Metabolomic Data Analysis with MetaboAnalyst 6.0

Name: guest12705925767388953312

March 4, 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 50 (samples) by 658 (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)
C9.1	658	0	658
C9.2	657	1	658
C9.3	657	1	658
C9.4	658	0	658
D9.1	658	0	658
D9.2	658	0	658
D9.3	658	0	658
D9.4	658	0	658
F9.1	658	0	658
F9.2	658	0	658
F9.3	658	0	658
F9.4	635	23	658
X9.1	657	1	658
X9.2	658	0	658
X9.3	658	0	658
X9.4	658	0	658
BLANK.2	625	33	658
QC.1	658	0	658
QC.2	658	0	658
QC.3	658	0	658
C12.1	645	13	658
C12.3	658	0	658
C12.4	658	0	658
D12.1	658	0	658
D12.2	658	0	658
D12.3	658	0	658
D12.4	658	0	658
F12.1	658	0	658
F12.2	658	0	658
F12.3	635	23	658
F12.4	656	2	658
X12.1	658	0	658
X12.2	658	0	658
X12.3	658	0	658
C15.1	658	0	658
C15.2	657	1	658
C15.3	658	0	658
C15.4	658	0	658
D15.1	624	34	658
D15.2	658	0	658
D15.3	635	23	658
D15.4	644	14	658
F15.1	636	22	658
F15.2	658	0	658
F15.3	650	8	658
F15.4	658	0	658
X15.1	658	0	658
X15.2	658	0	658
X15.3	658	0	658
X15.4	658	0	658

[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

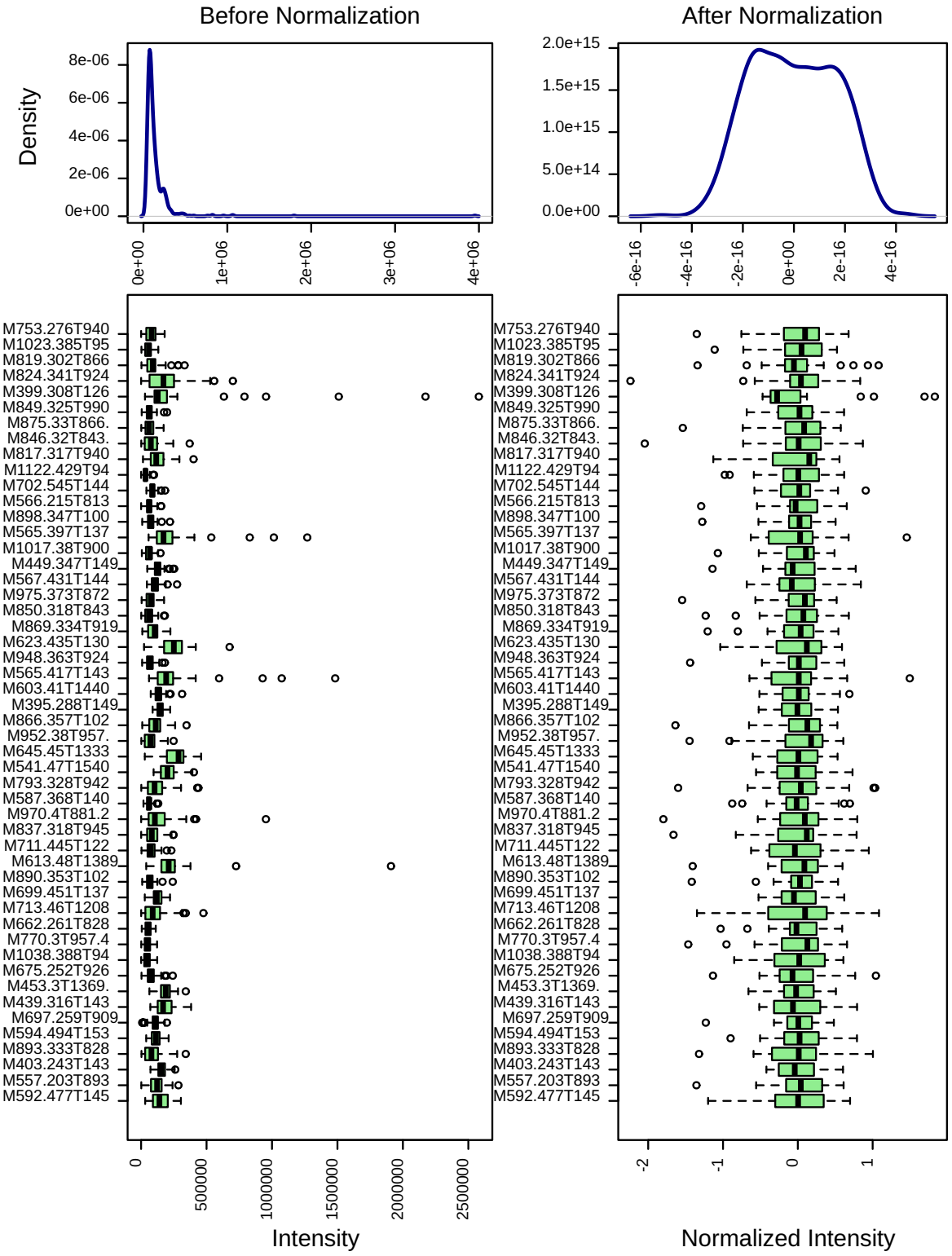

Figure 1: Box plots and kernel density plots before and after normalization. The boxplots show at most 50 features due to space limit. The density plots are based on all samples.

