Metabolomic Data Analysis with MetaboAnalyst 6.0

Name: guest15030457523348827757

March 1, 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 2292 (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	2282	10	2292
C9.2	2289	3	2292
C9.3	2288	4	2292
C9.4	2288	4	2292
X9.1	2288	4	2292
X9.2	2290	2	2292
X9.3	2291	1	2292
X9.4	2291	1	2292
D9.1	2291	1	2292
D9.2	2289	3	2292
D9.3	2277	15	2292
D9.4	2291	1	2292
F9.1	2287	5	2292
F9.2	2287	5	2292
F9.3	2290	2	2292
F9.4	2208	84	2292
BLANK.2	2187	105	2292
QC.1	2292	0	2292
QC.2	2291	1	2292
QC.3	2291	1	2292
D12.1	2288	4	2292
D12.2	2292	0	2292
D12.3	2291	1	2292
D12.4	2283	9	2292
F12.1	2290	2	2292
F12.2	2289	3	2292
F12.3	2216	76	2292
F12.4	2290	2	2292
C12.1	2250	42	2292
C12.3	2291	1	2292
C12.4	2289	3	2292
X12.1	2291	1	2292
X12.2	2291	1	2292
X12.3	2292	0	2292
C15.1	2290	2	2292
C15.2	2290	2	2292
C15.3	2292	0	2292
C15.4	2290	2	2292
F15.1	2231	61	2292
F15.2	2291	1	2292
F15.3	2251	41	2292
F15.4	2283	9	2292
D15.1	2200	92	2292
D15.2	2291	1	2292
D15.3	2217	75	2292
D15.4	2257	35	2292
X15.1	2283	9	2292
X15.2	2283	9	2292
X15.3	2291	1	2292
X15.4	2292	0	2292

[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

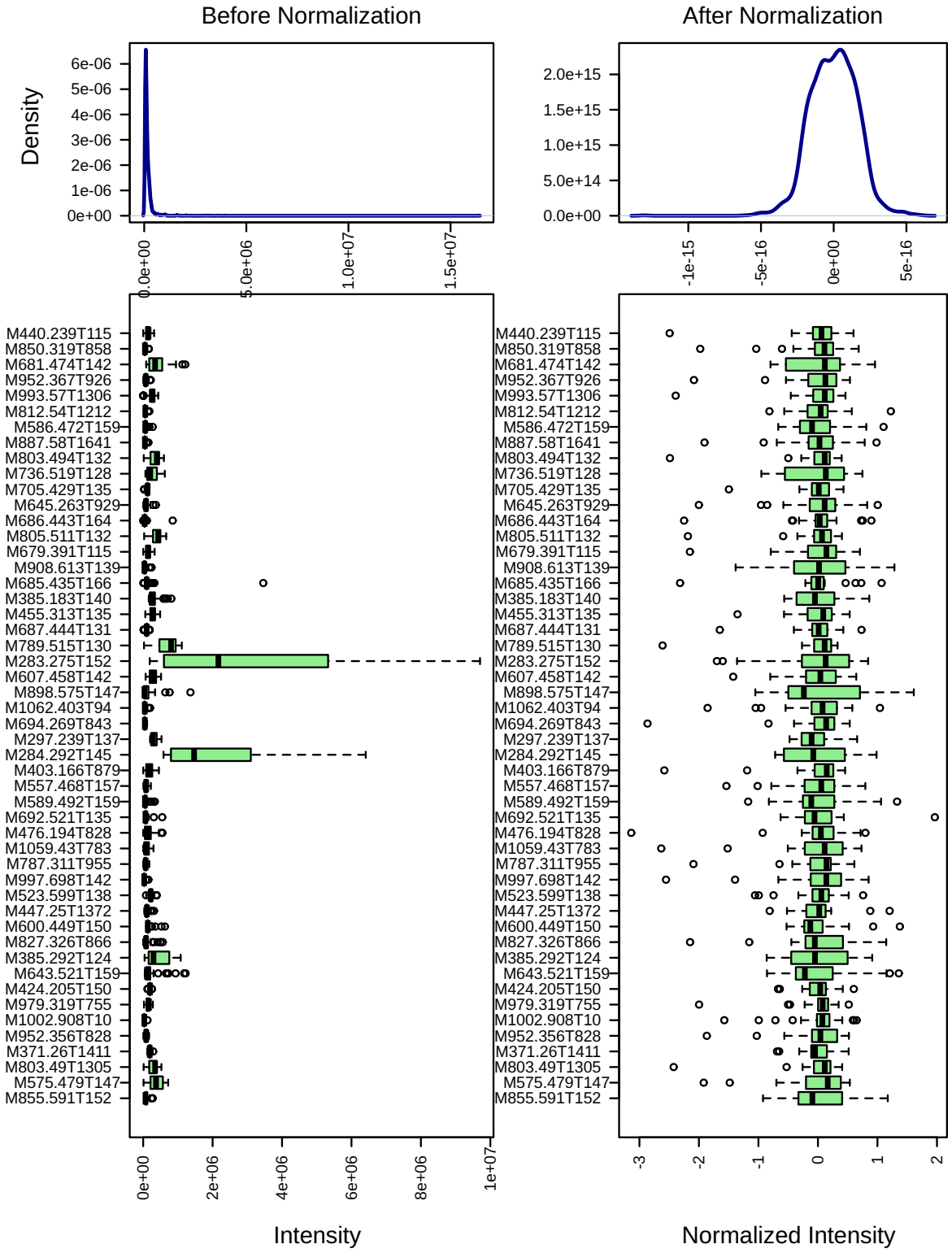

Figure 1: Box plots and kernel density plots before and after normalization. The boxplots show at most 50 features due to space limit. The density plots are based on all samples.

