

MA scripts

The bioinformatics group at Metabolomics Australia have created a number of scripts for use with R. These provide basic statistical functionality in a manner that allows a standard data input format to be transformed to a standard output format, allowing much more simple data analysis methods.

Input data format

If there is a consistent input format, scripts are much simpler to execute and troubleshoot if they do not work. In brief, the matrix should have the first column containing sample names, the second the groups and the remaining columns the variables to be processed by the script (Figure 1). These can be metabolites, retention times, masses, bins, or anything else that has been measured for the samples.

	Group	m1	m2	m3	m4	m5	m6	m7	m8	m9
S1	A	0.0737	0.6097	0.2508	0.3786	0.0885	0.363	0.079	1.0104	0.5021
S2	A	0.0827	0.6223	0.2572	0.3923	0.0953	0.3506	0.0789	1.0436	0.4875
S3	A	0.0764	0.6432	0.2533	0.3419	0.0933	0.3698	0.0768	0.999	0.5194
S4	A	0.0709	0.6341	0.2957	0.3356	0.0914	0.3268	0.0892	0.9002	0.5053
S5	B	0.0766	0.6557	0.2718	0.3848	0.0937	0.3115	0.2954	0.8977	0.4908
S6	B	0.0766	0.6331	0.2688	0.3703	0.0969	0.3173	0.2834	0.9831	0.5483
S7	B	0.0845	0.573	0.2661	0.3982	0.0815	0.3451	0.3061	1.0209	0.4658
S8	B	0.0768	0.6464	0.2724	0.395	0.0928	0.3582	0.2776	1.0826	0.5219

Figure 1: The standard format of the data matrix for statistical analyses.

Currently available scripts

Unless specified otherwise, the scripts in the ma-bioinformatics scripts repository, located at <http://code.google.com/p/ma-bioinformatics/> use the input data format specified above. These scripts are still under development and generally have comments in them that describes what each step achieves.

Examples provided below are created using the file `input.csv`.

General scripts

`boxplot.r`

This script will produce a boxplot for both samples (Figure 2) and variables (Figure 3).

`mms_plot.r`

Produces a plot of the mean, median and standard deviation across all samples (Figure 4). Enables a graphical overview of similarity of samples, and allows for rapid identification of outliers.

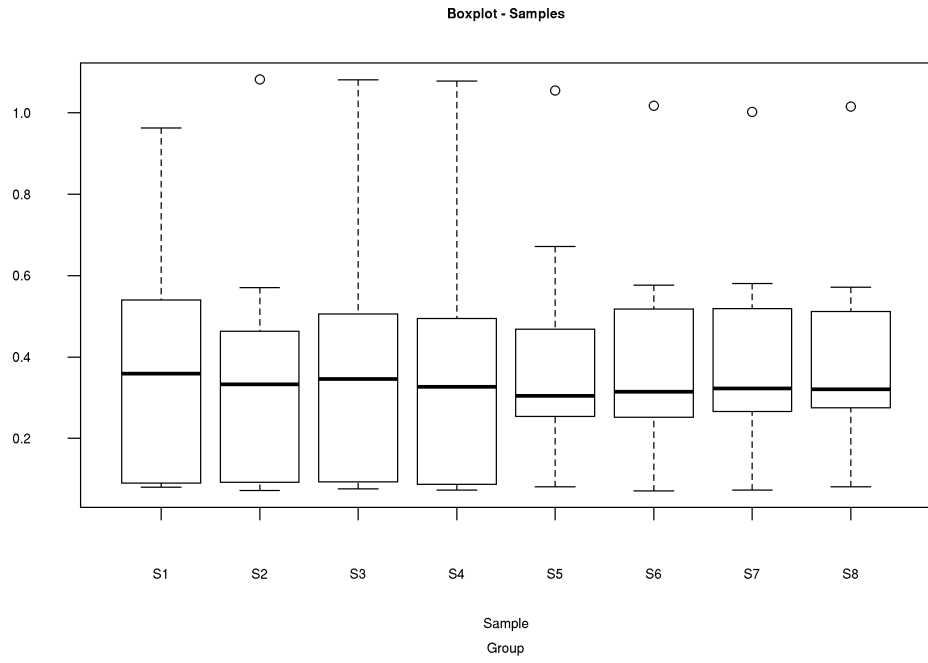


Figure 2: Boxplot for samples using data from input.csv.

