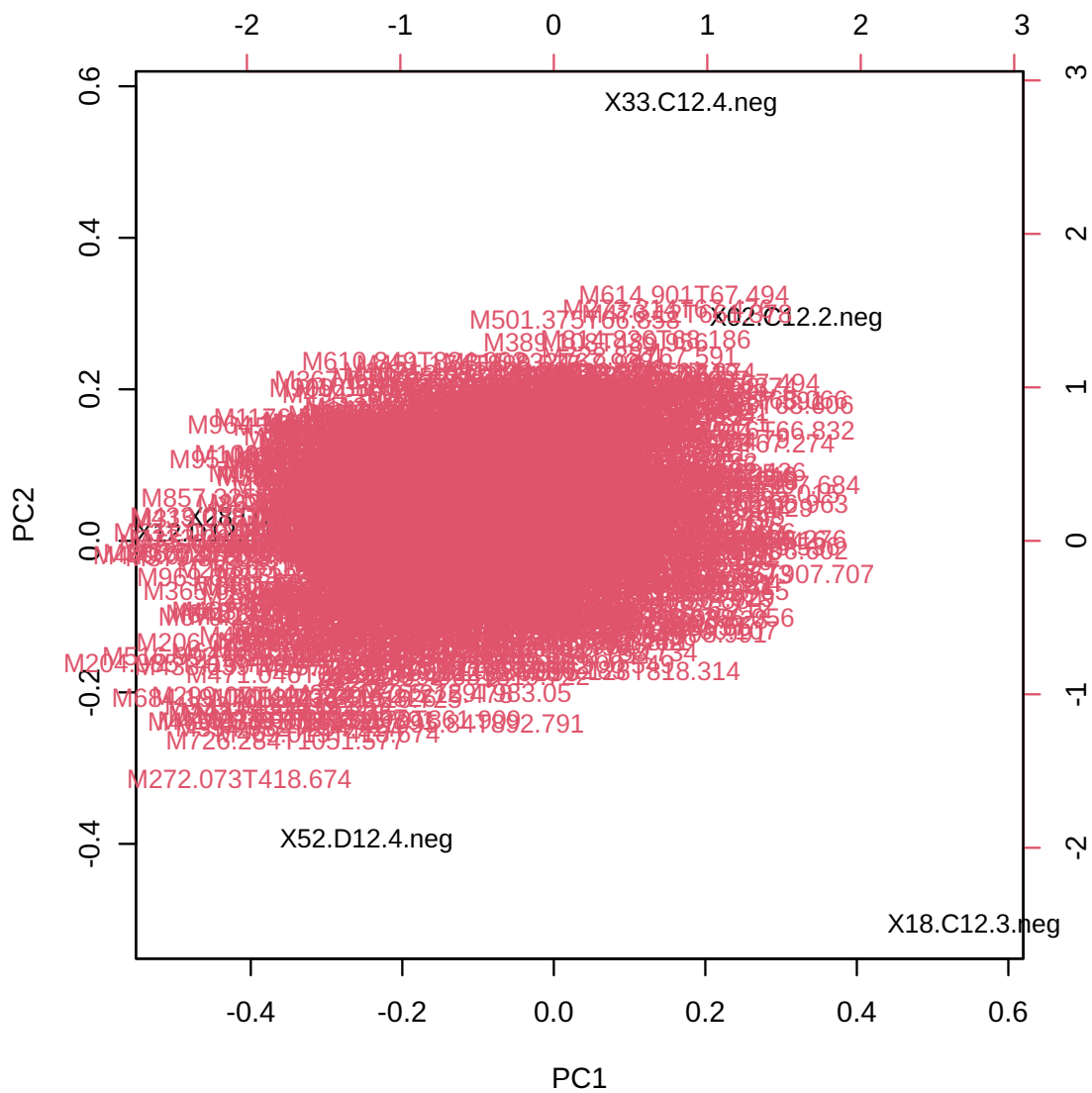
Figure 8: PCA biplot between the selected PCs. Note, you may want to test different centering and scaling normalization methods for the biplot to be displayed properly.