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 One-way ANOVA

Univariate analysis methods are the most common methods used for exploratory data analysis. For multi-group analysis, MetaboAnalyst provides one-way Analysis of Variance (ANOVA). As ANOVA only tells whether the overall comparison is significant or not, it is usually followed by post-hoc analyses in order to identify which two levels are different. MetaboAnalyst provides two most commonly used methods for this purpose - Fisher's least significant difference method (Fisher's LSD) and Tukey's Honestly Significant Difference (Tukey's HSD). The univariate analyses provide a preliminary overview about features that are potentially significant in discriminating the conditions under study.

Figure 2 shows the important features identified by ANOVA analysis. Table 2 shows the details of these features. The `post-hoc Sig. Comparison` column shows the comparisons between different levels that are significant given the p value threshold.

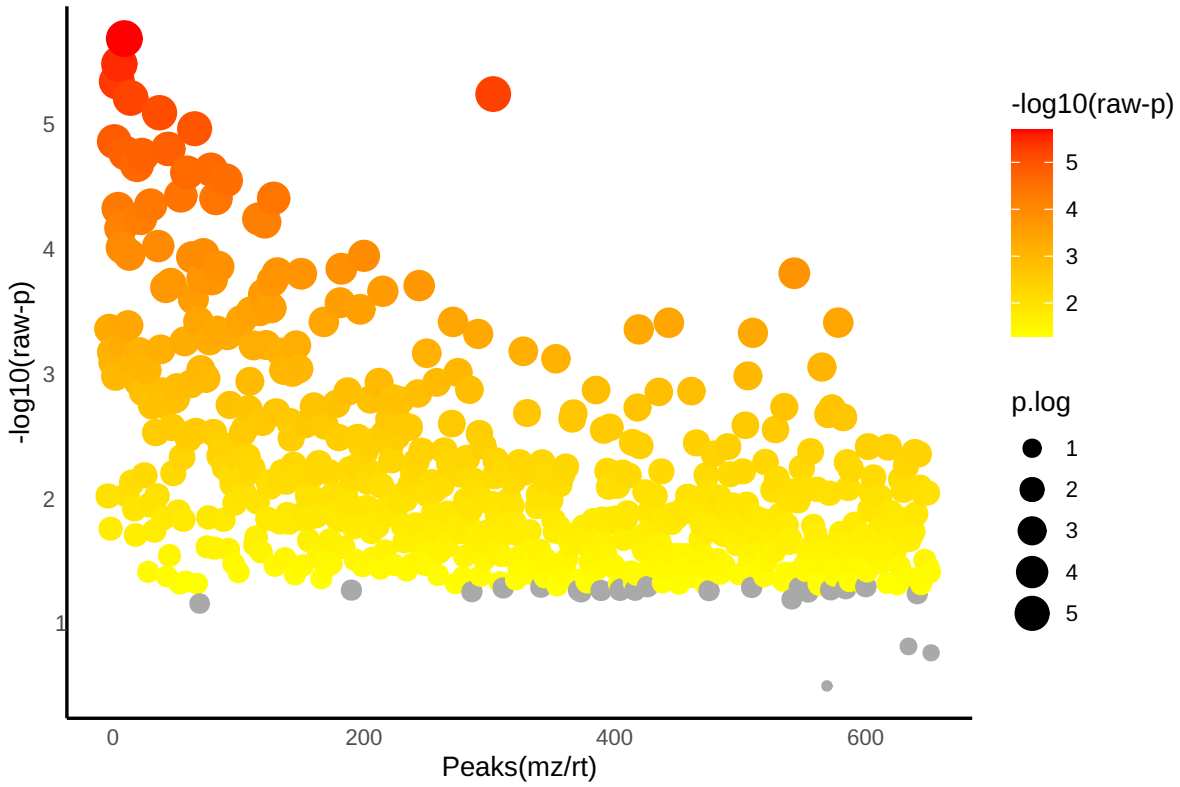

Figure 2: Important features selected by ANOVA plot with p value threshold 0.051556.

