

# Metabolomic Data Analysis with MetaboAnalyst 6.0

Name: guest16724914738269757780

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 <sup>1</sup>. Please choose the one that is the most appropriate for your data.

---

<sup>1</sup>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<sup>2</sup>.

*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 btween 500 and 1000, 25% of variables will be removed; And 40% of variabed 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

<sup>2</sup>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)<sup>3</sup>
  - 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.

---

<sup>3</sup>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

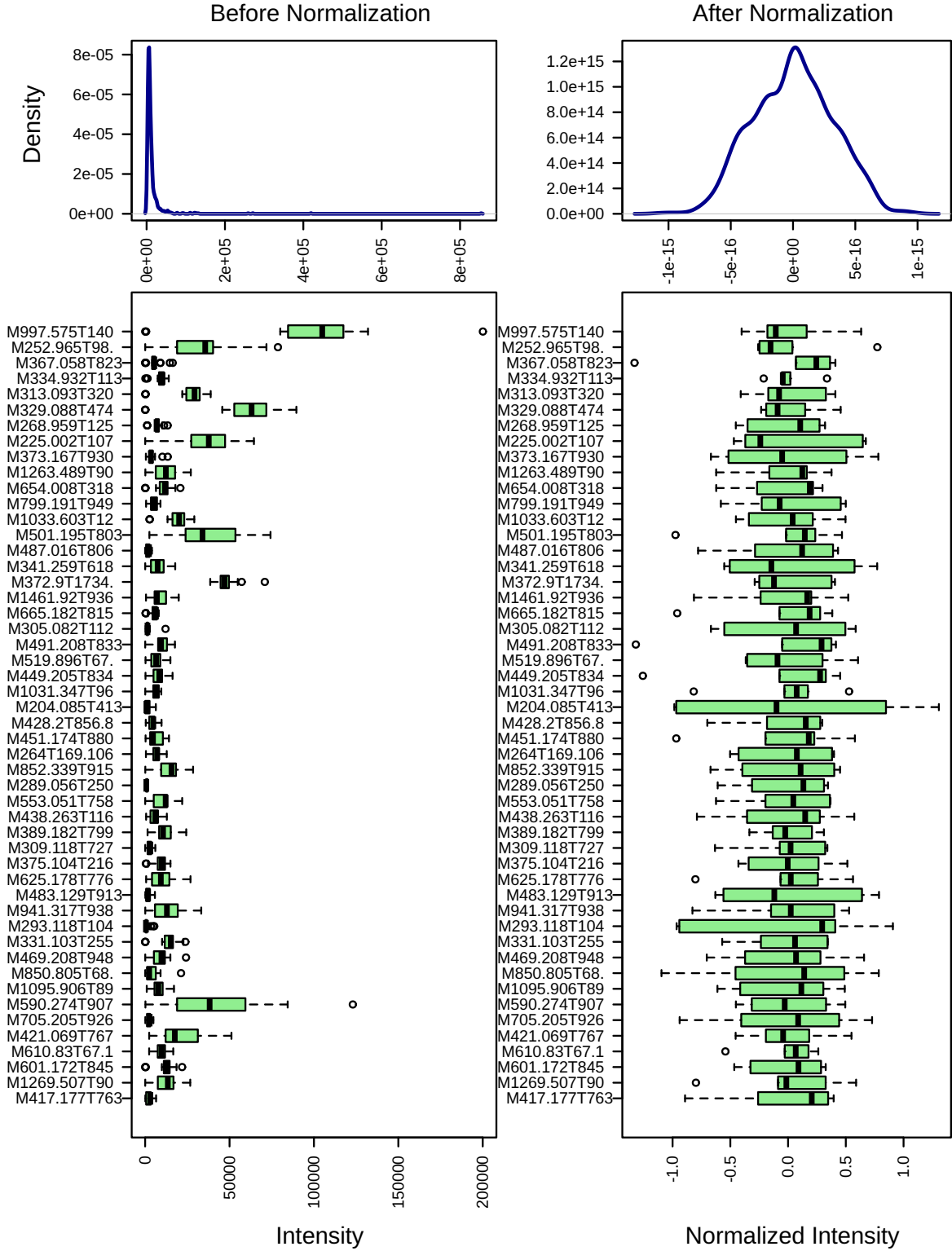


Figure 1: Box plots and kernel density plots before and after normalization. The boxplots show at most 50 features due to space limit. The density plots are based on all samples.

## 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 analysis.

## 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 significant 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;

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 log<sub>2</sub> scale, so that same fold change (up/down regulated) will have the same distance to the zero baseline.

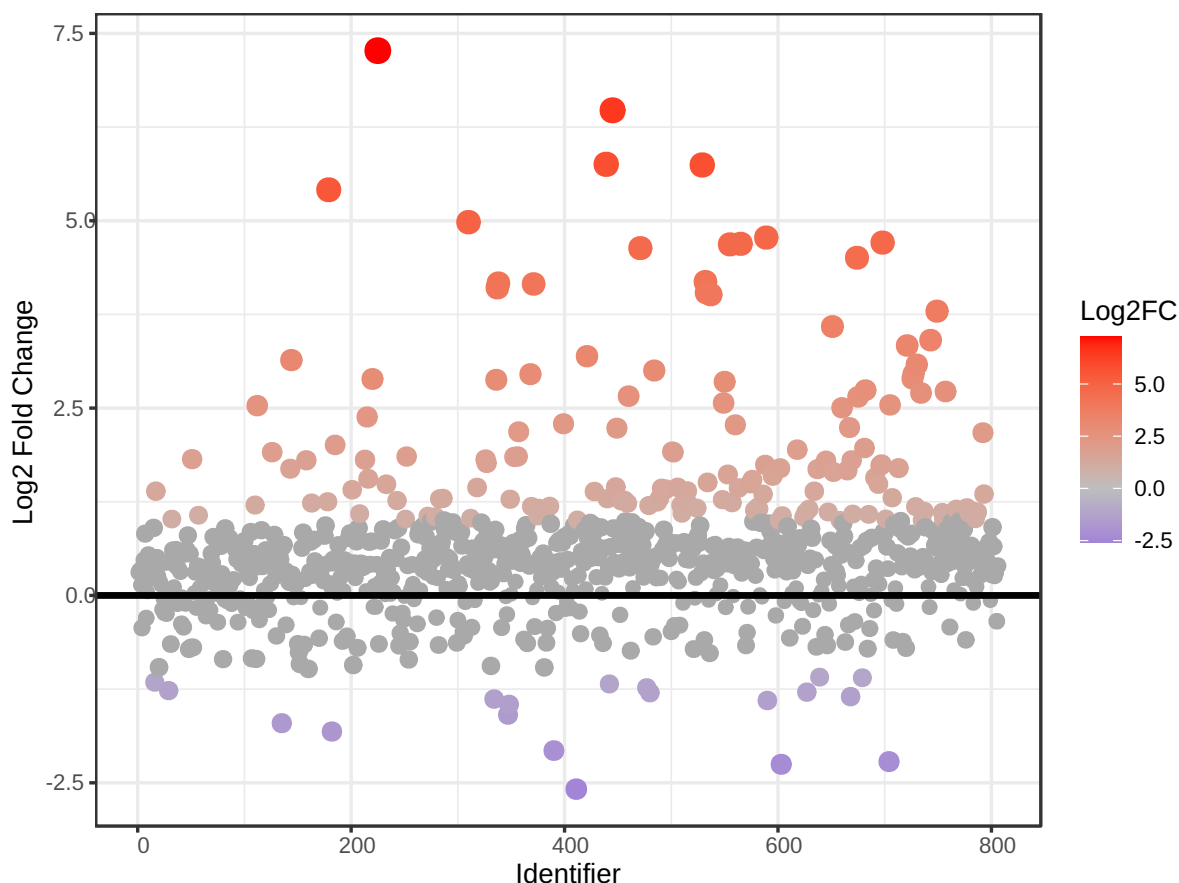


Figure 2: Important features selected by fold-change analysis with threshold 2. The red circles represent features above the threshold. Note the values are on log scale, so that both up-regulated and down-regulated features can be plotted in a symmetrical way