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
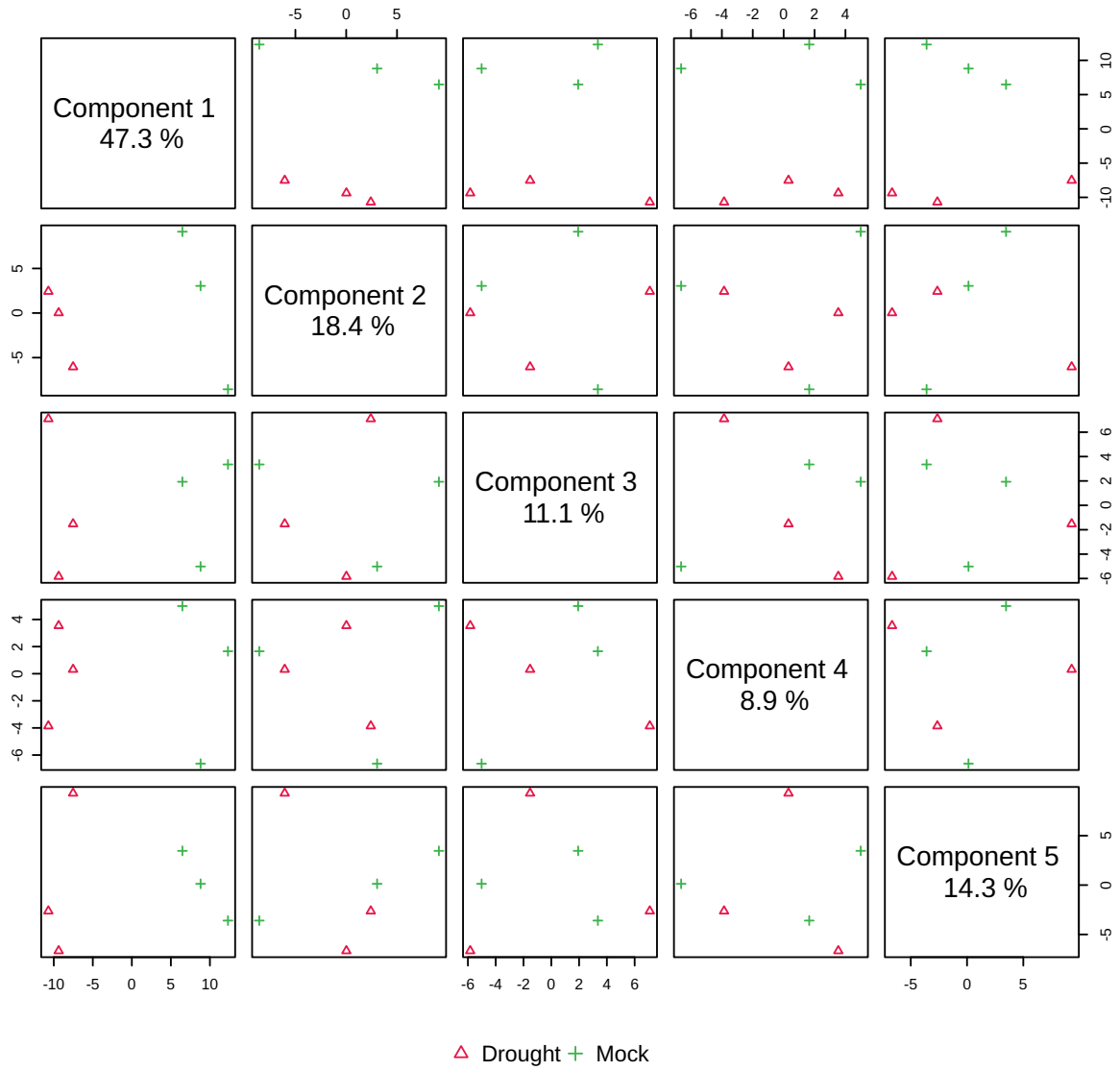
Figure 9: Pairwise scores plots between the selected components. The explained variance of each component is shown in the corresponding diagonal cell.