Table 2: Top 50 features identified by One-way ANOVA and post-hoc analysis

	Peaks(mz/rt)	F.stat	p.value	-log10(p)	FDR
1	M681.474T1424.253	14.521	2.0487e-06	5.6885	0.00080494
2	M681.469T1389.183	13.658	3.2654e-06	5.4861	0.00080494
3	M681.469T1408.485	13.083	4.5129e-06	5.3455	0.00080494
4	M449.365T1540.699	12.68	5.6985e-06	5.2442	0.00080494
5	M637.411T1406.122	12.557	6.1259e-06	5.2128	0.00080494
6	M695.45T1338.973	12.104	8.0359e-06	5.095	0.00087993
7	M679.553T1406.451	11.626	1.0796e-05	4.9667	0.0010133
8	M679.436T1338.726	11.254	1.3668e-05	4.8643	0.001048
9	M637.415T1439.736	11.043	1.5666e-05	4.805	0.001048
10	M679.452T1355.733	10.921	1.6962e-05	4.7705	0.001048
11	M637.412T1422.711	10.87	1.7546e-05	4.7558	0.001048
12	M682.472T1373.006	10.595	2.1062e-05	4.6765	0.0011362
13	M660.517T1313.696	10.468	2.2954e-05	4.6391	0.0011362
14	M649.479T1423.325	10.389	2.4212e-05	4.616	0.0011362
15	M555.454T1439.586	10.177	2.8004e-05	4.5528	0.0012266
16	M648.466T1423.012	9.7717	3.7222e-05	4.4292	0.0014267
17	M624.432T1423.013	9.7073	3.8973e-05	4.4092	0.0014267
18	M650.482T1423.013	9.7032	3.9089e-05	4.4079	0.0014267
19	M670.528T1488.279	9.5344	4.4139e-05	4.3552	0.0015263
20	M695.451T1225.817	9.4465	4.7052e-05	4.3274	0.0015457
21	M646.455T1423.013	9.1945	5.6648e-05	4.2468	0.0017003
22	M660.517T1423.005	9.1877	5.6936e-05	4.2446	0.0017003
23	M555.447T1456.772	9.1064	6.0493e-05	4.2183	0.001728
24	M680.474T1439.81	8.9579	6.7652e-05	4.1697	0.001852
25	M661.446T1372.452	8.5292	9.4105e-05	4.0264	0.0024435
26	M574.467T1456.399	8.4945	9.6698e-05	4.0146	0.0024435
27	M668.509T1456.373	8.3404	0.00010922	3.9617	0.0025294
28	M637.414T1384.421	8.3269	0.0001104	3.957	0.0025294
29	M655.418T1389.36	8.3024	0.00011258	3.9485	0.0025294
30	M646.461T1352.771	8.2704	0.0001155	3.9374	0.0025294
31	M646.454T1406.122	8.0523	0.00013772	3.861	0.0029188
32	M655.416T1372.761	8.0074	0.00014285	3.8451	0.0029329
33	M623.428T1439.576	7.9077	0.00015503	3.8096	0.002943
34	M481.362T1439.996	7.9051	0.00015536	3.8087	0.002943
35	M660.514T1439.78	7.8941	0.00015678	3.8047	0.002943
36	M668.506T1440.08	7.7947	0.00017024	3.7689	0.0030989
37	M680.472T1423.013	7.7649	0.00017452	3.7582	0.0030989
38	M555.451T1436.867	7.7323	0.00017933	3.7464	0.0031005
39	M646.453T1439.754	7.6761	0.00018798	3.7259	0.0031667
40	M664.462T1368.515	7.6322	0.00019505	3.7099	0.0032037
41	M590.465T1389.456	7.592	0.00020179	3.6951	0.0032336
42	M549.424T1386.472	7.5082	0.00021667	3.6642	0.0033894
43	M670.538T1472.8	7.4476	0.00022819	3.6417	0.0034866
44	M555.456T1423.317	7.3431	0.00024967	3.6026	0.003728
45	M623.429T1456.316	7.2633	0.00026758	3.5725	0.0039067
46	M650.482T1439.948	7.1543	0.00029435	3.5311	0.004204
47	M624.436T1440.08	7.1241	0.00030229	3.5196	0.0042256
48	M639.424T1423.012	7.0927	0.00031079	3.5075	0.0042318
49	M603.415T1456.538	7.0753	0.00031561	3.5008	0.0042318
50	M639.422T1384.421	6.8958	0.00037046	3.4313	0.004626

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

PLS is a supervised method that uses multivariate regression techniques to extract via linear combination of original variables (X) the information that can predict the class membership (Y). The PLS regression is performed using the `plsr` function provided by R `pls` package[4]. The classification and cross-validation are performed using the corresponding wrapper function offered by the `caret` package[5].

To assess the significance of class discrimination, a permutation test was performed. In each permutation, a PLS-DA model was built between the data (X) and the permuted class labels (Y) using the optimal number of components determined by cross validation for the model based on the original class assignment. MetaboAnalyst supports two types of test statistics for measuring the class discrimination. The first one is based on prediction accuracy during training. The second one is separation distance based on the ratio of the between group sum of the squares and the within group sum of squares (B/W-ratio). If the observed test statistic is part of the distribution based on the permuted class assignments, the class discrimination cannot be considered significant from a statistical point of view.[6].

There are two variable importance measures in PLS-DA. The first, Variable Importance in Projection (VIP) is a weighted sum of squares of the PLS loadings taking into account the amount of explained Y-variation in each dimension. Please note, VIP scores are calculated for each components. When more than components are used to calculate the feature importance, the average of the VIP scores are used. The other importance measure is based on the weighted sum of PLS-regression. The weights are a function of the reduction of the sums of squares across the number of PLS components. Please note, for multiple-group (more than two) analysis, the same number of predictors will be built for each group. Therefore, the coefficient of each feature will be different depending on which group you want to predict. The average of the feature coefficients are used to indicate the overall coefficient-based importance.

Figure 3 shows the overview of scores plots; Figure 4 shows the 2-D scores plot between selected components; Figure 5 shows the 3-D scores plot between selected components; Figure 6 shows the loading plot between the selected components;Figure 7 shows the classification performance with different number of components; Figure 8 shows the results of permutation test for model validation; Figure 9 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