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.

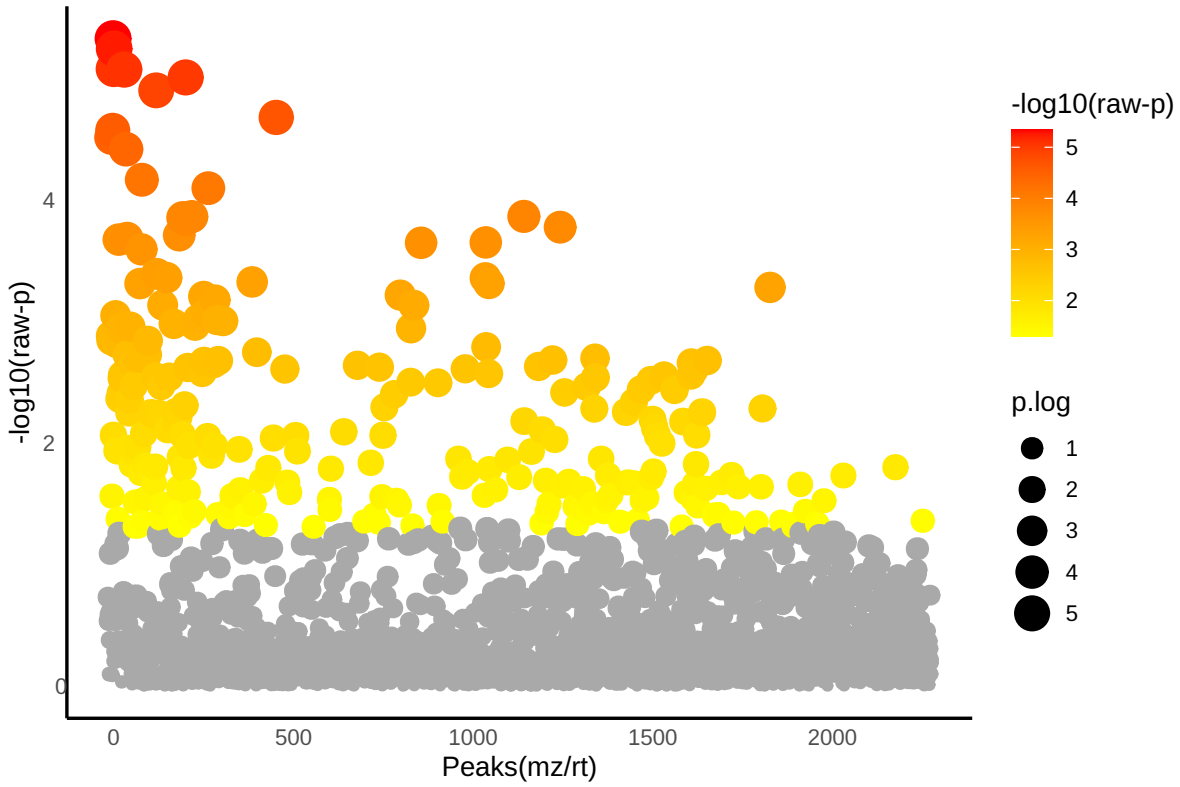

Figure 2: Important features selected by ANOVA plot with p value threshold 0.44499.