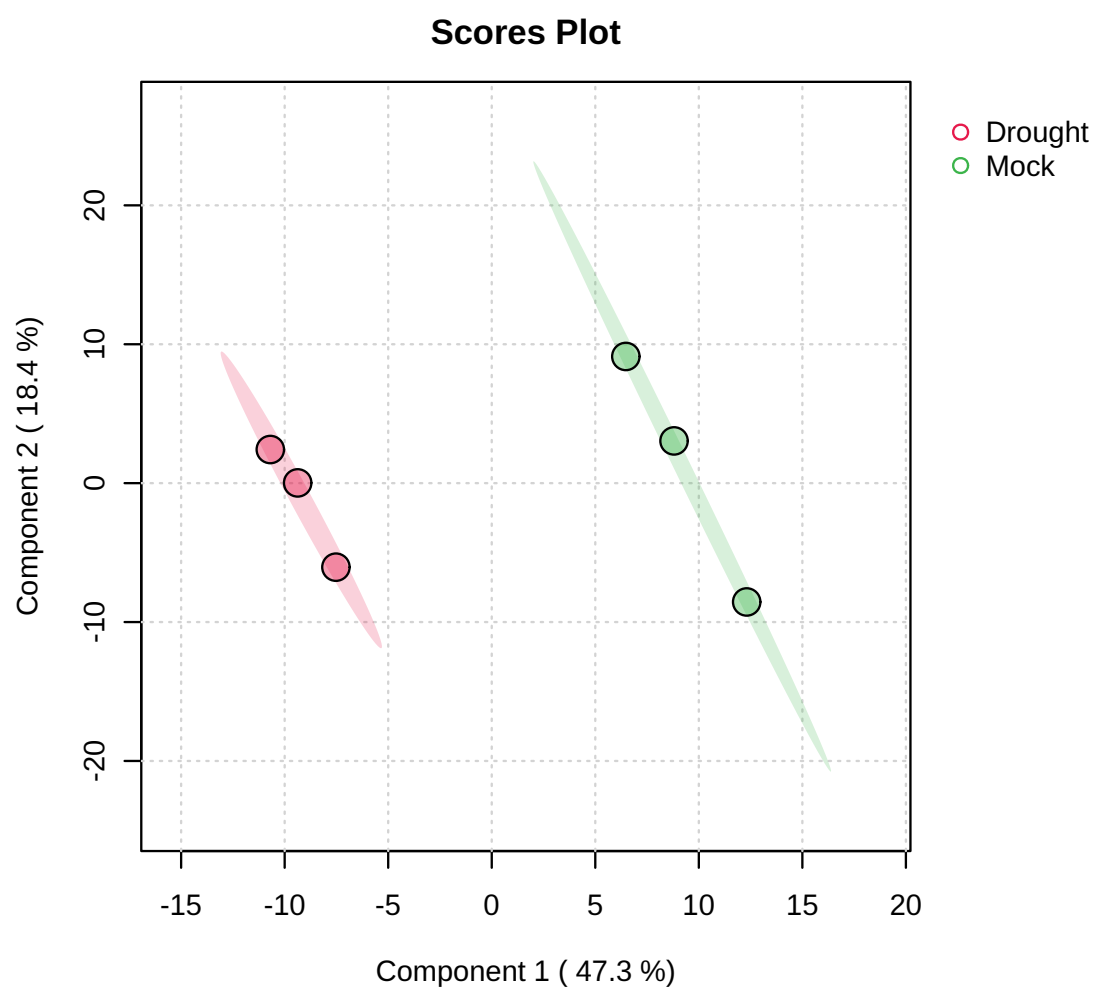
Figure 10: Scores plot between the selected PCs. The explained variances are shown in brackets.