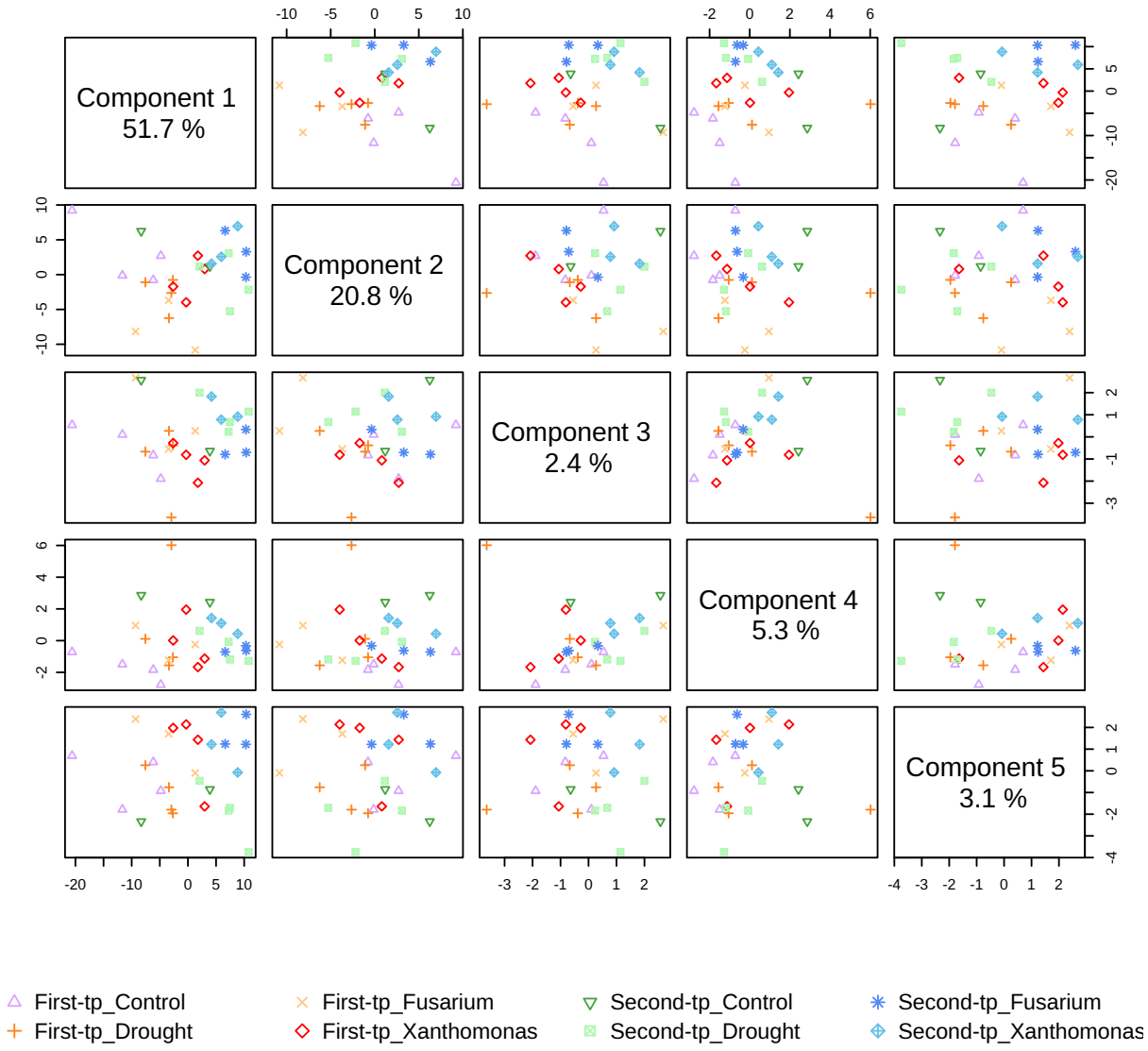

Figure 3: Pairwise scores plots between the selected components. The explained variance of each component is shown in the corresponding diagonal cell.

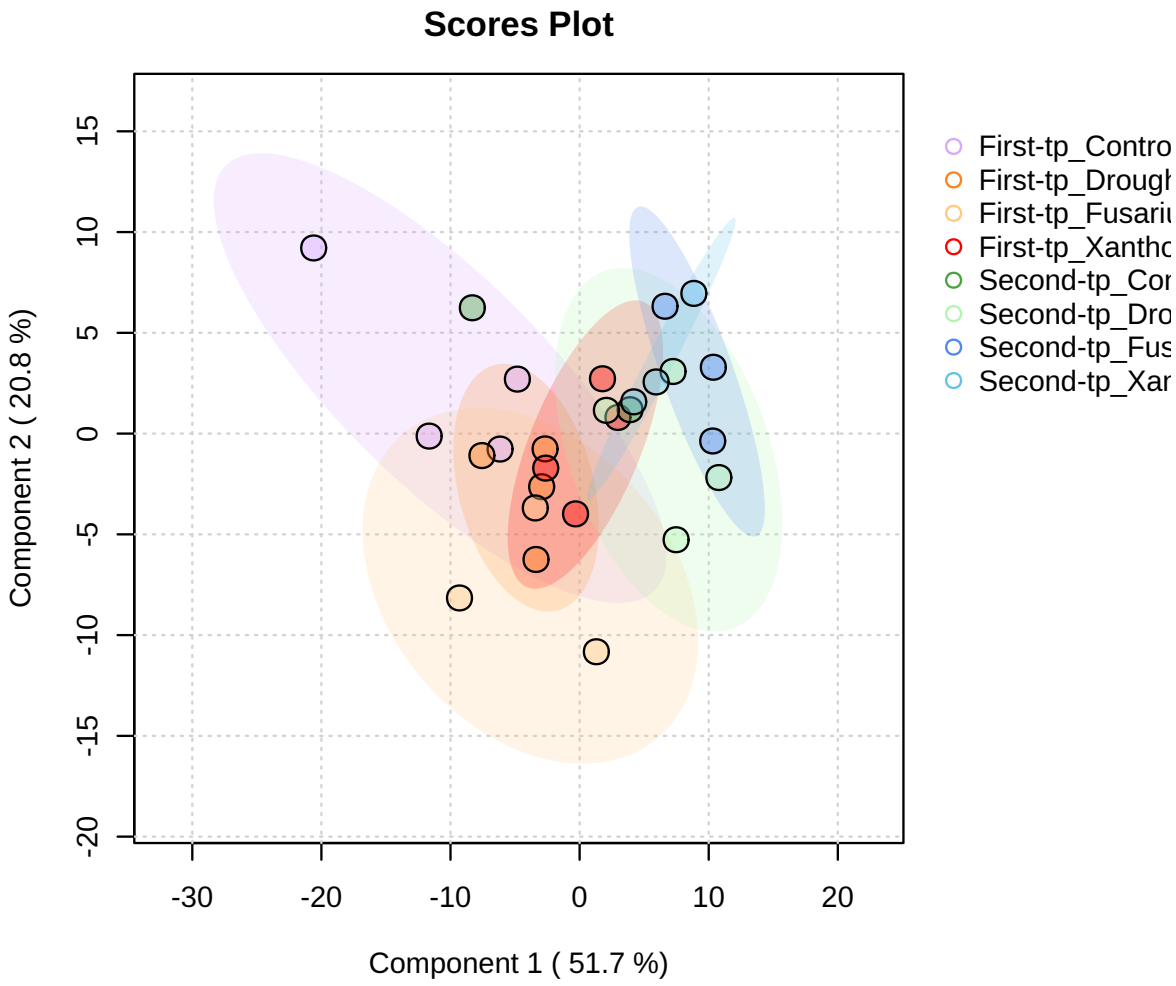

Figure 4: Scores plot between the selected PCs. The explained variances are shown in brackets.