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	M680.474T1439.81	20.041	4.6147e-06	5.3359	0.0044144
2	M681.474T1424.253	19.569	5.5894e-06	5.2526	0.0044144
3	M565.421T1295.888	18.643	8.2009e-06	5.0861	0.0044144
4	M670.528T1488.279	18.617	8.2909e-06	5.0814	0.0044144
5	M670.538T1472.8	18.262	9.6342e-06	5.0162	0.0044144
6	M680.472T1423.013	17.689	1.2315e-05	4.9096	0.0047024
7	M664.462T1368.515	16.525	2.0583e-05	4.6865	0.0067364
8	M681.469T1389.183	15.987	2.6277e-05	4.5804	0.0075251
9	M681.469T1408.485	15.69	3.012e-05	4.5211	0.0076673
10	M565.409T1275.021	15.212	3.7643e-05	4.4243	0.0086239
11	M679.553T1406.451	14.005	6.7202e-05	4.1726	0.013996
12	M660.514T1439.78	13.683	7.8812e-05	4.1034	0.015047
13	M481.362T1439.996	12.624	0.00013472	3.8706	0.020424
14	M555.451T1436.867	12.617	0.00013523	3.8689	0.020424
15	M660.517T1423.005	12.578	0.00013797	3.8602	0.020424
16	M555.447T1456.772	12.514	0.00014264	3.8458	0.020424
17	M481.361T1422.828	12.232	0.00016527	3.7818	0.022272
18	M603.415T1456.538	11.944	0.00019238	3.7158	0.023151
19	M695.45T1338.973	11.84	0.00020332	3.6918	0.023151
20	M682.472T1373.006	11.786	0.00020928	3.6793	0.023151
21	M581.399T1399.634	11.686	0.00022069	3.6562	0.023151
22	M688.545T1335.803	11.673	0.00022232	3.653	0.023151
23	M555.456T1423.317	11.441	0.00025193	3.5987	0.025095
24	M664.46T1406.422	10.597	0.00040083	3.397	0.038086
25	M555.454T1439.586	10.467	0.00043123	3.3653	0.038086
26	M547.366T1373.389	10.458	0.00043344	3.3631	0.038086
27	M603.41T1440.081	10.325	0.00046726	3.3304	0.038086
28	M567.43T1421.35	10.275	0.00048081	3.318	0.038086
29	M593.489T1519.555	10.27	0.0004821	3.3169	0.038086
30	M702.545T1440.116	10.142	0.00051849	3.2853	0.039596
31	M603.424T1440.046	9.8875	0.00060011	3.2218	0.044005
32	M664.461T1422.743	9.846	0.00061465	3.2114	0.044005
33	M663.458T1406.394	9.7293	0.00065768	3.182	0.045659
34	M565.397T1373.252	9.5615	0.00072531	3.1395	0.047628
35	M670.539T1507.08	9.556	0.00072763	3.1381	0.047628
36	M555.365T1423.011	9.3027	0.00084467	3.0733	0.053754
37	M679.452T1355.733	9.2283	0.00088277	3.0542	0.05466
38	M565.417T1439.955	9.1083	0.00094818	3.0231	0.056147
39	M727.561T1540.694	9.0864	0.00096066	3.0174	0.056147
40	M727.571T1439.78	9.0526	0.00098031	3.0086	0.056147
41	M565.403T1389.183	8.9609	0.0010357	2.9848	0.057602
42	M565.41T1416.7	8.9039	0.0010719	2.9699	0.057602
43	M637.415T1439.736	8.8709	0.0010934	2.9612	0.057602
44	M687.446T1352.626	8.8217	0.0011264	2.9483	0.057602
45	M637.412T1422.711	8.8143	0.0011314	2.9464	0.057602
46	M679.436T1338.726	8.5921	0.0012951	2.8877	0.064502
47	M646.455T1423.013	8.4794	0.0013877	2.8577	0.066661
48	M589.459T1372.544	8.4458	0.0014167	2.8487	0.066661
49	M668.509T1456.373	8.4354	0.0014257	2.846	0.066661
50	M637.411T1406.122	8.3518	0.0015012	2.8236	0.068784