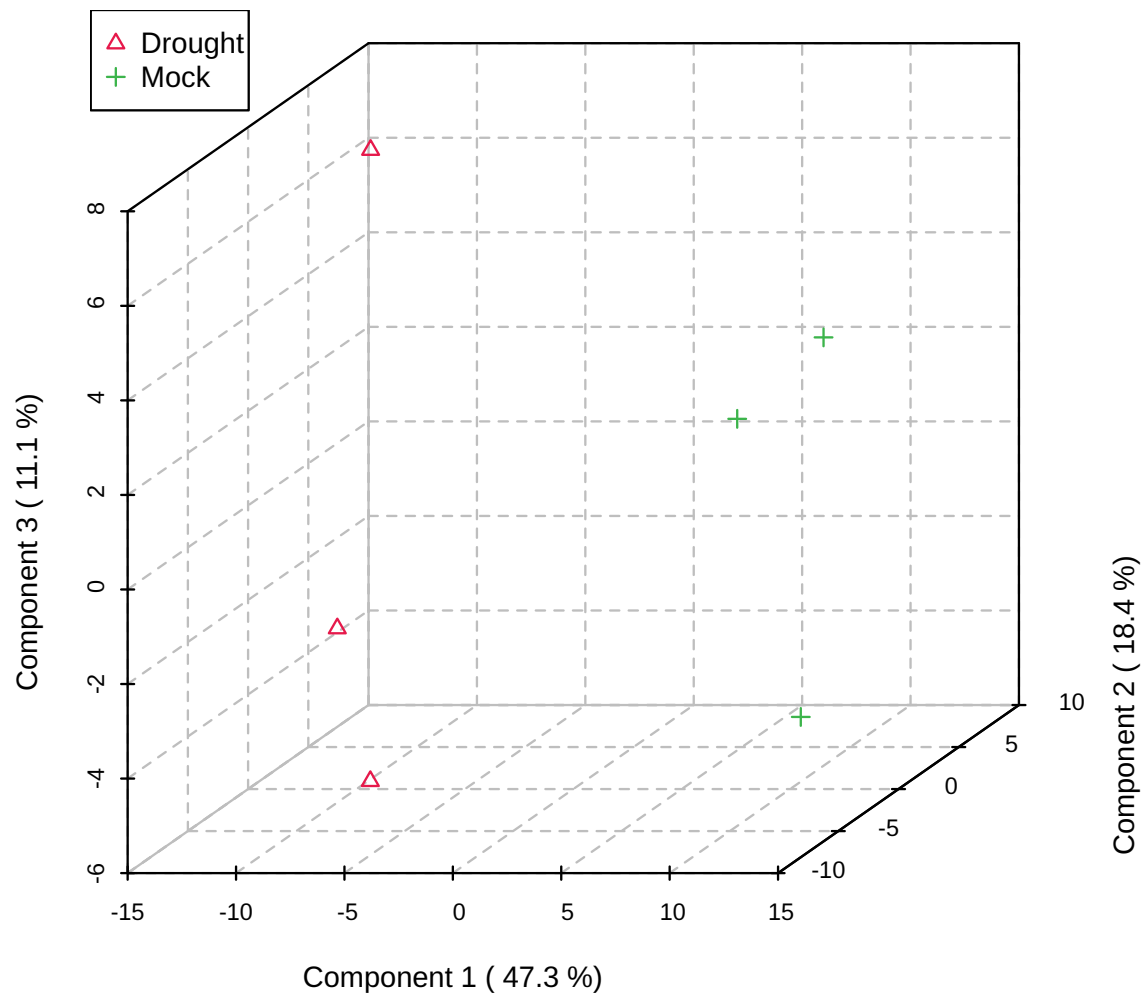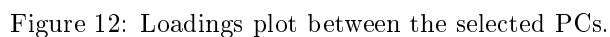
Figure 11: 3D scores plot between the selected PCs. The explained variances are shown in brackets.