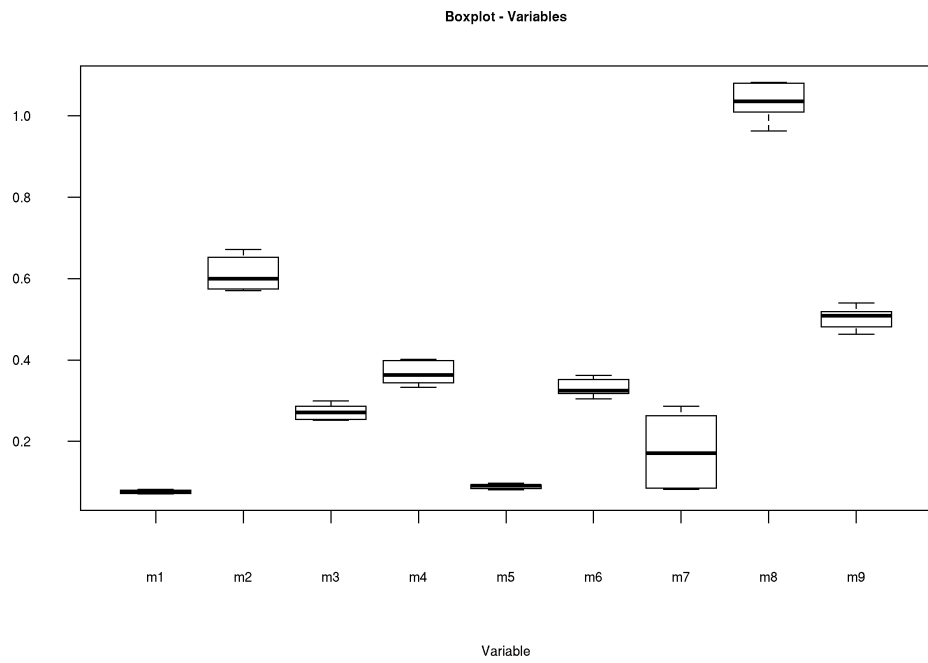


Figure 3: Boxplot for variables using input.csv.

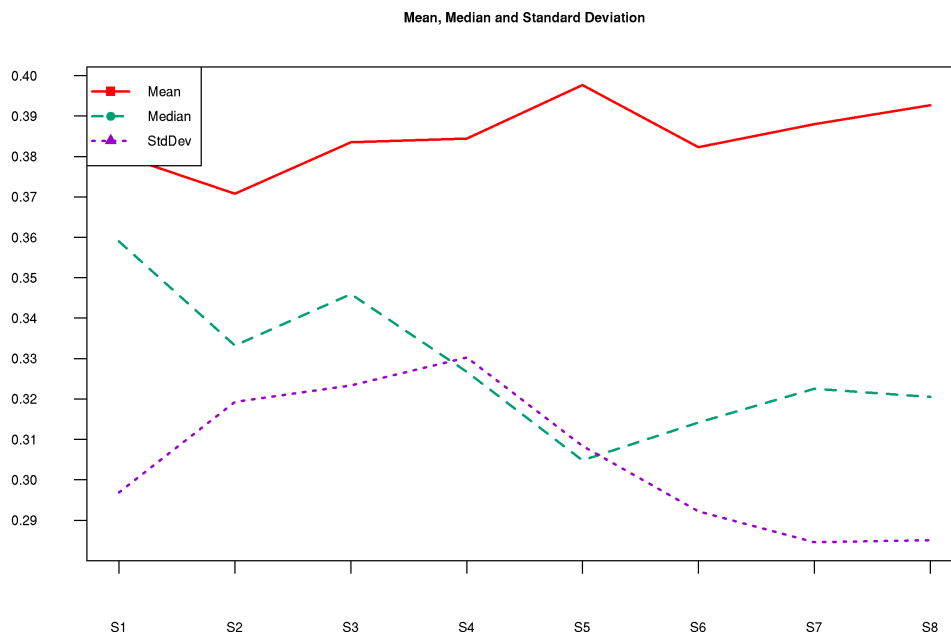


Figure 4: Line plot showing the mean, median and standard deviation.

mean_sd.r

It is important to explore a data set prior to choosing any normalisation or transformation method. Heteroscedastic noise (where the standard deviation of each metabolite in replicate samples changes with the mean of the metabolite) may change the results from the normalised data profoundly.