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

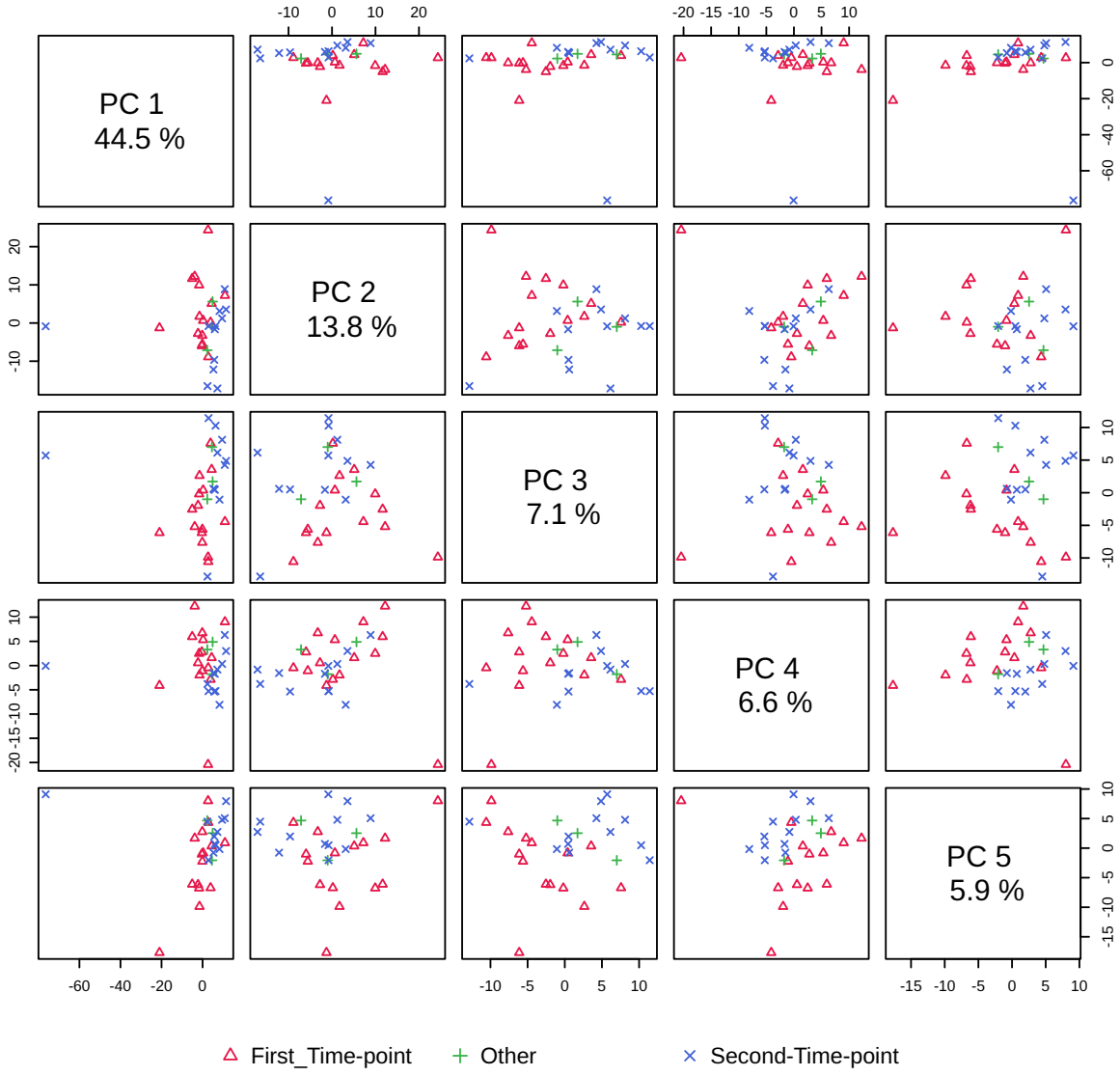

Figure 3: Pairwise score plots between the selected PCs. The explained variance of each PC is shown in the corresponding diagonal cell.

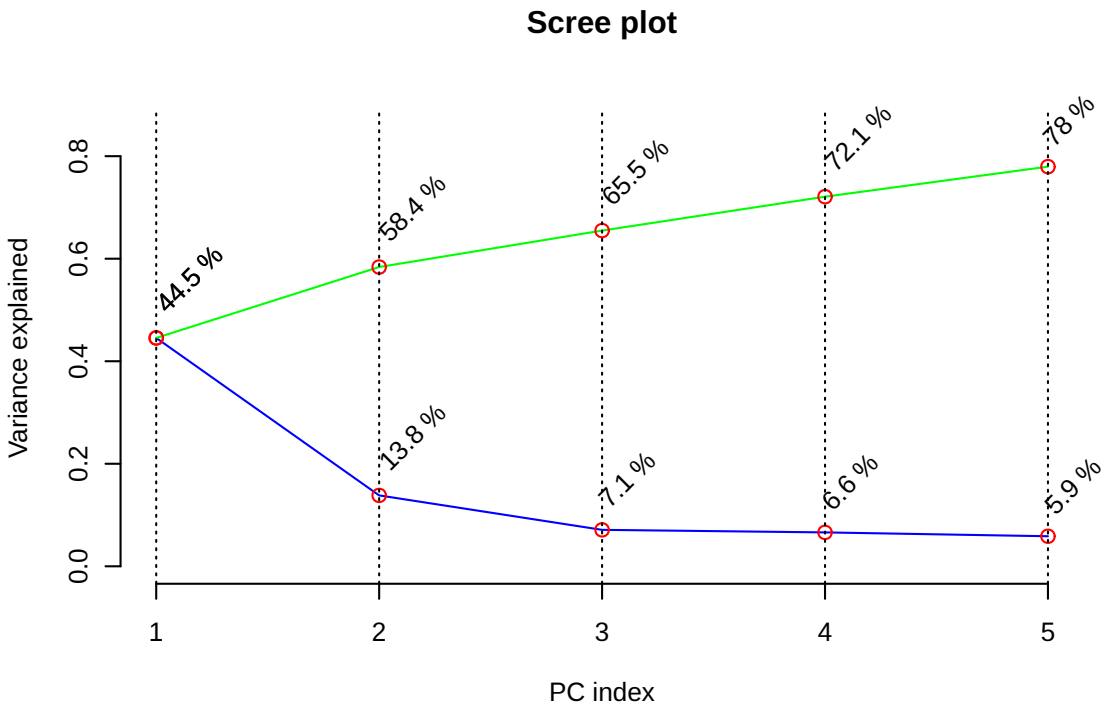

Figure 4: Scree plot shows the variance explained by PCs. The green line on top shows the accumulated variance explained; the blue line underneath shows the variance explained by individual PC.