Table 2: Top 50 features identified by fold change analysis

	Peaks(mz/rt)	Fold Change	log2(FC)
1	M204.085T413.719	154.29	7.2695
2	M684.191T618.708	88.869	6.4736
3	M272.073T418.674	53.94	5.7533
4	M516.992T825.04	53.609	5.7444
5	M433.039T456.919	42.611	5.4132
6	M410.078T615.621	31.581	4.981
7	M469.036T876.272	27.386	4.7754
8	M433.122T384.02	26.15	4.7087
9	M333.041T419.397	25.852	4.6922
10	M365.528T823.98	25.722	4.685
11	M449.004T832.738	24.844	4.6348
12	M203.083T418.775	22.728	4.5064
13	M379.565T877.71	18.205	4.1862
14	M455.021T825.04	17.961	4.1668
15	M365.052T459.24	17.833	4.1565
16	M387.034T831.689	17.225	4.1064
17	M433.038T831.69	16.448	4.0398
18	M531.009T876.272	16.162	4.0145
19	M319.076T917.392	13.866	3.7935
20	M463.02T875.074	12.03	3.5886
21	M339.058T415.891	10.606	3.4068
22	M333.062T954.304	10.093	3.3353
23	M263.13T940.795	9.1432	3.1927
24	M407.173T414.983	8.8204	3.1408
25	M824.341T916.725	8.4603	3.0807
26	M402.053T876.272	8.0093	3.0017
27	M759.145T875.232	7.7576	2.9556
28	M857.325T1034.658	7.7422	2.9527
29	M317.064T416.942	7.4319	2.8937
30	M380.071T778.258	7.4045	2.8884
31	M341.535T617.585	7.3506	2.8779
32	M409.075T617.585	7.2184	2.8517
33	M365.143T832.741	6.6795	2.7397
34	M401.05T875.074	6.5966	2.7217
35	M238.069T801.926	6.5063	2.7018
36	M721.31T972.44	6.317	2.6592
37	M341.465T625.973	6.276	2.6499
38	M273.314T67.476	0.1667	-2.5846
39	M661.142T919.275	5.9298	2.568
40	M349.057T865.046	5.8254	2.5424
41	M279.139T1150.344	5.7863	2.5326
42	M975.202T831.412	5.6734	2.5042
43	M384.092T841.195	5.2114	2.3817
44	M379.067T781.128	4.8944	2.2911
45	M365.136T768.138	4.8468	2.277
46	M915.823T68.606	0.20975	-2.2532
47	M429.141T889.195	4.7256	2.2405
48	M951.397T1014.118	4.6992	2.2324
49	M732.379T69.273	0.21508	-2.217
50	M863.373T951.624	4.5455	2.1844

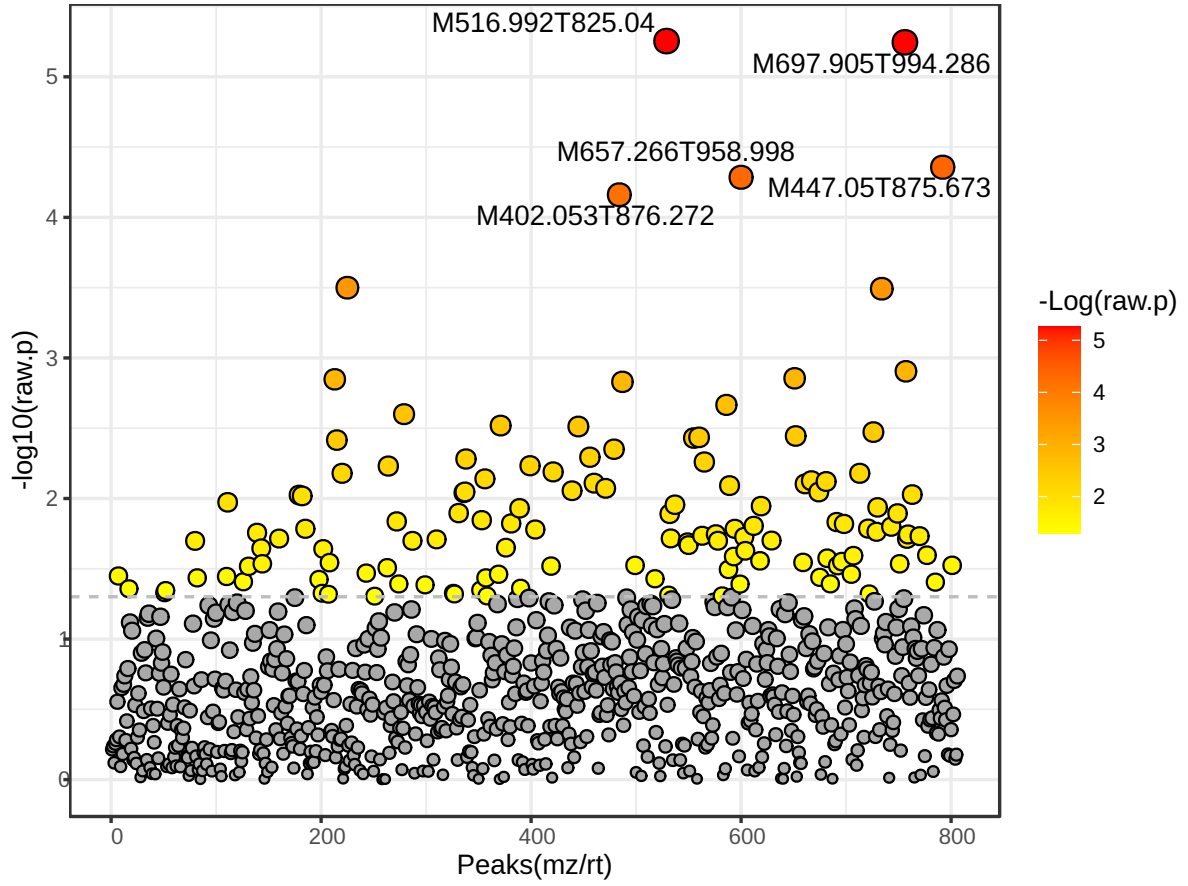


Figure 3: Important features selected by t-tests with threshold 0.05. The red circles represent features above the threshold. Note the p values are transformed by  $-\log_{10}$  so that the more significant features (with smaller p values) will be plotted higher on the graph.