The optimal transformation to change heteroscedastic noise into homoscedastic noise depends on the structure of the heteroscedasticity in the signals. If the standard deviation is proportional to the mean of the signal (as in Figure 5), then a log transform is optimal to treat the data. If the standard deviation is proportional to the root of the mean then square root transformation provides a homoscedastic noise pattern (as in Figure 6).

t_test.r

Performs a Student's t -test, and outputs a .csv file with the P -values for each variable. This test assumes a normal distribution [p_t_test.csv].

wilcoxn.r

Performs Wilcoxon Rank Sum Test, a test that does not assume a normal distribution. Output is a .csv file containing the P -values for each variable [p_wilcoxon_test.csv].

fold_chg.r

Calculates the fold change from the (alphanumerically) first group in the data matrix (i.e. for our example input data, this would be fold change from group A). Fold change is

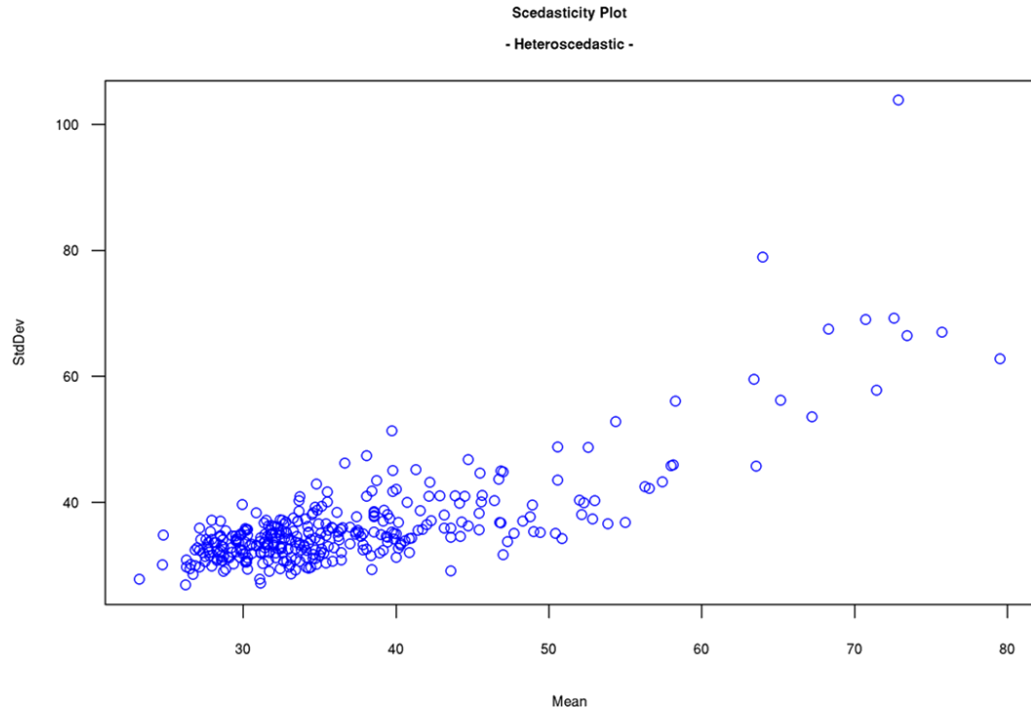


Figure 5: Heteroscedastic data. The overall shape of the data is a fanned line, as the standard deviation increases with the size of the mean.