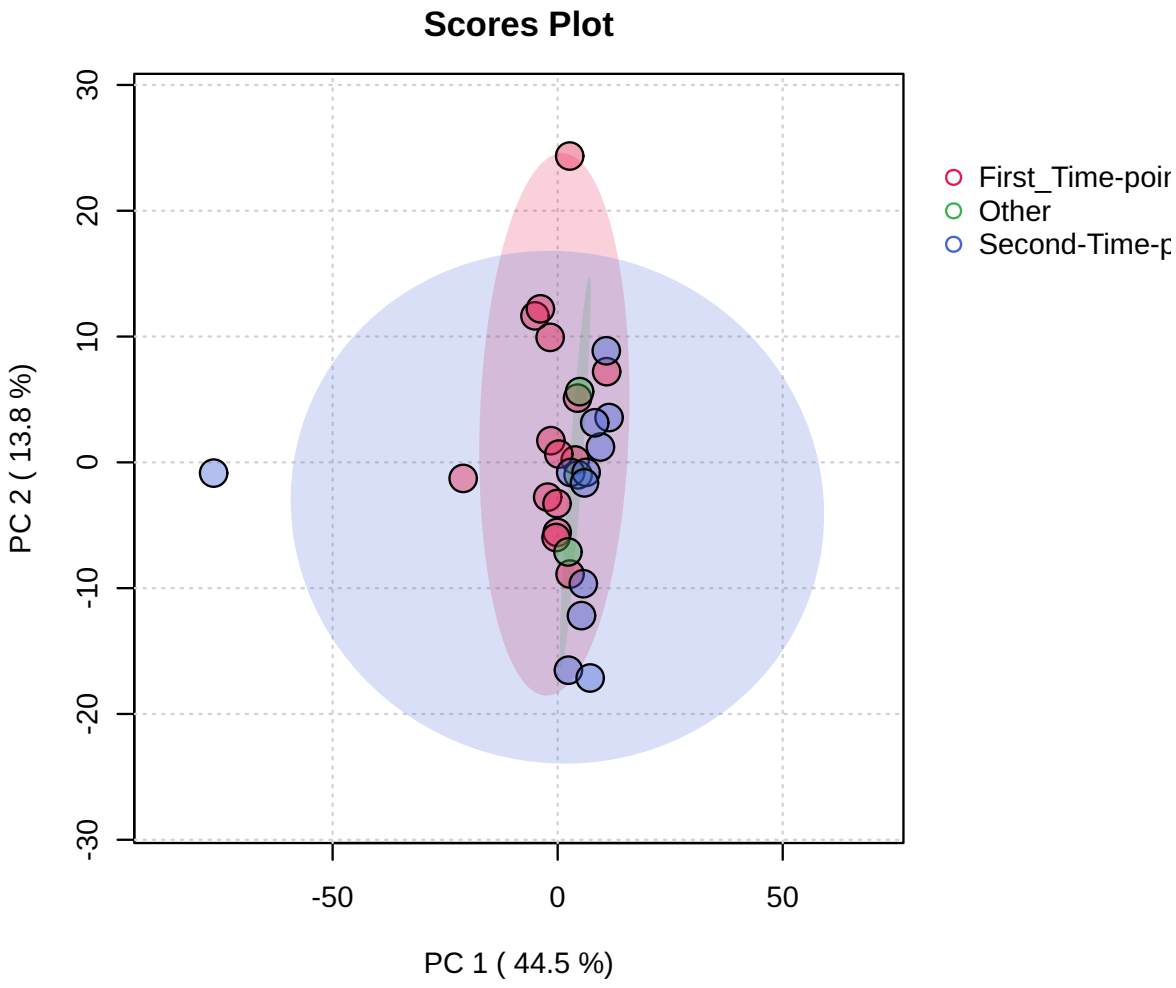

Figure 5: Scores plot between the selected PCs. The explained variances are shown in brackets.