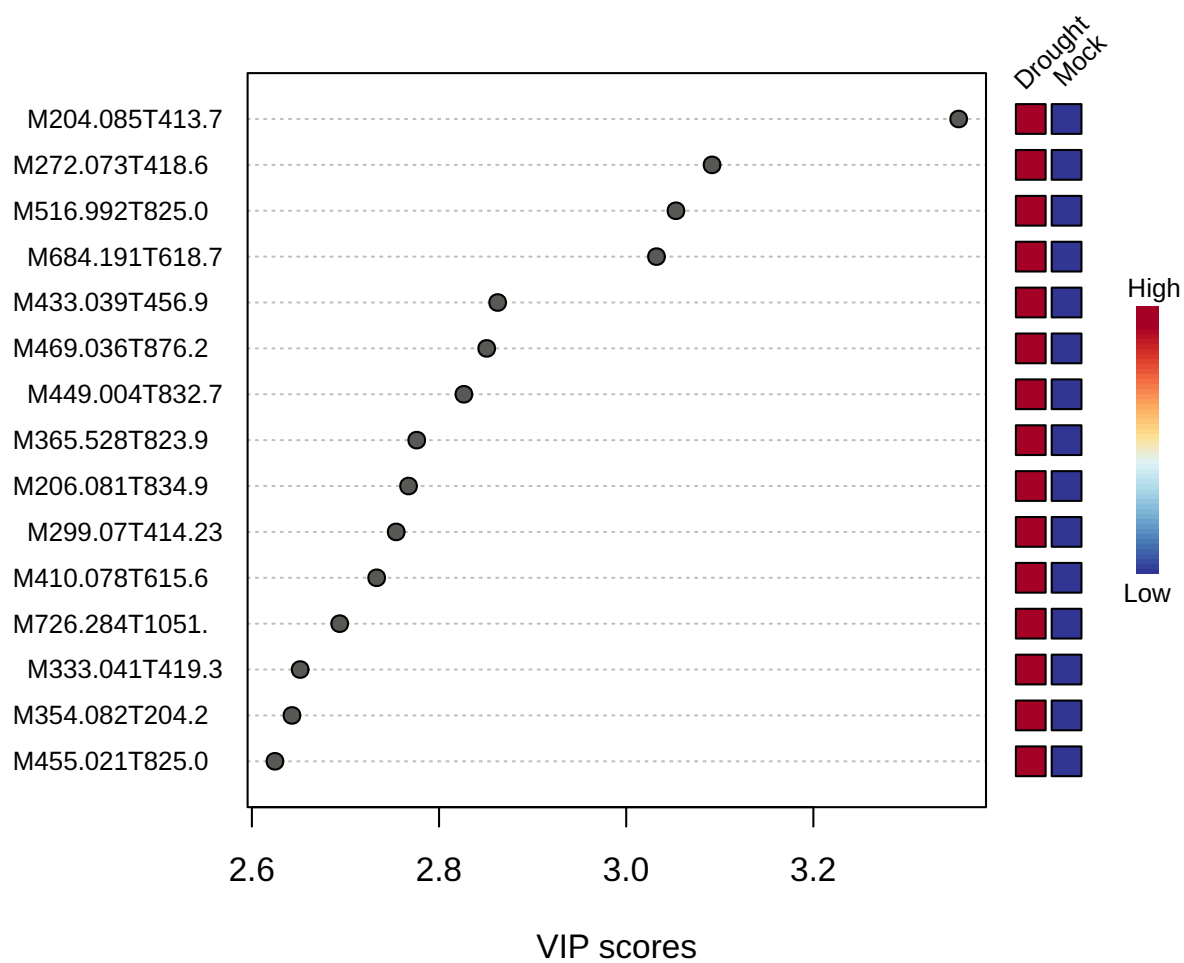Figure 12: Loadings plot between the selected PCs.

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.