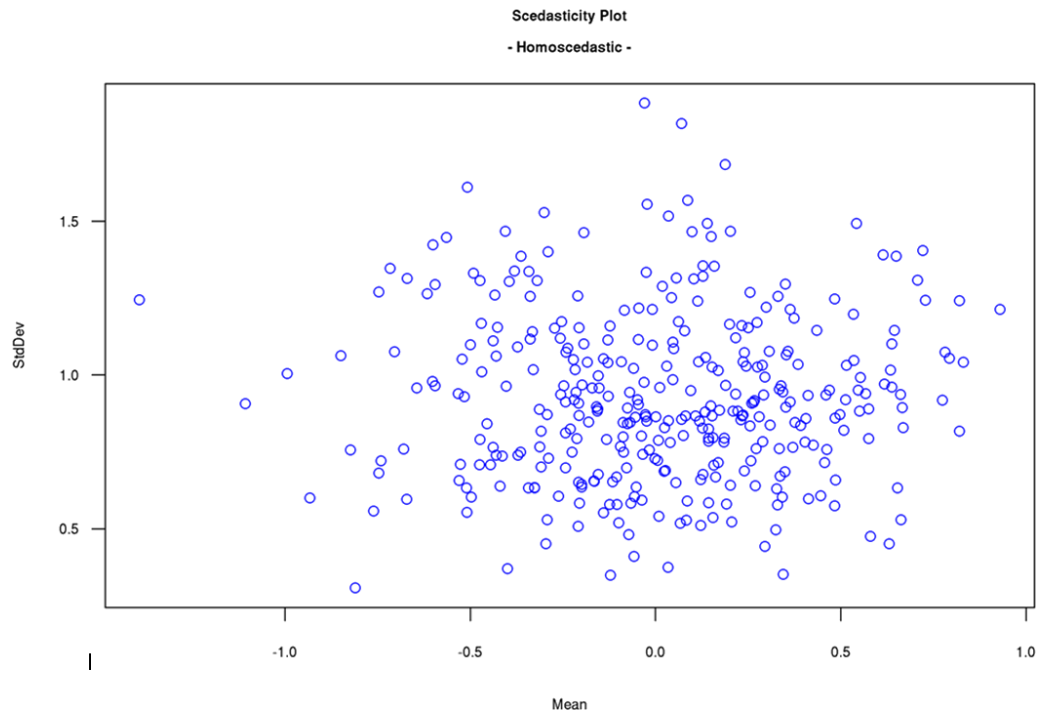


Figure 6: Homoscedastic data. Points are distributed randomly over the plot, and standard deviation does not vary with the size of the mean.

calculated by determining the mean for each variable in each sample, and then dividing each mean by the mean for the first group. If the value is less than 1, the returned value is the negative inverse (i.e. if $\frac{mean_{expt}}{mean_{ctrl}} > 1$, no change; if $\frac{mean_{expt}}{mean_{ctrl}} < 1$, then fold change $= -1 \times \frac{mean_{ctrl}}{mean_{expt}}$). It returns a table of the means as well as the table of fold changes.

volcano_plot.r

A volcano plot is a type of scatter plot which shows fold change on the x -axis and the statistical significance (P -values from a t -test) on the y -axis. It readily shows which metabolites have large magnitude changes that are statistically significant between two conditions.

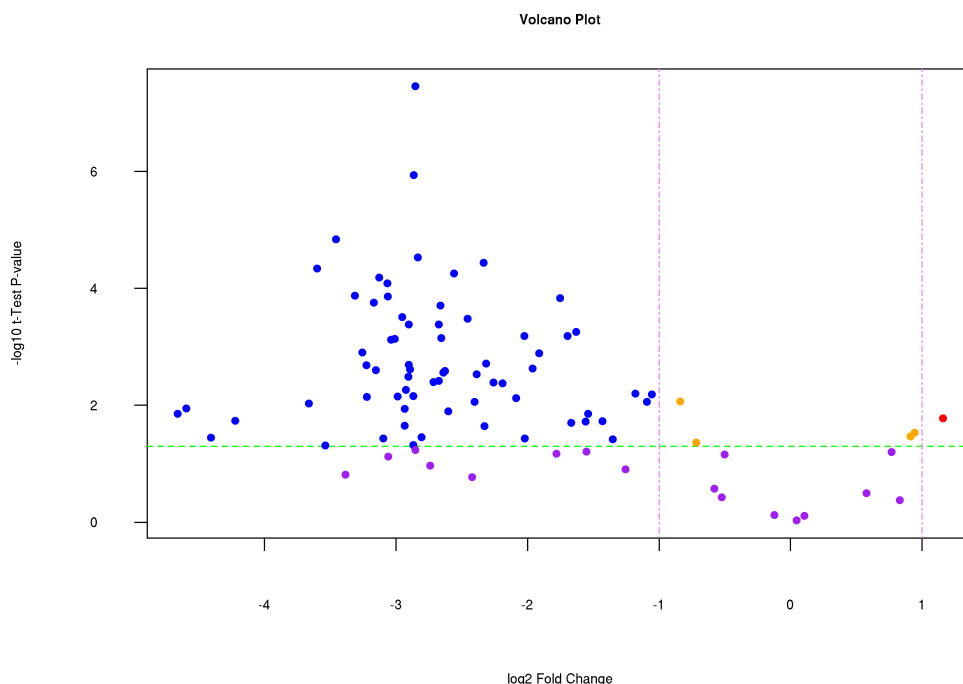


Figure 7: Volcano plot.

correlation.r

This script calculates the correlation between the variables (metabolites) in a data set. The output [cor_matrix.csv] is a correlation matrix that can be used to draw a heat map (using the heatmap.r script, the output of which is shown in figure 8), an example of which is presented in table 1. The output cor_variables.csv gives a .csv file in which the correlation coefficients between all the variables are ordered in decreasing values of correlation co-efficient (table 2).

z-score.r

This function assumes that the input data has been normalised and transformed appropriately. If that is the case, then running this script will produce a plot of variables

Table 1: Output correlation matrix.