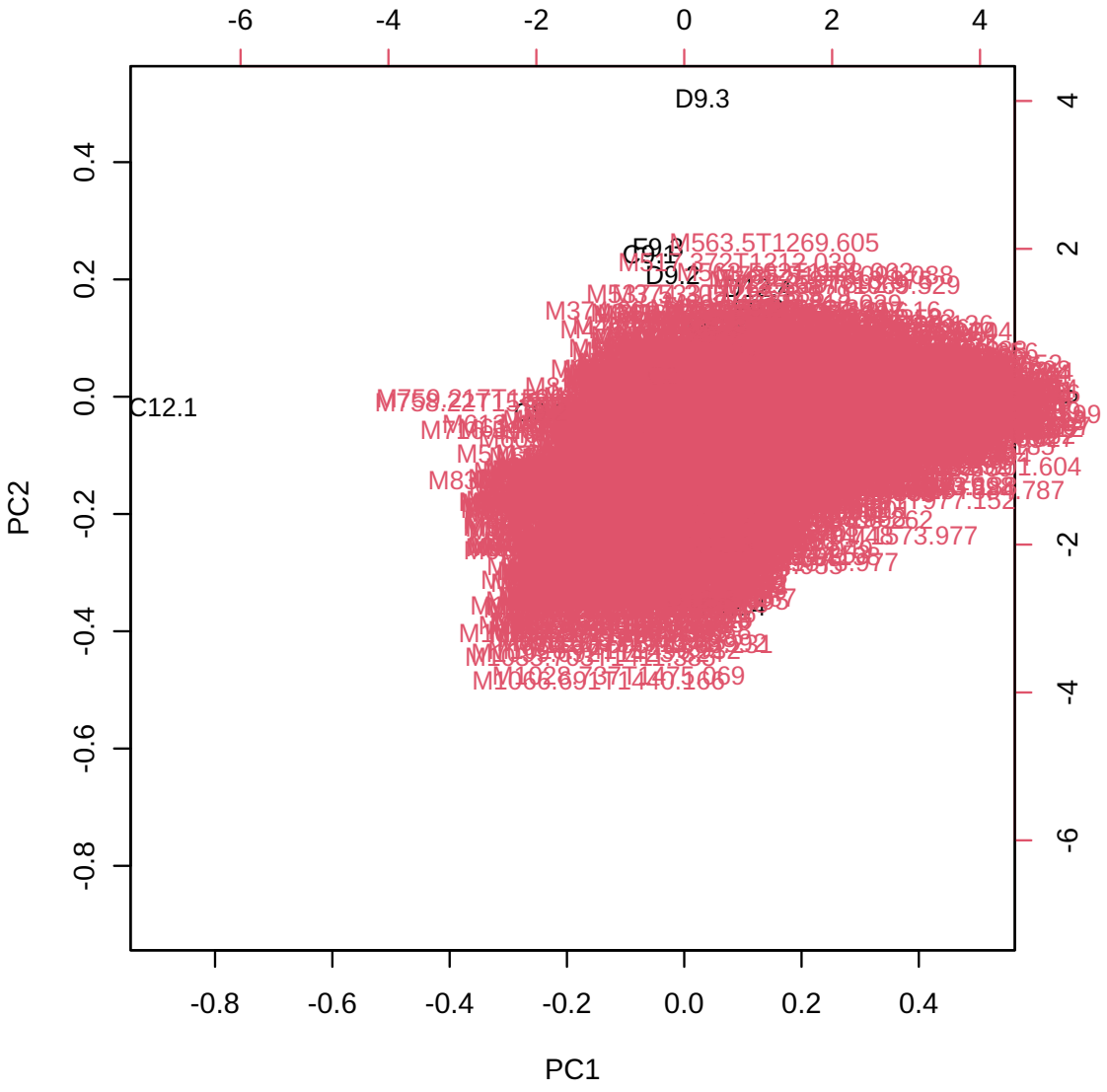

Figure 6: PCA biplot between the selected PCs. Note, you may want to test different centering and scaling normalization methods for the biplot to be displayed properly.

2.3 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 7 shows the clustering result in the form of a dendrogram. Figure 8 shows the clustering result in the form of a heatmap.