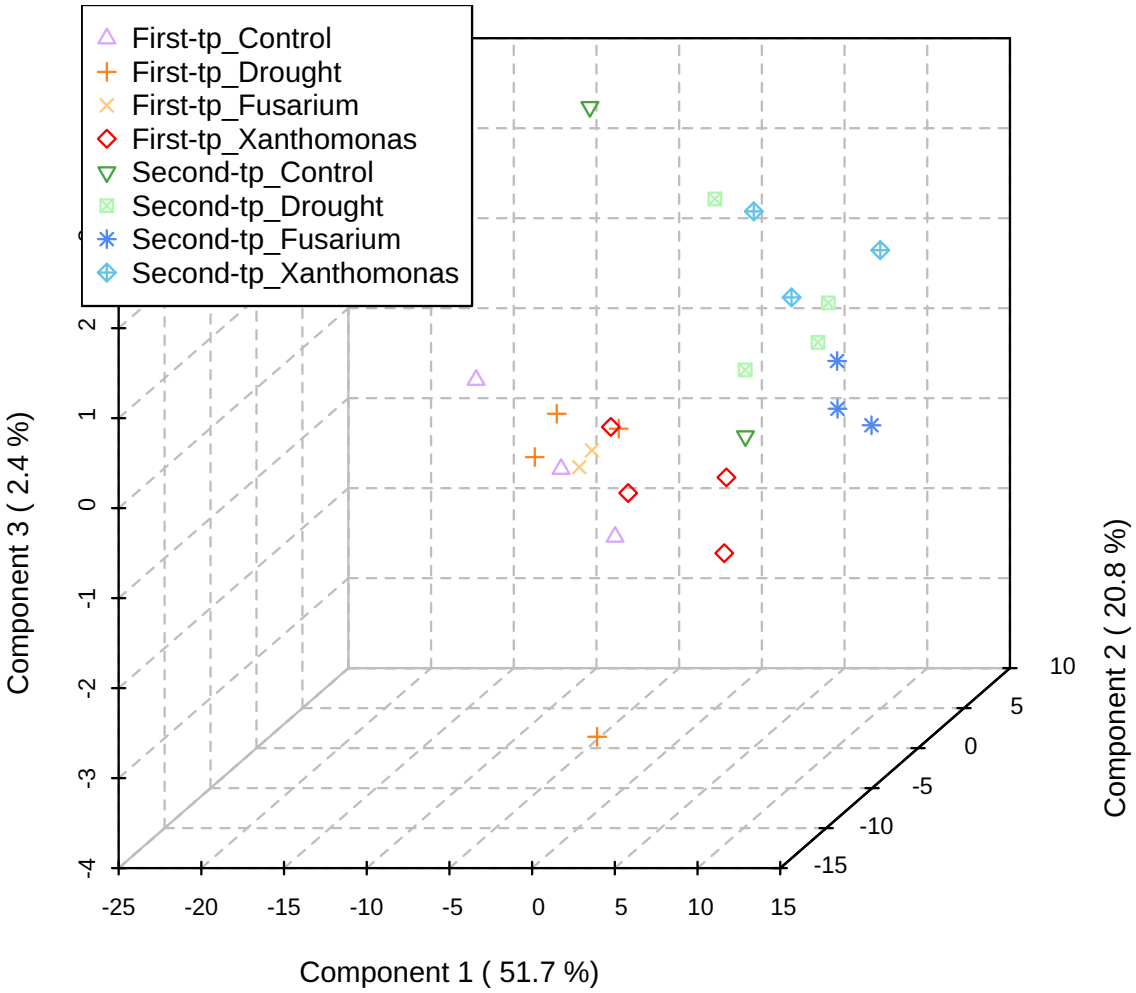

Figure 5: 3D scores plot between the selected PCs. The explained variances are shown in brackets.