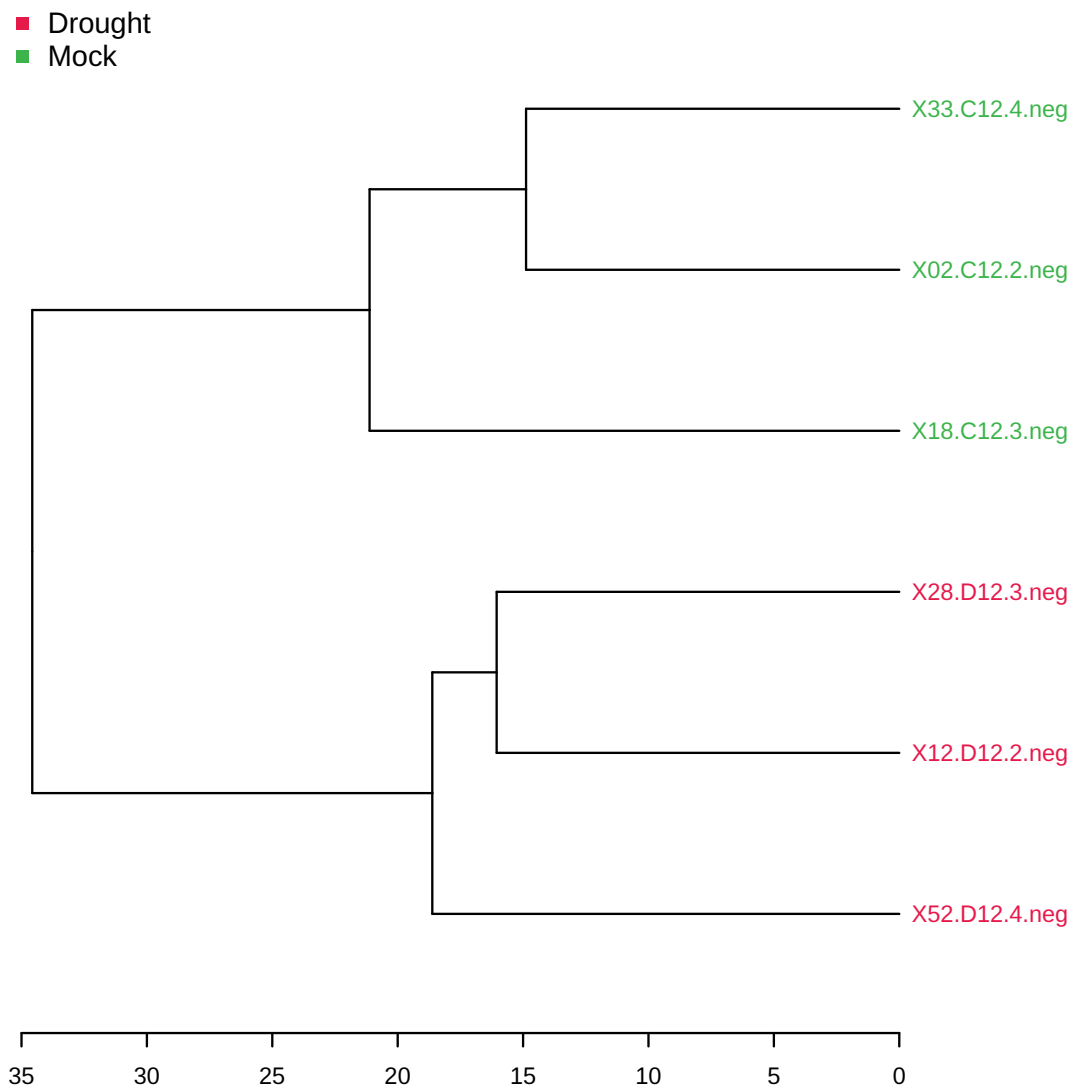

Figure 14: Clustering result shown as dendrogram (distance measure using `euclidean`, and clustering algorithm using `ward.D`).