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	0.0022898
2	M697.905T994.286	-32.004	5.682e-06	5.2455	0.0022898
3	M447.05T875.673	-19.133	4.3972e-05	4.3568	0.01045
4	M657.266T958.998	-18.352	5.186e-05	4.2852	0.01045
5	M402.053T876.272	-17.076	6.8987e-05	4.1612	0.011121
6	M204.085T413.719	-11.592	0.00031644	3.4997	0.037057
7	M238.069T801.926	-11.542	0.00032184	3.4924	0.037057
8	M401.05T875.074	-8.13	0.0012451	2.9048	0.10841
9	M463.02T875.074	-7.8893	0.0013959	2.8551	0.10841
10	M407.045T415.442	-7.8547	0.0014194	2.8479	0.10841
11	M475.366T1425.395	-7.7693	0.0014795	2.8299	0.10841
12	M232.059T185.959	-7.0292	0.0021581	2.6659	0.14496
13	M439.083T756.683	-6.7498	0.0025118	2.6	0.15481
14	M365.052T459.24	-6.4185	0.0030285	2.5188	0.15481
15	M684.191T618.708	-6.3919	0.0030753	2.5121	0.15481
16	M317.064T416.942	-6.2361	0.0033689	2.4725	0.15481
17	M289.12T769.482	-6.129	0.0035909	2.4448	0.15481
18	M365.136T768.138	-6.0921	0.0036715	2.4352	0.15481
19	M365.528T823.98	-6.0729	0.0037142	2.4301	0.15481
20	M384.092T841.195	-6.0175	0.0038413	2.4155	0.15481
21	M564.929T1201.416	-5.776	0.0044614	2.3505	0.17123
22	M852.339T915.443	-5.5716	0.0050848	2.2937	0.17968
23	M455.021T825.04	-5.5268	0.0052356	2.281	0.17968
24	M333.041T419.397	-5.4521	0.0054989	2.2597	0.17968
25	M379.067T781.128	-5.3601	0.005846	2.2331	0.17968
26	M398.263T1317.762	-5.3519	0.0058781	2.2308	0.17968
27	M263.13T940.795	-5.2111	0.0064668	2.1893	0.17968
28	M395.28T1470.273	-5.1774	0.006618	2.1793	0.17968
29	M380.071T778.258	-5.1772	0.0066189	2.1792	0.17968
30	M919.334T939.971	-5.047	0.0072465	2.1399	0.17968
31	M429.141T889.195	-5.0043	0.0074676	2.1268	0.17968
32	M373.167T930.605	-4.9851	0.0075699	2.1209	0.17968
33	M721.31T972.44	-4.9472	0.0077769	2.1092	0.17968
34	M341.088T439.204	-4.9305	0.0078698	2.104	0.17968
35	M469.036T876.272	-4.8911	0.0080955	2.0918	0.17968
36	M449.004T832.738	-4.8296	0.0084636	2.0724	0.17968
37	M272.073T418.674	-4.7756	0.008804	2.0553	0.17968
38	M387.034T831.689	-4.7493	0.0089752	2.047	0.17968
39	M203.083T418.775	-4.7468	0.008992	2.0461	0.17968
40	M341.535T617.585	-4.7308	0.0090987	2.041	0.17968
41	M401.049T776.54	-4.6904	0.0093749	2.028	0.17968
42	M433.039T456.919	-4.6802	0.0094462	2.0247	0.17968
43	M1134.038T907.684	4.6605	0.0095858	2.0184	0.17968
44	M1095.063T894.34	4.522	0.010642	1.973	0.19494
45	M531.009T876.272	-4.4698	0.011076	1.9556	0.19714
46	M409.002T109.878	-4.4406	0.011329	1.9458	0.19714
47	M824.341T916.725	-4.4138	0.011567	1.9368	0.19714
48	M445.037T424.564	-4.3946	0.01174	1.9303	0.19714
49	M1094.946T896.052	4.2948	0.012697	1.8963	0.20326
50	M319.076T917.392	-4.2882	0.012765	1.894	0.20326

## 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 4 is pairwise score plots providing an overview of the various separation patterns among the most significant PCs; Figure 5 is the scree plot showing the variances explained by the selected PCs; Figure 6 shows the 2-D scores plot between selected PCs; Figure 7 shows the biplot between the selected PCs. Interactive 3-D scores plots are not included here and can be directly downloaded from website.

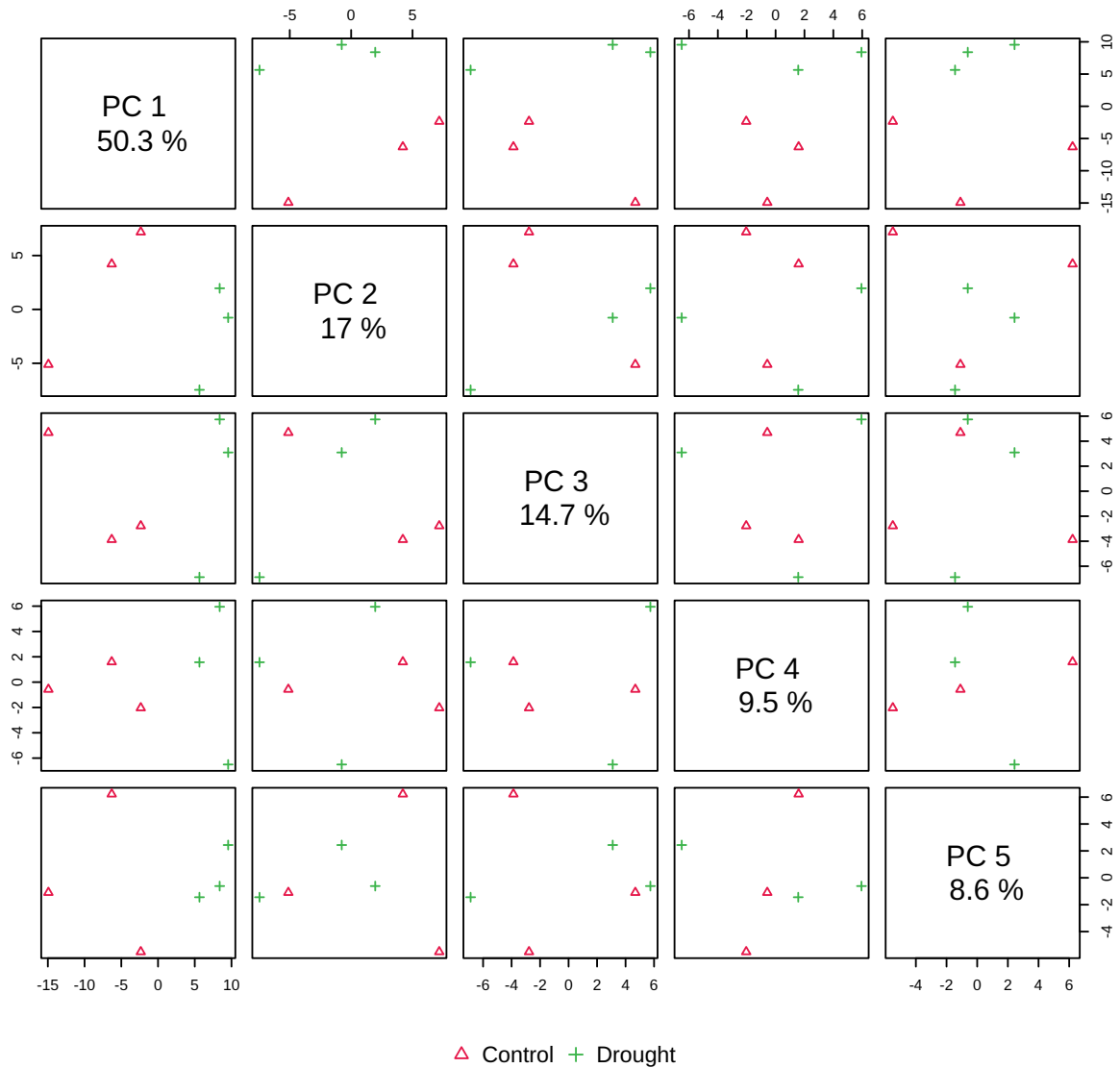


Figure 4: Pairwise score plots between the selected PCs. The explained variance of each PC is shown in the corresponding diagonal cell.