	m1	m2	m3	m4	m5	m6	m7	m8	m9
m1	1	-0.551	-0.407	0.696	-0.319	0.142	0.344	0.512	-0.527
m2	-0.551	1	0.234	-0.358	0.821	-0.236	-0.052	-0.308	0.580
m3	-0.407	0.234	1	-0.326	0.042	-0.619	0.230	-0.546	0.066
m4	0.696	-0.358	-0.326	1	-0.260	0.033	0.553	0.546	-0.407
m5	-0.319	0.821	0.042	-0.260	1	-0.259	-0.155	-0.121	0.690
m6	0.142	-0.236	-0.619	0.033	-0.259	1	-0.502	0.714	-0.078
m7	0.344	-0.052	0.230	0.553	-0.155	-0.502	1	0.024	0.004
m8	0.512	-0.308	-0.546	0.546	-0.121	0.714	0.024	1	0.041
m9	-0.527	0.580	0.066	-0.407	0.690	-0.078	0.004	0.041	1

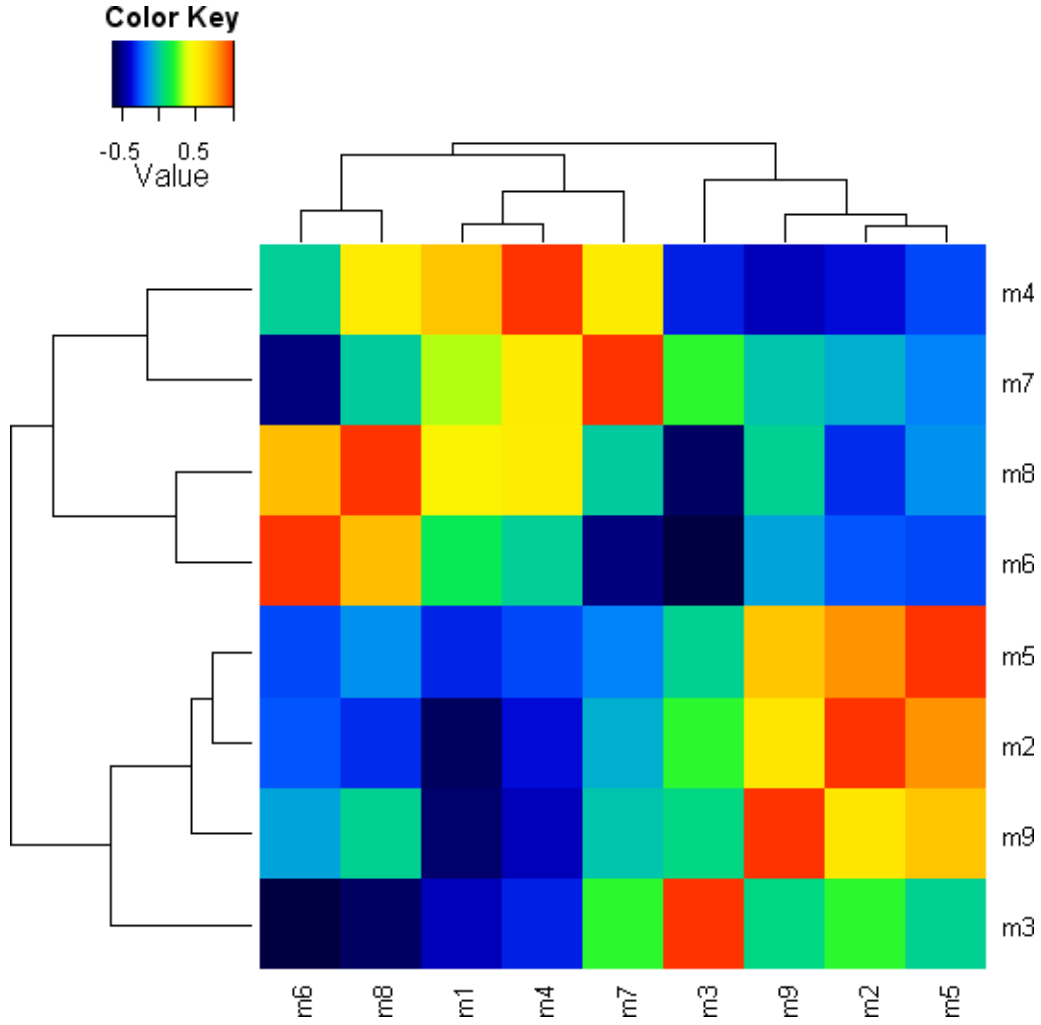


Figure 8: Heatmap of the correlation matrix (drawn using heatmap.r script)