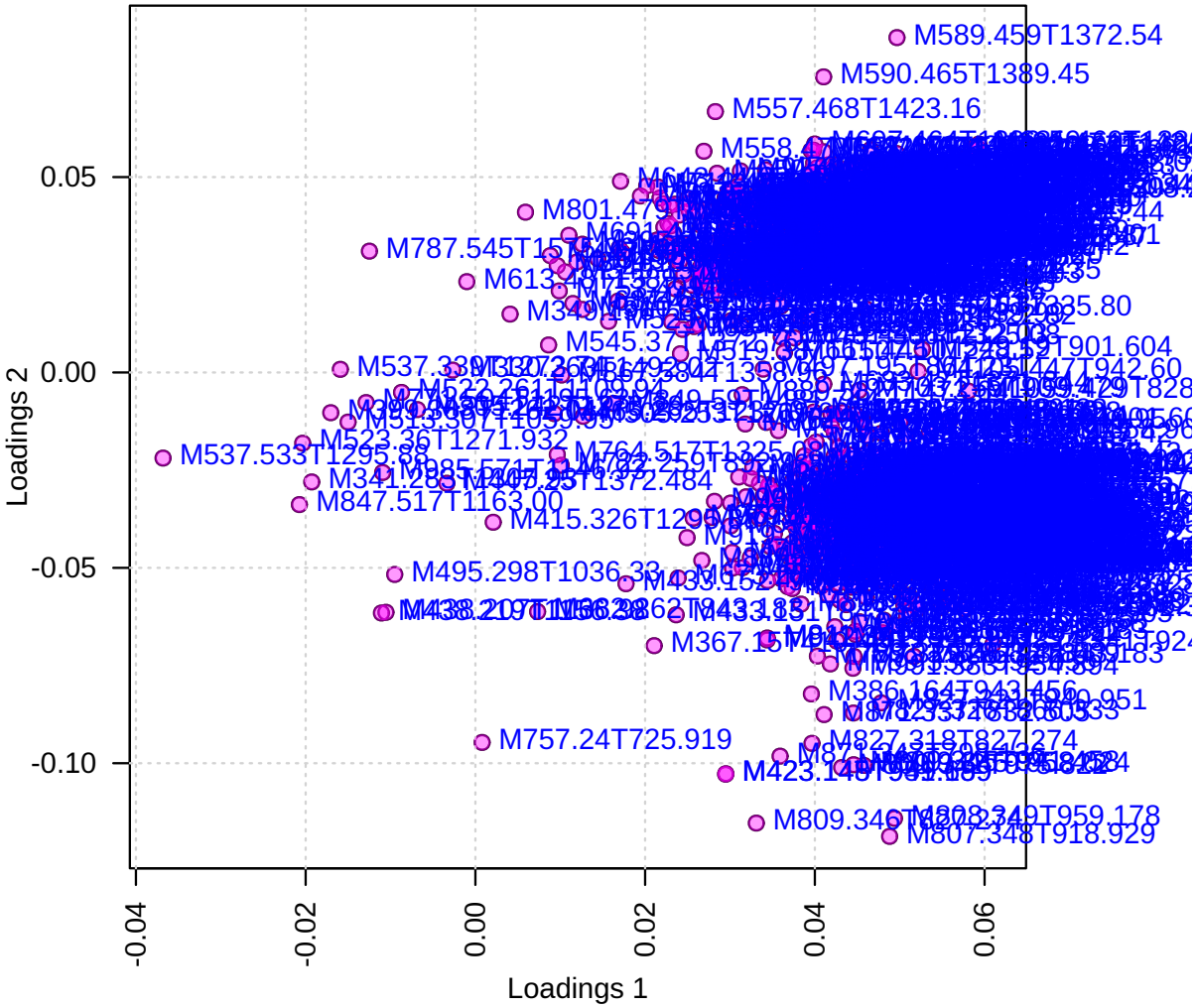

Figure 6: Loadings plot between the selected PCs.