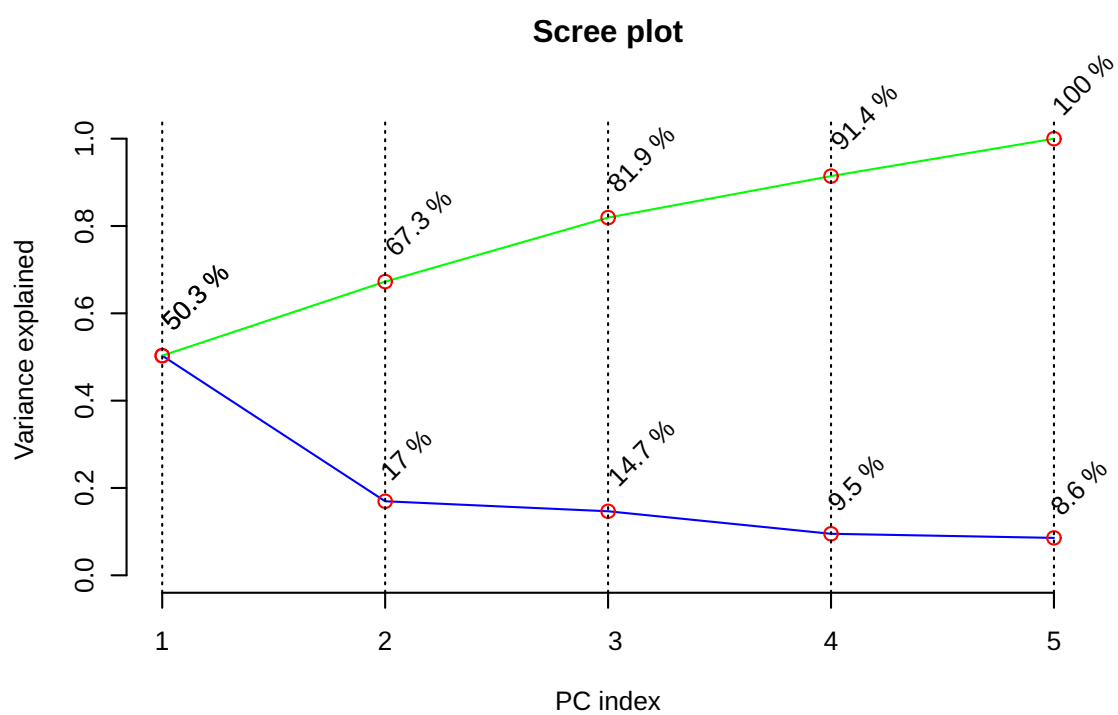


Figure 5: Scree plot shows the variance explained by PCs. The green line on top shows the accumulated variance explained; the blue line underneath shows the variance explained by individual PC.

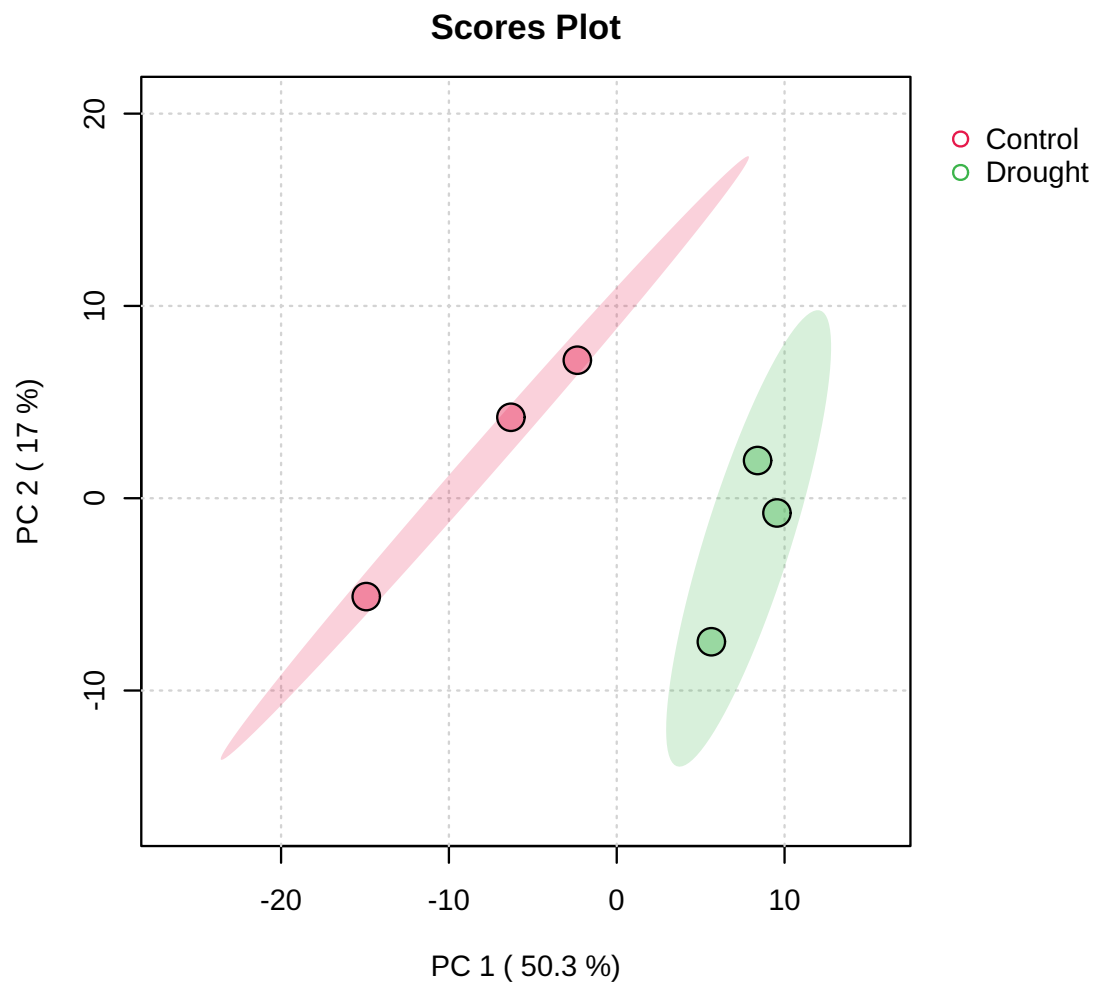


Figure 6: Scores plot between the selected PCs. The explained variances are shown in brackets.



## 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 `pls` function provided by R `pls` package<sup>4</sup>. The classification and cross-validation are performed using the corresponding wrapper function offered by the `caret` package<sup>5</sup>.

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.<sup>6</sup>

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 8 shows the overview of scores plots; Figure 9 shows the 2-D scores plot between selected components; Figure 10 shows the 3-D scores plot between selected components; Figure 11 shows the loading plot between the selected components; Figure 12 shows the classification performance with different number of components; Figure 13 shows the results of permutation test for model validation; Figure 14 shows important features identified by PLS-DA.

---

<sup>4</sup>Ron Wehrens and Bjorn-Helge Mevik. *pls: Partial Least Squares Regression (PLSR) and Principal Component Regression (PCR)*, 2007, R package version 2.1-0

<sup>5</sup>Max Kuhn. Contributions from Jed Wing and Steve Weston and Andre Williams. *caret: Classification and Regression Training*, 2008, R package version 3.45

<sup>6</sup>Bijlsma et al. *Large-Scale Human Metabolomics Studies: A Strategy for Data (Pre-) Processing and Validation*, Anal Chem. 2006, 78 567 - 574