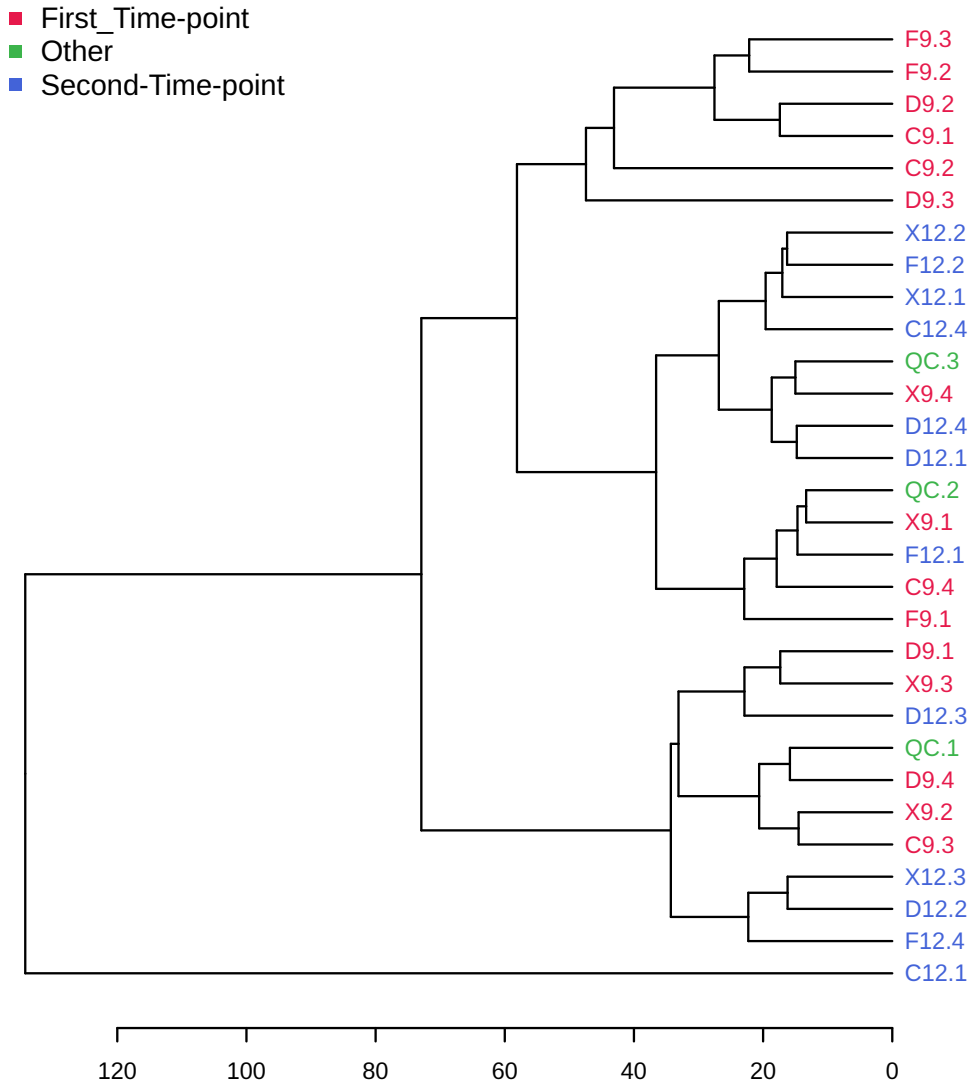

Figure 7: Clustering result shown as dendrogram (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

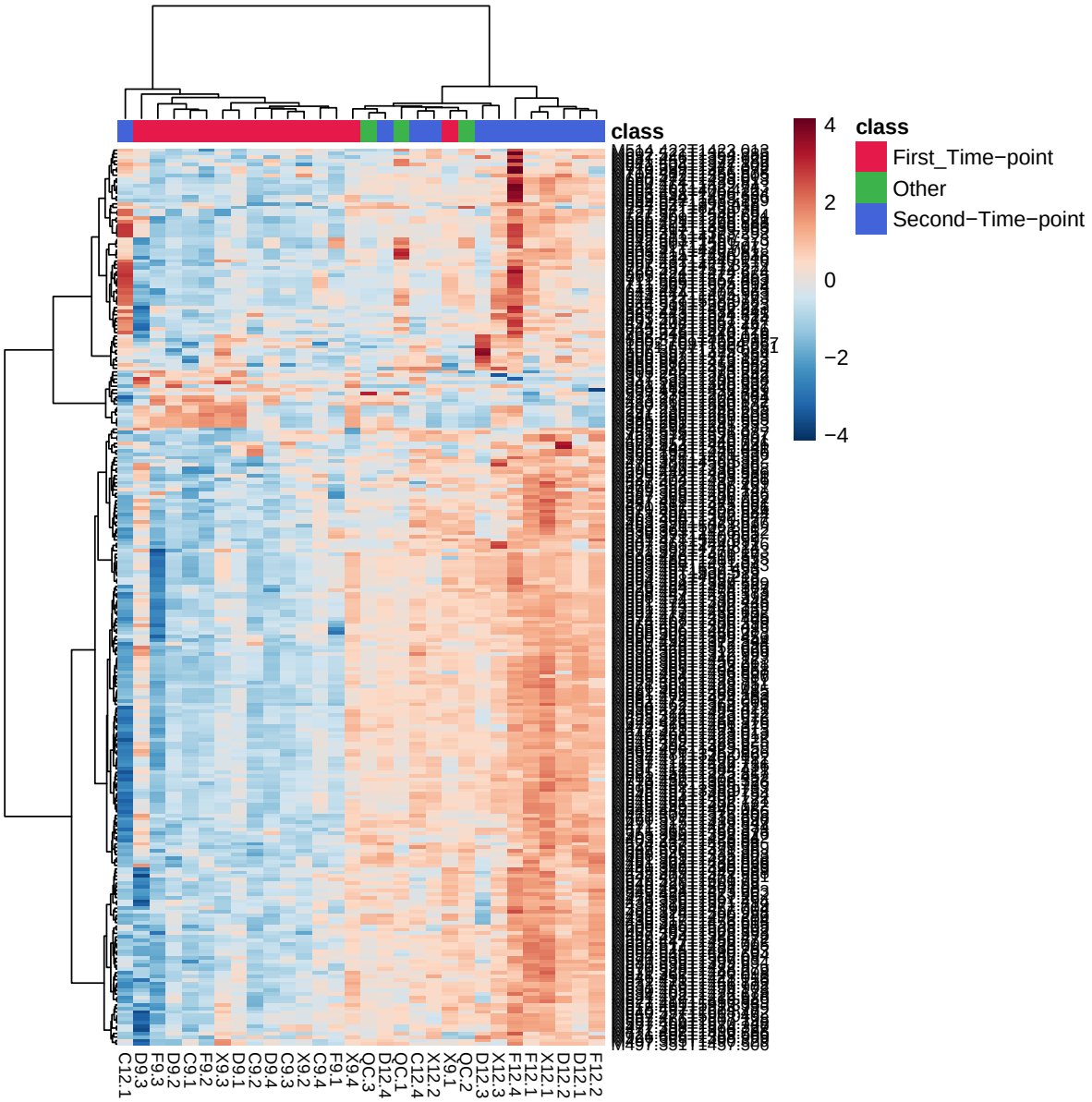

Figure 8: Clustering result shown as heatmap (distance measure using `euclidean`, and clustering algorithm using `ward.D`).