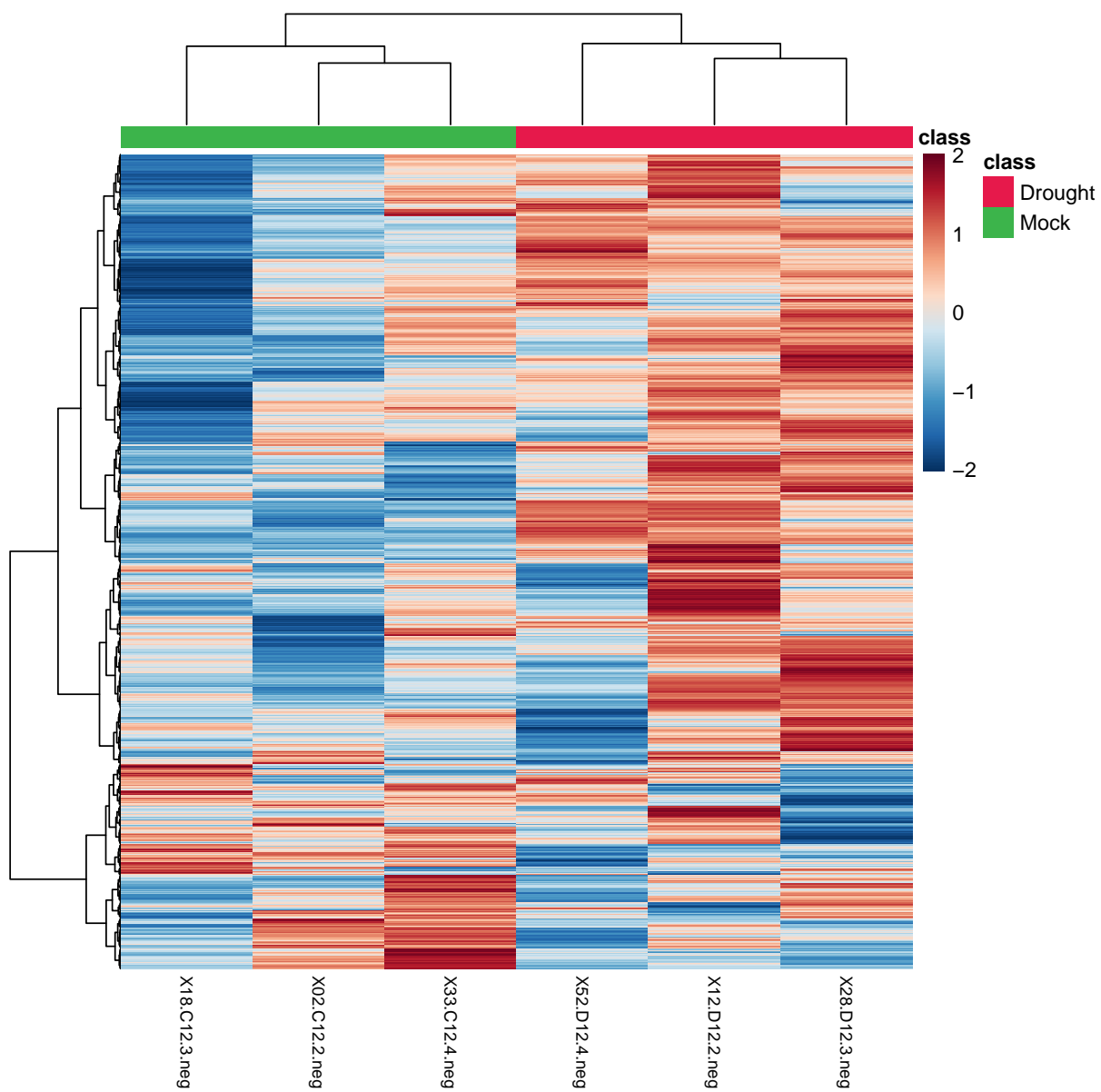
Figure 15: Clustering result shown as heatmap (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

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(\"Drought\",\"Mock\")"
[12] "mSet<-UpdateData(mSet, T)"
[13] "mSet<-PreparePrenormData(mSet)"
[14] "feature.nm.vec <- c(\"\")"
[15] "smpl.nm.vec <- c(\"X44.D12.1.neg\",\"X09.X12.3.neg\")"
[16] "grp.nm.vec <- c(\"Drought\",\"Mock\")"
[17] "mSet<-UpdateData(mSet, T)"
[18] "mSet<-PreparePrenormData(mSet)"
[19] "mSet<-Normalization(mSet, \"CompNorm\", \"LogNorm\", \"ParetoNorm\", \"sodium_formate\", ratio
[20] "mSet<-PlotNormSummary(mSet, \"norm_0_\", \"png\", 72, width=NA)"
[21] "mSet<-PlotSampleNormSummary(mSet, \"snorm_0_\", \"png\", 72, width=NA)"
[22] "mSet<-FC.Anal(mSet, 2.0, 0, FALSE)"
[23] "mSet<-PlotFC(mSet, \"fc_0_\", \"png\", 72, width=NA)"
[24] "mSet<-FC.Anal(mSet, 2.0, 1, FALSE)"
[25] "mSet<-PlotFC(mSet, \"fc_1_\", \"png\", 72, width=NA)"
[26] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, TRUE, \"fdr\", FALSE)"
[27] "mSet<-PlotTT(mSet, \"tt_0_\", \"png\", 72, width=NA)"
[28] "mSet<-UpdateLoadingCmpd(mSet, \"M516.992T825.04\")"
[29] "mSet<-SetCmpdSummaryType(mSet, \"violin\")"
[30] "mSet<-PlotCmpdSummary(mSet, \"M516.992T825.04\",\"NA\",\"NA\", 0, \"png\", 72)"
[31] "mSet<-SetCmpdSummaryType(mSet, \"boxplot\")"
[32] "mSet<-PlotCmpdSummary(mSet, \"M516.992T825.04\",\"NA\",\"NA\", 1, \"png\", 72)"
[33] "mSet<-PlotHeatMap(mSet, \"heatmap_1_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\
[34] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, TRUE, \"raw\", FALSE)"
[35] "mSet<-PlotTT(mSet, \"tt_1_\", \"png\", 72, width=NA)"
[36] "mSet<-Volcano.Anal(mSet, FALSE, 2.0, 0, F, 0.1, TRUE, \"raw\")"