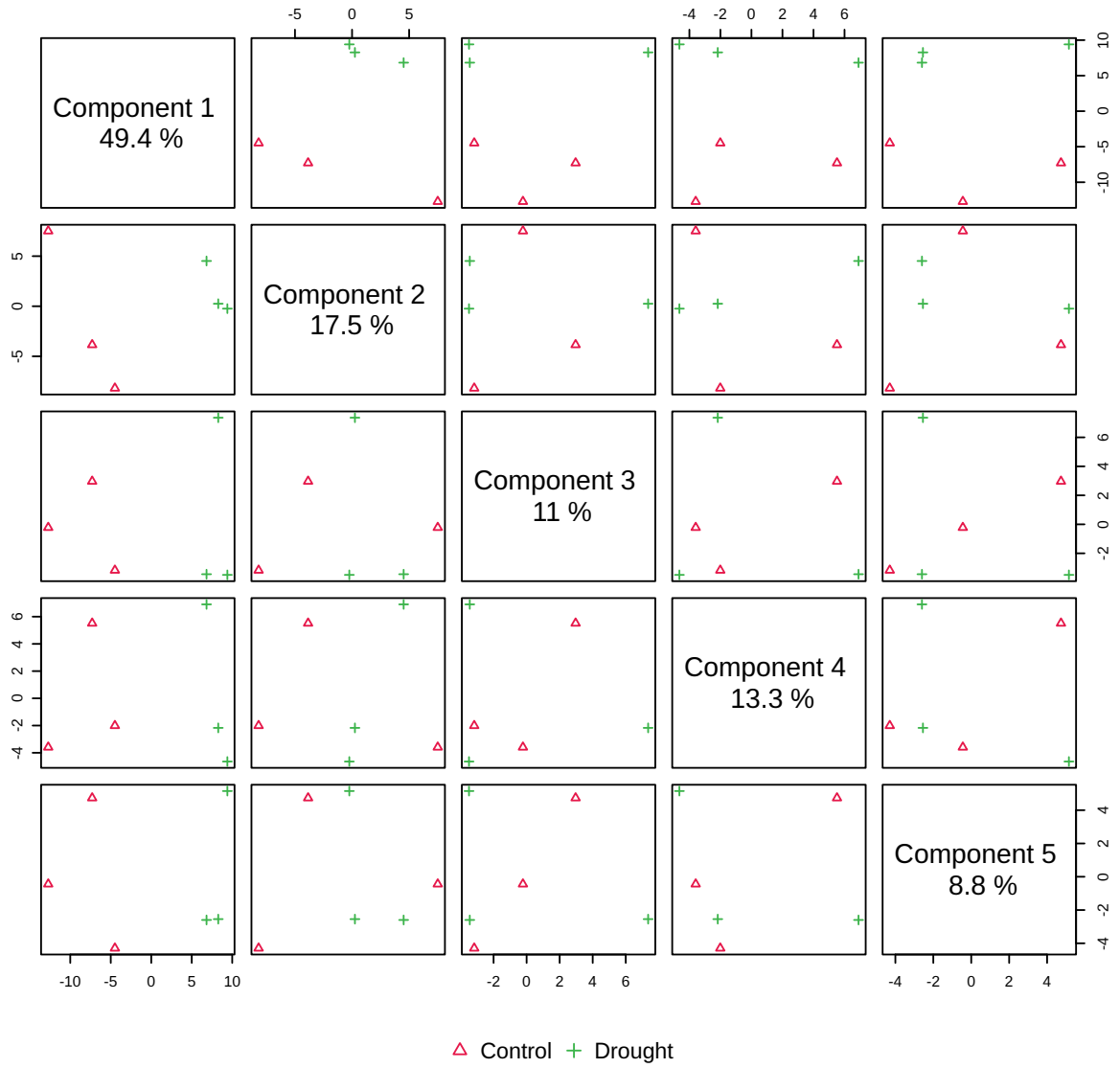


Figure 8: Pairwise scores plots between the selected components. The explained variance of each component is shown in the corresponding diagonal cell.

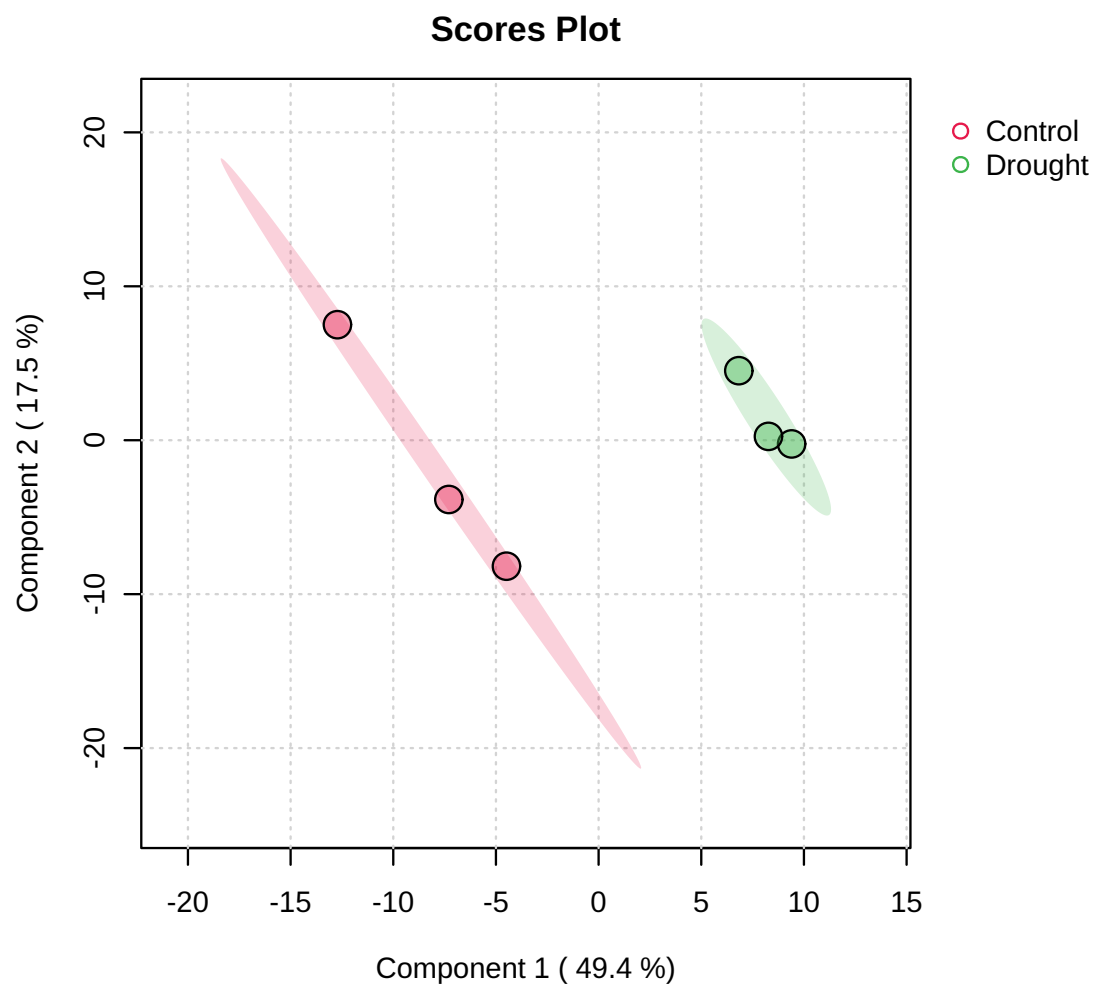


Figure 9: Scores plot between the selected PCs. The explained variances are shown in brackets.