Table 2: Output cor_variables.csv file

Variable_1	Variable_2	cor_coefficient
m5	m2	0.821
m8	m6	0.714
m4	m1	0.696
m9	m5	0.690
m9	m2	0.580
m7	m4	0.553
m8	m4	0.546
m8	m1	0.512
m7	m1	0.344
m3	m2	0.234
m7	m3	0.230
m6	m1	0.142
m9	m3	0.066
m5	m3	0.042
m9	m8	0.041
m6	m4	0.033
m8	m7	0.024
m9	m7	0.004
m7	m2	-0.052
m9	m6	-0.078
m8	m5	-0.121
m7	m5	-0.155
m6	m2	-0.236
m6	m5	-0.259
m5	m4	-0.260
m8	m2	-0.308
m5	m1	-0.319
m4	m3	-0.326
m4	m2	-0.358
m3	m1	-0.407
m9	m4	-0.407
m7	m6	-0.502
m9	m1	-0.527
m8	m3	-0.546
m2	m1	-0.551
m6	m3	-0.619

against the z-score for each of the variable. The z-score is calculated as $z = \frac{x-\mu}{\sigma}$, where x is the value of the variable to be normalised, μ is the mean and σ is the standard deviation of the variable for that group.

Transformation

This is an operation that is applied to each variable, and the type of transformation applied depends on the statistical test that is necessary to answer a biological question. For example, a t-test assumes that the data is normally distributed, hence it is important to transform a right- or left-skewed data set to a normal distribution.

log.r

Performs a \log_{10} transform of the input data, and generates a new data matrix [log_transform.csv].

Normalisation

This process is applied to the entire data set, and enables samples to be directly compared to one another by reducing the influence that is non-biological in origin. As an example, running samples on different days can cause samples to appear different when in actual fact they are the same. Normalisation attempt to removes this influence so that samples from different groups can directly be compared. The output of these scripts are all .csv files.

norm_is.r

Uses an internal standard (in the column immediately following the “Group” column) to normalise the data set [norm_data_IS.csv].

norm_ext_vec.r

Uses an external vector (e.g. creatinine for urine samples saved as a separate file) to normalise the data set [norm_data_ext_vec.csv].

norm_median.r

Takes the median value for each sample (i.e. row) and uses this value to normalise the data set [norm_data_median.csv].

norm_Zscore.r

Z score is a measure of how many standard deviations a sample is from the mean. This script calculates the Z score for each sample and uses that value to normalise the data set [norm_data_zscore.csv].

norm_sum.r

This script normalises the values on a per-sample basis by taking the sum of all values for each sample and using that value as the normalisation factor [norm_data_sum.csv].

norm_quantile.r

This script normalises the data based upon quantiles of each sample [norm_data_quantile.csv].

Cluster analysis

hca.r

Hierarchical cluster analysis (HCA) is a technique that identifies groups of samples that show similar characteristics, and then quantifies the structural characteristics of the sample (or variable) data by constructing a hierarchy in a tree-like structure. There are two kinds of procedures that are commonly used to construct the tree, namely agglomerative and divisive. In an agglomerative procedure, each sample (or variable) starts in a cluster of its own, and then continually joins clusters until there is only one cluster containing all samples. The divisive method operates in the exact reverse.

This script uses complete linkage method but other different linkage measures can be used in HCA, including average, single and Ward's, which may result in different clusters. The main objective of HCA is to classify the data into groups by structuring it, which helps to identify relationships among observations (samples/variables).

This script uses Manhattan distance to calculate the similarity, but other methods can be used by changing “manhattan” to one of “euclidean”, “maximum”, “canberra”, “binary” or “minkowski”.

It will generate a plot of both the sample distances (see Figure 9) as well as the distance for the variables (Figure 10).