[37] "mSet<-PlotVolcano(mSet, \"volcano_0_\",1, 0, \"png\", 72, width=NA)"
[38] "mSet<-Volcano.Anal(mSet, FALSE, 2.0, 1, F, 0.1, TRUE, \"raw\")"
[39] "mSet<-PlotVolcano(mSet, \"volcano_1_\",1, 0, \"png\", 72, width=NA)"
[40] "mSet<-PlotVolcano(mSet, \"volcano_1_\",1, 0, \"pdf\", 72, width=NA)"
[41] "mSet<-PlotHCTree(mSet, \"tree_0_\", \"png\", 72, width=NA, \"euclidean\", \"ward.D\")"
[42] "mSet<-PCA.Anal(mSet)"
[43] "mSet<-PlotPCAPairSummary(mSet, \"pca_pair_0_\", \"png\", 72, width=NA, 5)"
[44] "mSet<-PlotPCAScree(mSet, \"pca_scree_0_\", \"png\", 72, width=NA, 5)"
[45] "mSet<-PlotPCA2DScore(mSet, \"pca_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[46] "mSet<-PlotPCALoading(mSet, \"pca_loading_0_\", \"png\", 72, width=NA, 1,2);"
[47] "mSet<-PlotPCABiplot(mSet, \"pca_biplot_0_\", \"png\", 72, width=NA, 1,2)"
[48] "mSet<-PlotPCA3DLoading(mSet, \"pca_loading3d_0_\", \"json\", 1,2,3)"
[49] "mSet<-PLSR.Anal(mSet, reg=TRUE)"
[50] "mSet<-PlotPLSPairSummary(mSet, \"pls_pair_0_\", \"png\", 72, width=NA, 5)"
[51] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[52] "mSet<-PlotPLS3DScoreImg(mSet, \"pls_score3d_0_\", \"png\", 72, width=NA, 1,2,3, 40)"
[53] "mSet<-PlotPLSLoading(mSet, \"pls_loading_0_\", \"png\", 72, width=NA, 1, 2);"
[54] "mSet<-PlotPLS3DLoading(mSet, \"pls_loading3d_0_\", \"json\", 1,2,3)"
[55] "mSet<-PlotPLS.Imp(mSet, \"pls_imp_0_\", \"png\", 72, width=NA, \"vip\", \"Comp. 1\", 15,FALSE)
[56] "mSet<-SaveTransformedData(mSet)"
```

```
[57] "mSet<-PreparePDFReport(mSet, \"guest1500149795325540422\")\n"
```

The report was generated on Tue Mar 12 14:21:33 2024 with R version 4.3.2 (2023-10-31), OS system: Linux, version: -Ubuntu SMP Tue Jan 9 15:25:40 UTC 2024 .