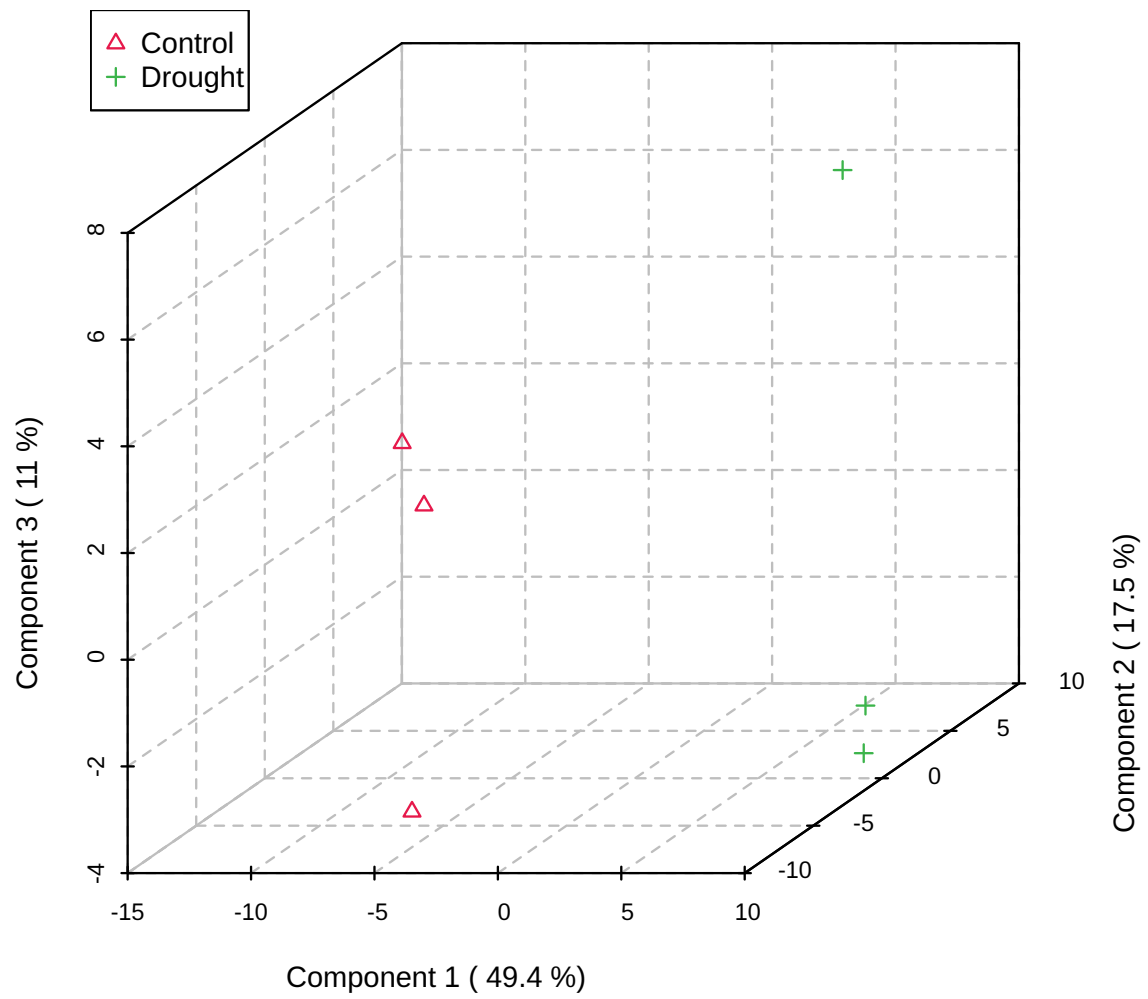


Figure 10: 3D scores plot between the selected PCs. The explained variances are shown in brackets.

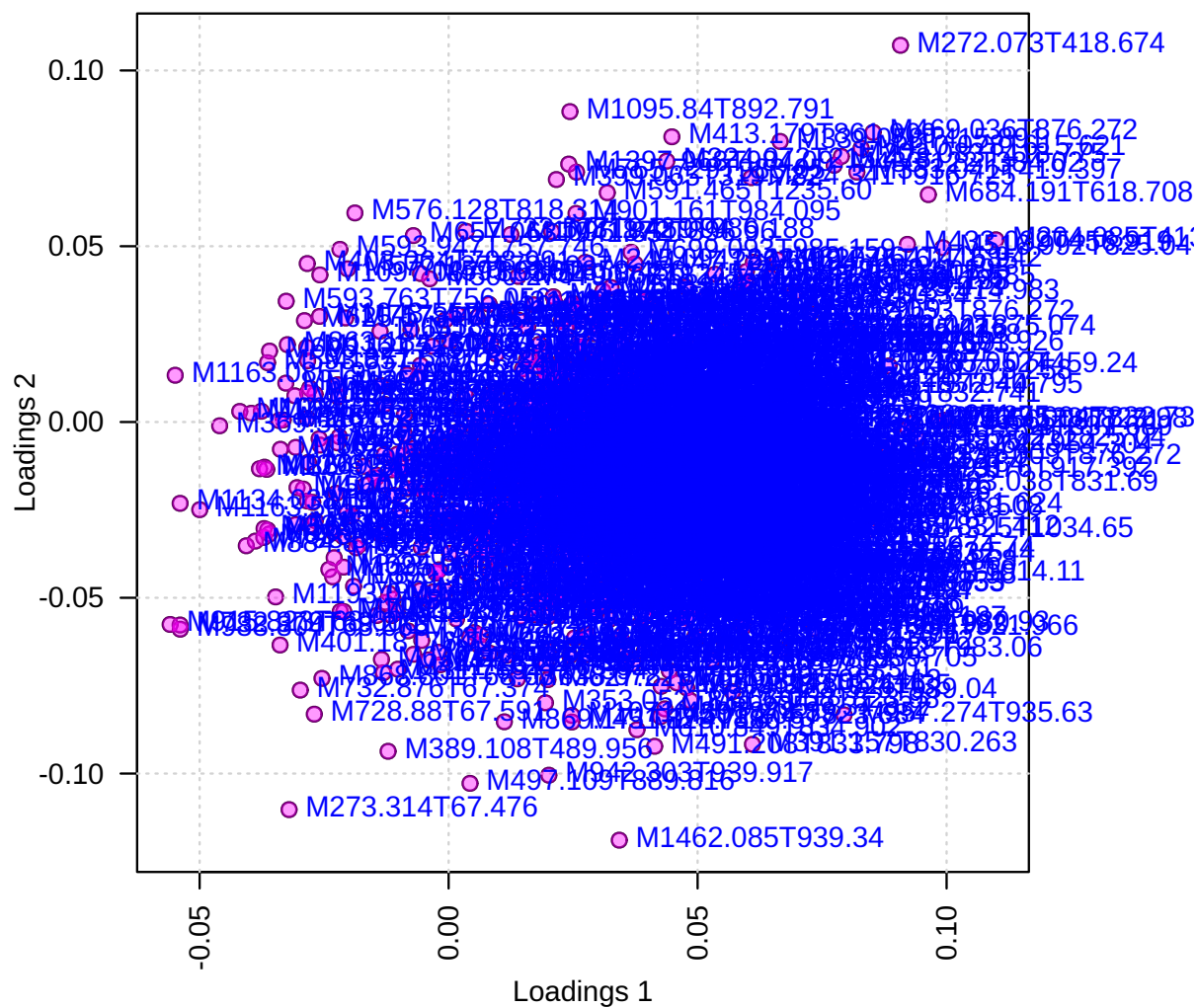


Figure 11: Loadings plot between the selected PCs.