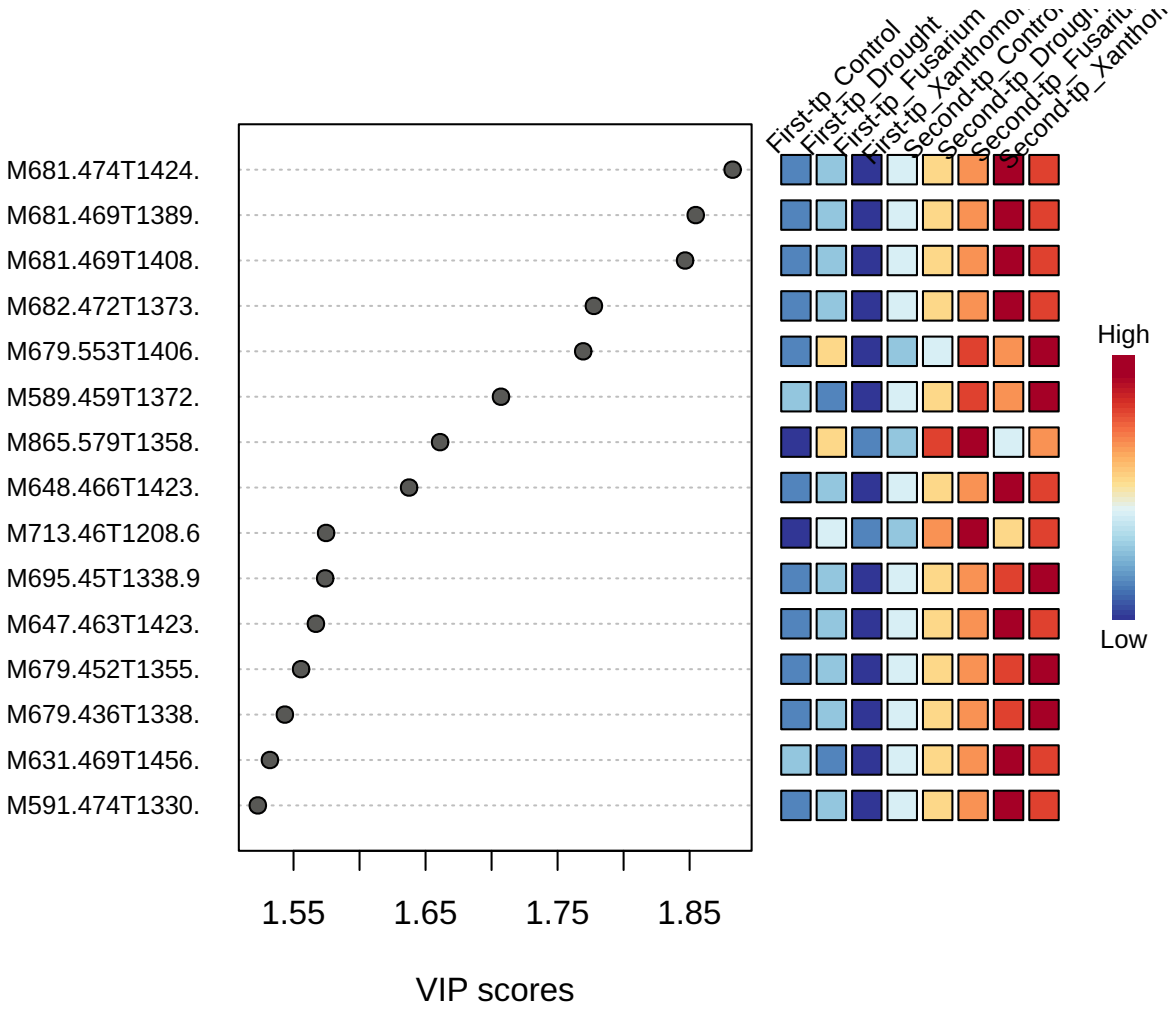

Figure 7: Important features identified by PLS-DA. The colored boxes on the right indicate the relative concentrations of the corresponding metabolite in each group under study.

2.3 Hierarchical Clustering

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

Hierachical clustering is performed with the `hclust` function in package `stat`. Figure 10 shows the clustering result in the form of a dendrogram. Figure 11 shows the clustering result in the form of a heatmap.

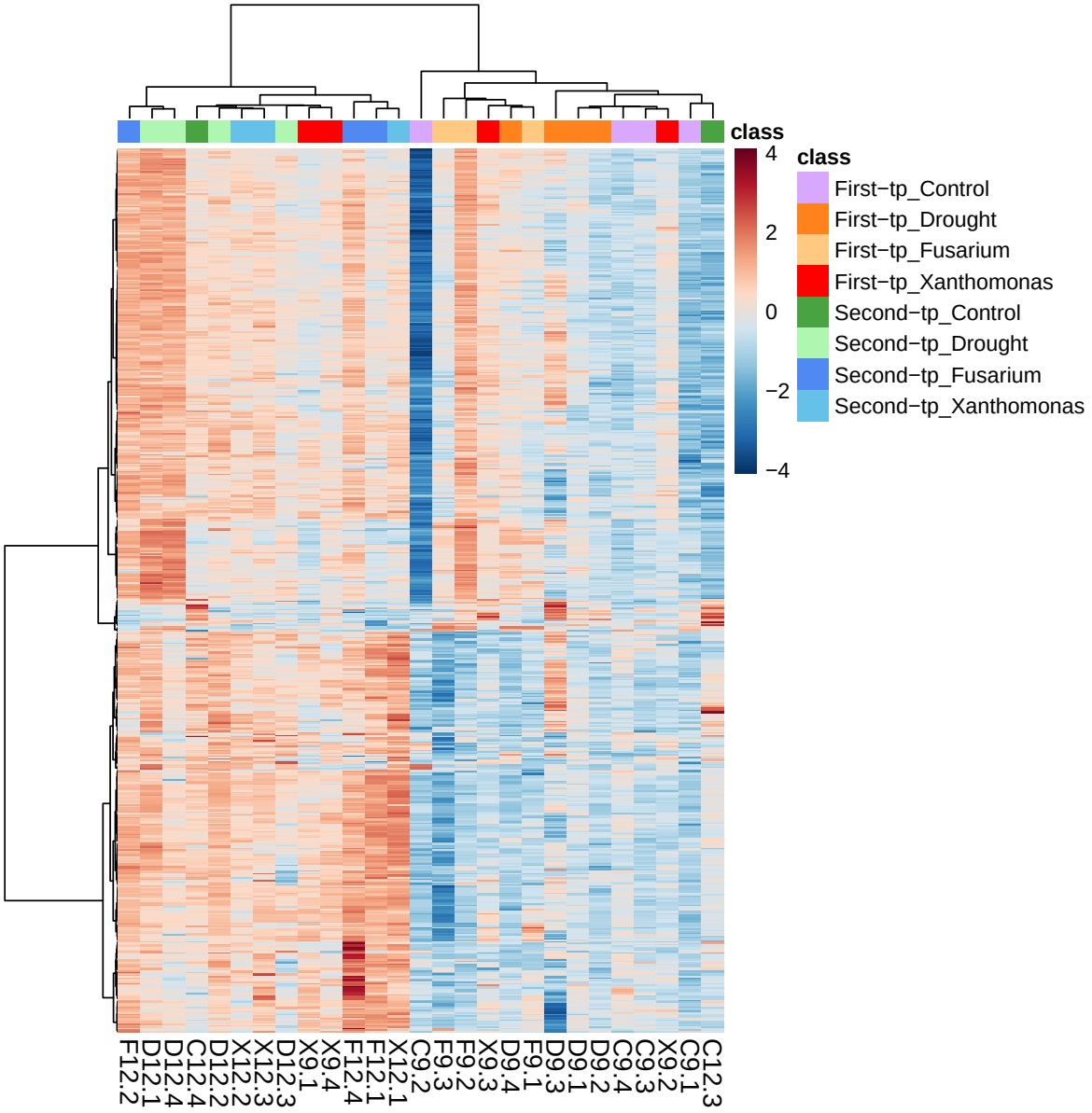

Figure 8: Clustering result shown as heatmap (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