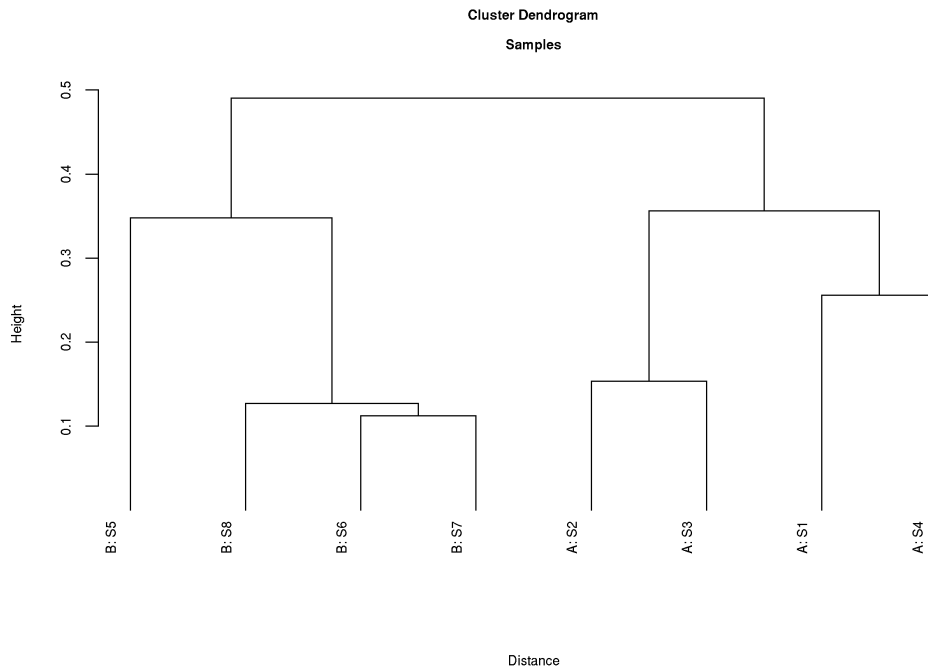


Figure 9: Hierarchical Cluster Analysis (HCA) for samples. Group labels are incorporated in the names on the leaves of the dendrogram.

pca.r

Principal Component Analysis (PCA) is a data transformation technique which is used to reduce a multidimensional data set to a lower number of dimensions for further analysis.

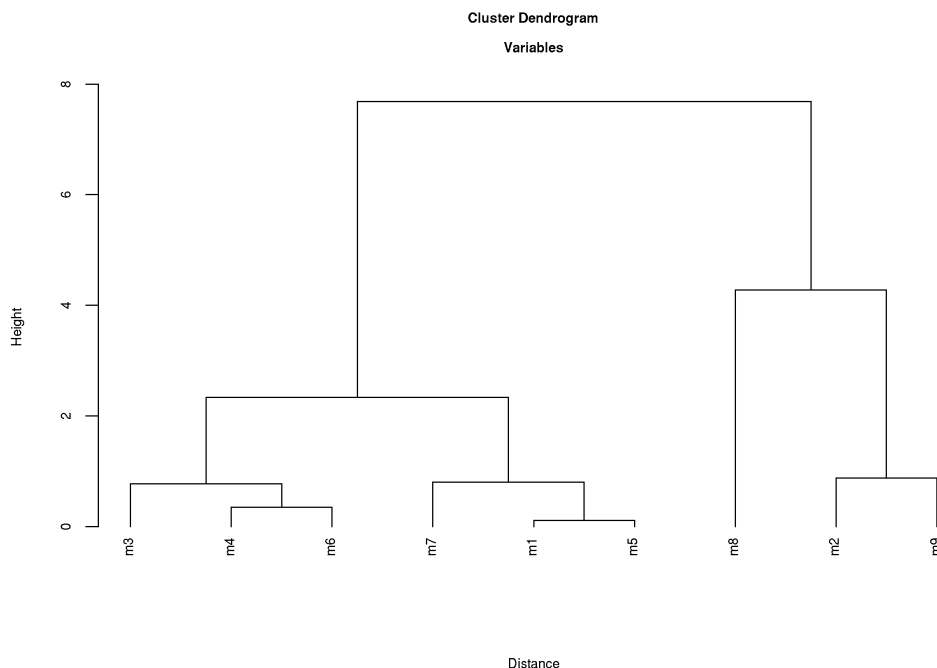


Figure 10: HCA for variables.

In PCA, a data set of interrelated variables is transformed to a new set of variables called principal components (PCs) in such a way that they are uncorrelated, and the first few of these PCs retain most of the variation present in the entire data set. The first PC is a linear combination of all the actual variables in such a way that it has the greatest amount of variation, and the second PC is a combination of the variables that have the next greatest variation in the remaining PCs.

This script produces multiple plots – residual variance (Figure 11), scores plot labelled with sample names (Figure 12), scores plot labelled with group names (Figure 13) and a loading plot (Figure 14).

lda.r

Linear Discriminant Analysis (LDA) is a classical technique to predict groups of samples. This is a supervised method that requires prior knowledge of the groups. LDA is therefore well suited for non-targeted metabolic profiling data which is almost always “grouped”. LDA is very similar to PCA, except that the technique maximises the ratio of between-class variance to the within-class variance in a set of data, and thus gives maximal separation between the classes, as shown in Figure 15.

heatmap.r

Produces a hierarchical cluster dendrogram for both samples (x -axis) and variables (y -axis), and uses colours to represent the values. This results in a readily interpretable figure as shown in Figure 16.