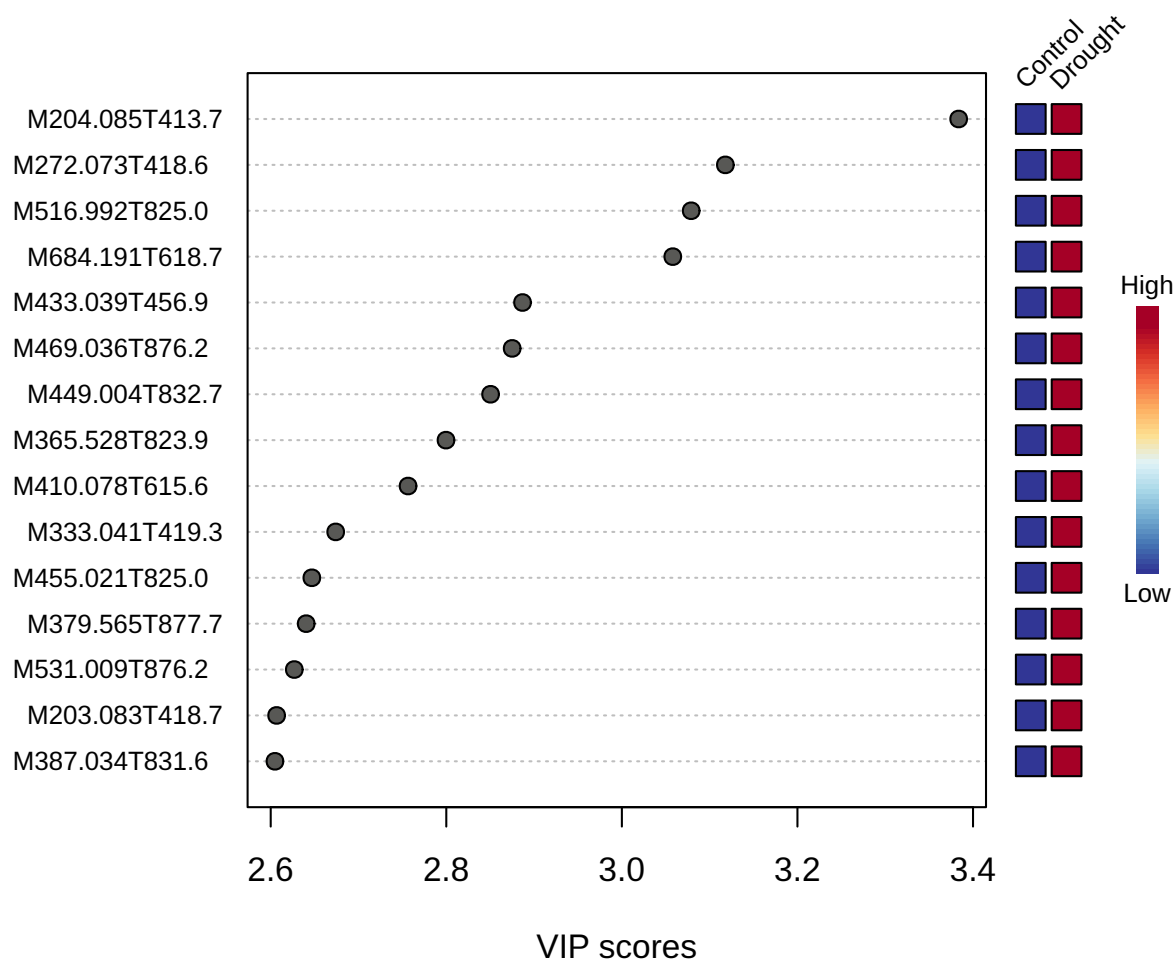


Figure 12: Important features identified by PLS-DA. The colored boxes on the right indicate the relative concentrations of the corresponding metabolite in each group under study.

## 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.

Hierarchical clustering is performed with the `hclust` function in package `stat`. Figure 15 shows the clustering result in the form of a dendrogram. Figure 16 shows the clustering result in the form of a heatmap.

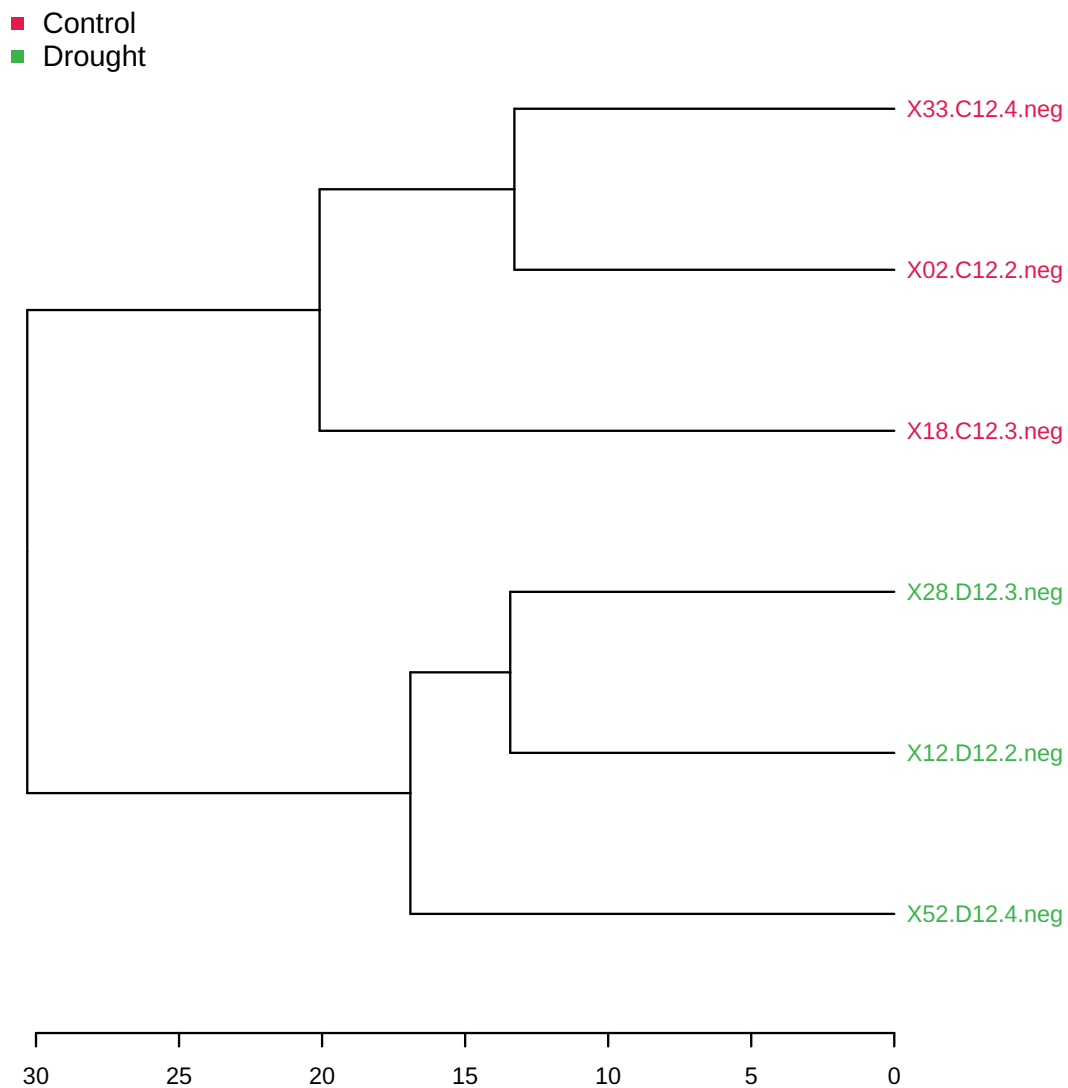


Figure 13: Clustering result shown as dendrogram (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