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<-SanityCheckData(mSet)"
 [5] "mSet<-ReplaceMin(mSet);"
 [6] "mSet<-SanityCheckData(mSet)"
 [7] "mSet<-FilterVariable(mSet, \"F\", 25, \"iqr\", 0, \"mean\", 0)"
 [8] "mSet<-PreparePrenormData(mSet)"
 [9] "mSet<-GetGroupNames(mSet, \"\")"
[10] "feature.nm.vec <- c(\"\")"
[11] "smpl.nm.vec <- c(\"F9.4\",\"C12.1\",\"F12.3\")"
[12] "grp.nm.vec <- c(\"First-tp_Control\",\"First-tp_Drought\",\"First-tp_Fusarium\",\"First-tp_Xan
[13] "mSet<-UpdateData(mSet, T)"
[14] "mSet<-PreparePrenormData(mSet)"
[15] "mSet<-Normalization(mSet, \"CompNorm\", \"LogNorm\", \"ParetoNorm\", \"sodium_formate\", ratio
[16] "mSet<-PlotNormSummary(mSet, \"norm_0_\", \"png\", 72, width=NA)"
[17] "mSet<-PlotSampleNormSummary(mSet, \"snorm_0_\", \"png\", 72, width=NA)"
[18] "mSet<-ANOVA.Anal(mSet, F, 0.05, FALSE)"
[19] "mSet<-PlotANOVA(mSet, \"aov_0_\", \"png\", 72, width=NA)"
[20] "mSet<-PreparePrenormData(mSet)"
[21] "mSet<-Normalization(mSet, \"CompNorm\", \"LogNorm\", \"ParetoNorm\", \"sodium_formate\", ratio
[22] "mSet<-PlotNormSummary(mSet, \"norm_1_\", \"png\", 72, width=NA)"
[23] "mSet<-PlotSampleNormSummary(mSet, \"snorm_1_\", \"png\", 72, width=NA)"
[24] "mSet<-PlotHeatMap(mSet, \"heatmap_1_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\
[25] "mSet<-PlotHeatMap(mSet, \"heatmap_2_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\
[26] "mSet<-ANOVA.Anal(mSet, F, 0.05, FALSE)"
[27] "mSet<-PlotANOVA(mSet, \"aov_0_\", \"png\", 72, width=NA)"
[28] "mSet<-ANOVA.Anal(mSet, F, 1.0, FALSE)"
[29] "mSet<-PlotANOVA(mSet, \"aov_1_\", \"png\", 72, width=NA)"
[30] "mSet<-Calculate.ANOVA.posthoc(mSet, \"fisher\", 0.05)"
[31] "mSet<-ANOVA.Anal(mSet, F, 0.051556, FALSE)"
[32] "mSet<-PlotANOVA(mSet, \"aov_2_\", \"png\", 72, width=NA)"
[33] "mSet<-PlotSubHeatMap(mSet, \"heatmap_3_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclide
[34] "mSet<-GetGroupNames(mSet, \"null\")"
[35] "colVec<-c(\"##d8a8ff\",\"#NA\",\"#NA\",\"#NA\",\"#NA\",\"#NA\",\"#NA\",\"#NA\")"
[36] "shapeVec<-c(0,0,0,0,0,0,0,0)"
[37] "mSet<-UpdateGraphSettings(mSet, colVec, shapeVec)"
[38] "colVec<-c(\"##d8a8ff\",\"##ff821c\",\"##ffc982\",\"##ff0000\",\"#NA\",\"#NA\",\"#NA\",\"#NA\")
[39] "shapeVec<-c(0,0,0,0,0,0,0,0)"
[40] "mSet<-UpdateGraphSettings(mSet, colVec, shapeVec)"
[41] "mSet<-PlotSubHeatMap(mSet, \"heatmap_4_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclide
[42] "colVec<-c(\"##d8a8ff\",\"##ff821c\",\"##ffc982\",\"##ff0000\",\"#NA\",\"#NA\",\"#NA\",\"#NA\")
[43] "shapeVec<-c(0,0,0,0,0,0,0,0)"
[44] "mSet<-UpdateGraphSettings(mSet, colVec, shapeVec)"
[45] "mSet<-PlotSubHeatMap(mSet, \"heatmap_4_\", \"pdf\", 72, width=NA, \"norm\", \"row\", \"euclide
[46] "colVec<-c(\"##d8a8ff\",\"##ff821c\",\"##ffc982\",\"##ff0000\",\"##4aa341\",\"##aff7b0\",\"##50
[47] "shapeVec<-c(0,0,0,0,0,0,0,0)"
[48] "mSet<-UpdateGraphSettings(mSet, colVec, shapeVec)"
[49] "mSet<-PlotSubHeatMap(mSet, \"heatmap_4_\", \"pdf\", 72, width=NA, \"norm\", \"row\", \"euclide
[50] "mSet<-PLSR.Anal(mSet, reg=TRUE)"
[51] "mSet<-PlotPLSPairSummary(mSet, \"pls_pair_0_\", \"png\", 72, width=NA, 5)"
[52] "mSet<-PlotPLS2DScore(mSet, \"pls_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[53] "mSet<-PlotPLS3DScoreImg(mSet, \"pls_score3d_0_\", \"png\", 72, width=NA, 1,2,3, 40)"
[54] "mSet<-PlotPLSLoading(mSet, \"pls_loading_0_\", \"png\", 72, width=NA, 1, 2);"
[55] "mSet<-PlotPLS3DLoading(mSet, \"pls_loading3d_0_\", \"json\", 1,2,3)"
[56] "mSet<-PlotPLS.Imp(mSet, \"pls_imp_0_\", \"png\", 72, width=NA, \"vip\", \"Comp. 1\", 15,FALSE)
```

```
[57] "mSet<-SaveTransformedData(mSet)"
[58] "mSet<-PreparePDFReport(mSet, \"guest12705925767388953312\")\n"
```

The report was generated on Mon Mar 4 11:22:36 2024 with R version 4.3.2 (2023-10-31), OS system: Linux, version: -Ubuntu SMP Tue Jan 9 15:25:40 UTC 2024 .