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<-ReplaceMin(mSet);"
 [5] "mSet<-SanityCheckData(mSet)"
 [6] "mSet<-FilterVariable(mSet, \"F\", 25, \"iqr\", 0, \"mean\", 0)"
 [7] "mSet<-PreparePrenormData(mSet)"
 [8] "mSet<-Normalization(mSet, \"CompNorm\", \"LogNorm\", \"ParetoNorm\", \"sodium_formate\", ratio
 [9] "mSet<-PlotNormSummary(mSet, \"norm_0_\", \"png\", 72, width=NA)"
[10] "mSet<-PlotSampleNormSummary(mSet, \"snorm_0_\", \"png\", 72, width=NA)"
[11] "mSet<-GetGroupNames(mSet, \"\")"
[12] "feature.nm.vec <- c(\"\")"
[13] "smpl.nm.vec <- c(\"F9.4\",\"F12.3\",\"C12.3\")"
[14] "grp.nm.vec <- c(\"First_Time-point\",\"Other\",\"Second-Time-point\")"
[15] "mSet<-UpdateData(mSet, T)"
[16] "mSet<-PreparePrenormData(mSet)"
[17] "mSet<-Normalization(mSet, \"CompNorm\", \"LogNorm\", \"ParetoNorm\", \"sodium_formate\", ratio
[18] "mSet<-PlotNormSummary(mSet, \"norm_1_\", \"png\", 72, width=NA)"
[19] "mSet<-PlotSampleNormSummary(mSet, \"snorm_1_\", \"png\", 72, width=NA)"
[20] "feature.nm.vec <- c(\"\")"
[21] "smpl.nm.vec <- c(\"F9.4\",\"F12.3\",\"C12.3\",\"BLANK.2\")"
[22] "grp.nm.vec <- c(\"First_Time-point\",\"Other\",\"Second-Time-point\")"
[23] "mSet<-UpdateData(mSet, T)"
[24] "mSet<-PreparePrenormData(mSet)"
[25] "mSet<-Normalization(mSet, \"CompNorm\", \"LogNorm\", \"ParetoNorm\", \"sodium_formate\", ratio
[26] "mSet<-PlotNormSummary(mSet, \"norm_2_\", \"png\", 72, width=NA)"
[27] "mSet<-PlotSampleNormSummary(mSet, \"snorm_2_\", \"png\", 72, width=NA)"
[28] "mSet<-ANOVA.Anal(mSet, F, 0.05, FALSE)"
[29] "mSet<-PlotANOVA(mSet, \"aov_0_\", \"png\", 72, width=NA)"
[30] "mSet<-PlotHeatMap(mSet, \"heatmap_1_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclidean\
[31] "mSet<-ANOVA.Anal(mSet, F, 1.0, FALSE)"
[32] "mSet<-PlotANOVA(mSet, \"aov_1_\", \"png\", 72, width=NA)"
[33] "mSet<-Calculate.ANOVA.posthoc(mSet, \"fisher\", 0.05)"
[34] "mSet<-ANOVA.Anal(mSet, F, 0.44499, FALSE)"
[35] "mSet<-PlotANOVA(mSet, \"aov_2_\", \"png\", 72, width=NA)"
[36] "mSet<-PlotSubHeatMap(mSet, \"heatmap_2_\", \"png\", 72, width=NA, \"norm\", \"row\", \"euclide
[37] "mSet<-PlotHCTree(mSet, \"tree_0_\", \"png\", 72, width=NA, \"euclidean\", \"ward.D\")"
[38] "mSet<-PCA.Anal(mSet)"
[39] "mSet<-PlotPCAPairSummary(mSet, \"pca_pair_0_\", \"png\", 72, width=NA, 5)"
[40] "mSet<-PlotPCAScree(mSet, \"pca_scree_0_\", \"png\", 72, width=NA, 5)"
[41] "mSet<-PlotPCA2DScore(mSet, \"pca_score2d_0_\", \"png\", 72, width=NA, 1,2,0.95,0,0, \"na\")"
[42] "mSet<-PlotPCALoading(mSet, \"pca_loading_0_\", \"png\", 72, width=NA, 1,2);"
[43] "mSet<-PlotPCABiplot(mSet, \"pca_biplot_0_\", \"png\", 72, width=NA, 1,2)"
[44] "mSet<-PlotPCA3DLoading(mSet, \"pca_loading3d_0_\", \"json\", 1,2,3)"
[45] "mSet<-SaveTransformedData(mSet)"
[46] "mSet<-PreparePDFReport(mSet, \"guest15030457523348827757\")\n"
```

The report was generated on Fri Mar 1 05:55:32 2024 with R version 4.3.2 (2023-10-31), OS system: Linux, version: -Ubuntu SMP Tue Jan 9 15:25:40 UTC 2024 .