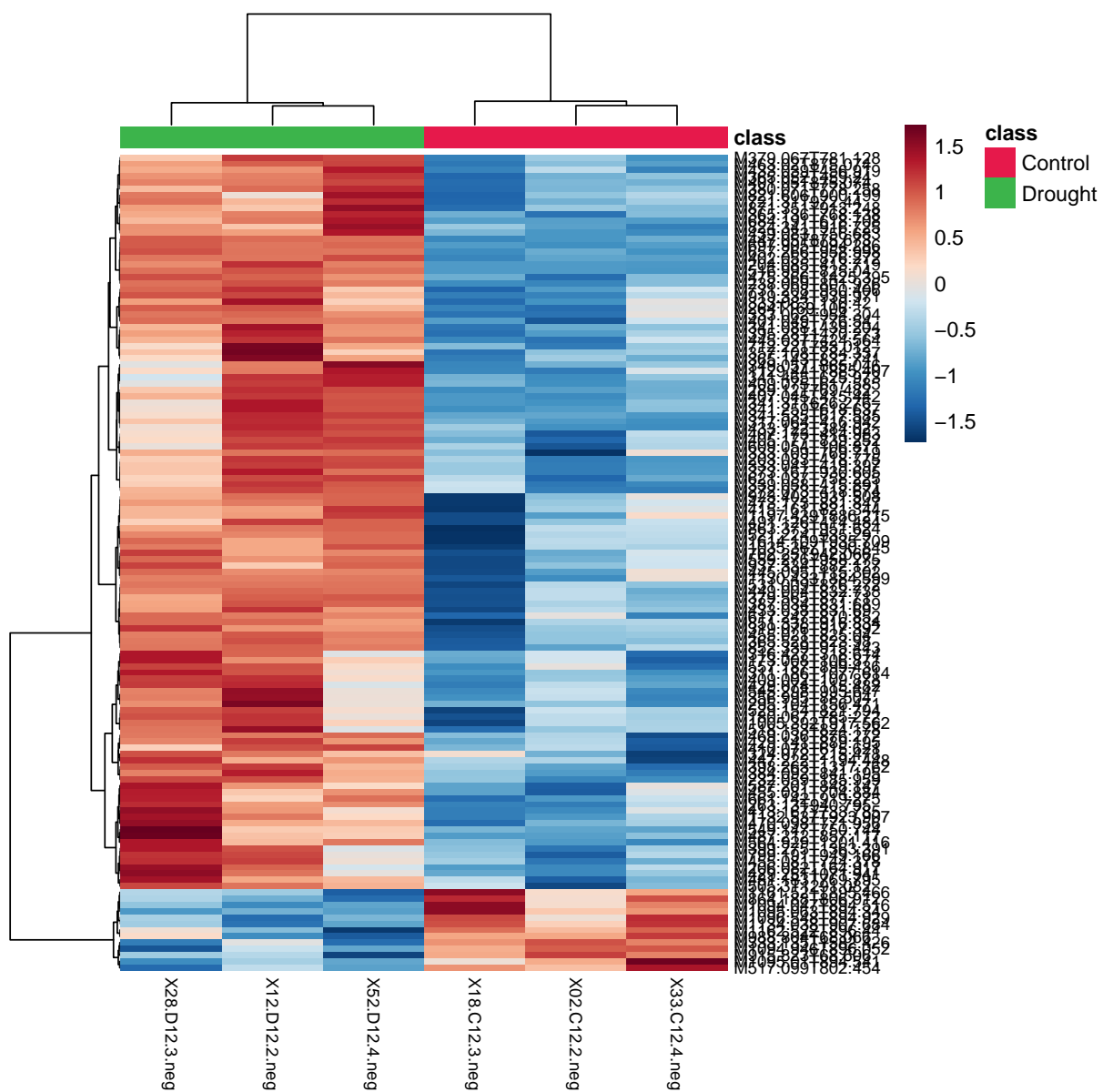


Figure 14: Clustering result shown as heatmap (distance measure using euclidean, and clustering algorithm using ward.D).

### 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] "smp1.nm.vec <- c(\"X44.D12.1.neg\", \"X09.X12.3.neg\")"
[11] "grp.nm.vec <- c(\"Control\", \"Drought\", \"Fusarium\", \"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] "feature.nm.vec <- c(\"\")"
[18] "smp1.nm.vec <- c(\"X44.D12.1.neg\", \"X09.X12.3.neg\")"
[19] "grp.nm.vec <- c(\"Control\", \"Drought\")"
[20] "mSet<-UpdateData(mSet, T)"
[21] "mSet<-PreparePrenormData(mSet)"
[22] "mSet<-Normalization(mSet, \"CompNorm\", \"LogNorm\", \"ParetoNorm\", \"sodium_formate\", ratio
[23] "mSet<-PlotNormSummary(mSet, \"norm_1\", \"png\", 72, width=NA)"
[24] "mSet<-PlotSampleNormSummary(mSet, \"snorm_1\", \"png\", 72, width=NA)"
[25] "mSet<-FC.Anal(mSet, 2.0, 0, FALSE)"
[26] "mSet<-PlotFC(mSet, \"fc_0\", \"png\", 72, width=NA)"
[27] "mSet<-FC.Anal(mSet, 2.0, 1, FALSE)"
[28] "mSet<-PlotFC(mSet, \"fc_1\", \"png\", 72, width=NA)"
[29] "mSet<-FC.Anal(mSet, 2.0, 0, FALSE)"
[30] "mSet<-PlotFC(mSet, \"fc_2\", \"png\", 72, width=NA)"
[31] "mSet<-FC.Anal(mSet, 2.0, 1, FALSE)"
[32] "mSet<-PlotFC(mSet, \"fc_3\", \"png\", 72, width=NA)"
[33] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, TRUE, \"fdr\", FALSE)"
[34] "mSet<-PlotTT(mSet, \"tt_0\", \"png\", 72, width=NA)"
[35] "mSet<-Ttests.Anal(mSet, F, 0.05, FALSE, TRUE, \"raw\", FALSE)"
[36] "mSet<-PlotTT(mSet, \"tt_1\", \"png\", 72, width=NA)"
[37] "mSet<-PCA.Anal(mSet)"
[38] "mSet<-PlotPCAPairSummary(mSet, \"pca_pair_0\", \"png\", 72, width=NA, 5)"
[39] "mSet<-PlotPCAScree(mSet, \"pca_scee_0\", \"png\", 72, width=NA, 5)"
[40] "mSet<-PlotPCA2DScore(mSet, \"pca_score2d_0\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[41] "mSet<-PlotPCALoading(mSet, \"pca_loading_0\", \"png\", 72, width=NA, 1,2);"
[42] "mSet<-PlotPCABiplot(mSet, \"pca_biplot_0\", \"png\", 72, width=NA, 1,2)"
[43] "mSet<-PlotPCA3DLoading(mSet, \"pca_loading3d_0\", \"json\", 1,2,3)"
[44] "mSet<-PLSR.Anal(mSet, reg=TRUE)"
[45] "mSet<-PlotPLSPairSummary(mSet, \"pls_pair_0\", \"png\", 72, width=NA, 5)"
[46] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_0\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[47] "mSet<-PlotPLS3DScoreImg(mSet, \"pls_score3d_0\", \"png\", 72, width=NA, 1,2,3, 40)"
[48] "mSet<-PlotPLSLoading(mSet, \"pls_loading_0\", \"png\", 72, width=NA, 1, 2);"
[49] "mSet<-PlotPLS3DLoading(mSet, \"pls_loading3d_0\", \"json\", 1,2,3)"
[50] "mSet<-PlotPLS.Imp(mSet, \"pls_imp_0\", \"png\", 72, width=NA, \"vip\", \"Comp. 1\", 15,FALSE)"
[51] "mSet<-PlotHCTree(mSet, \"tree_0\", \"png\", 72, width=NA, \"euclidean\", \"ward.D\")"
[52] "mSet<-PlotHeatMap(mSet, \"heatmap_1\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\"
[53] "mSet<-GetGroupNames(mSet, \"null\")"
[54] "mSet<-PlotStaticHeatMap(mSet, \"heatmap_1\", \"pdf\", 72, width=NA, \"norm\", \"row\", \"eucl
[55] "mSet<-PlotSubHeatMap(mSet, \"heatmap_2\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclide
[56] "mSet<-PlotSubHeatMap(mSet, \"heatmap_2\", \"pdf\", 72, width=NA, \"norm\", \"row\", \"euclide
```

```
[57] "mSet<-SaveTransformedData(mSet)"
[58] "mSet<-PreparePDFReport(mSet, \"guest16724914738269757780\")\n"
```

---

The report was generated on Fri Mar 15 06:24:50 2024 with R version 4.3.2 (2023-10-31), OS system: Linux, version: -Ubuntu SMP Tue Jan 9 15:25:40 UTC 2024 .