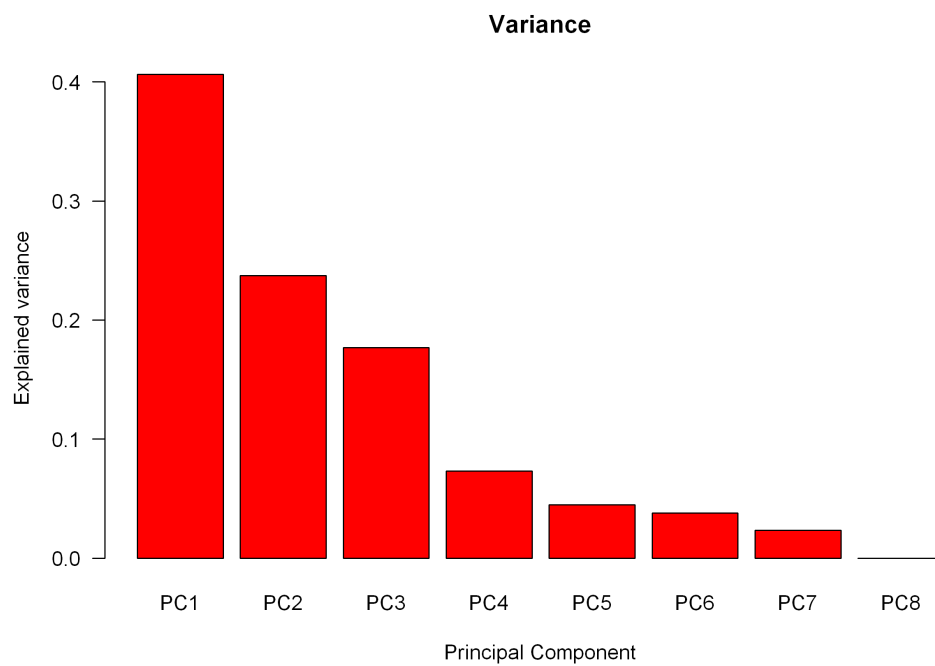


Figure 11: Residual variance plot.

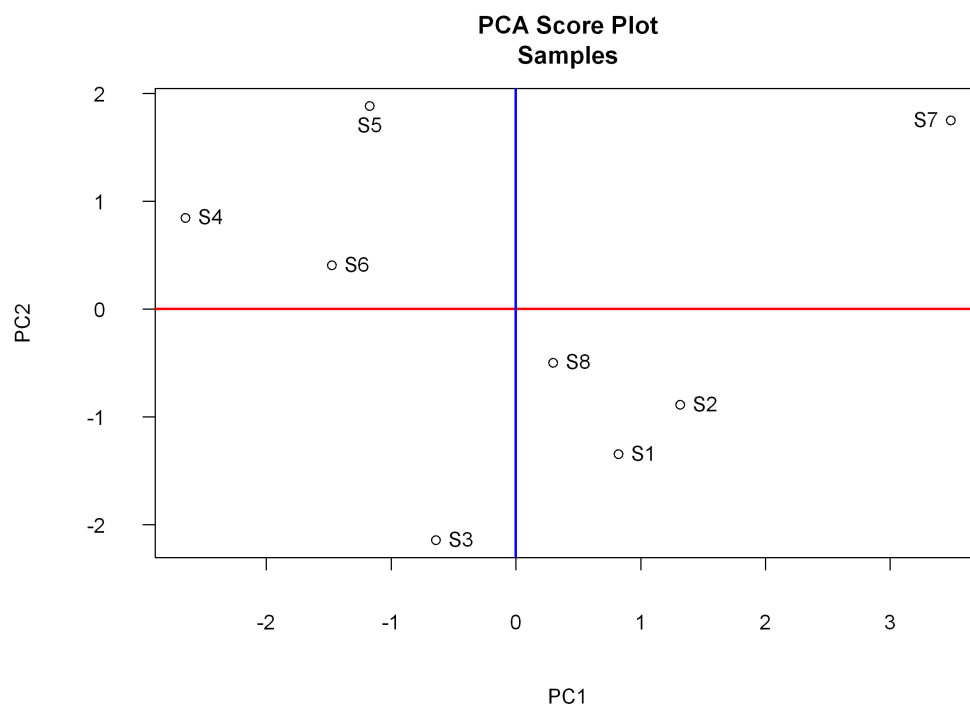


Figure 12: Scores plot with sample names as labels.

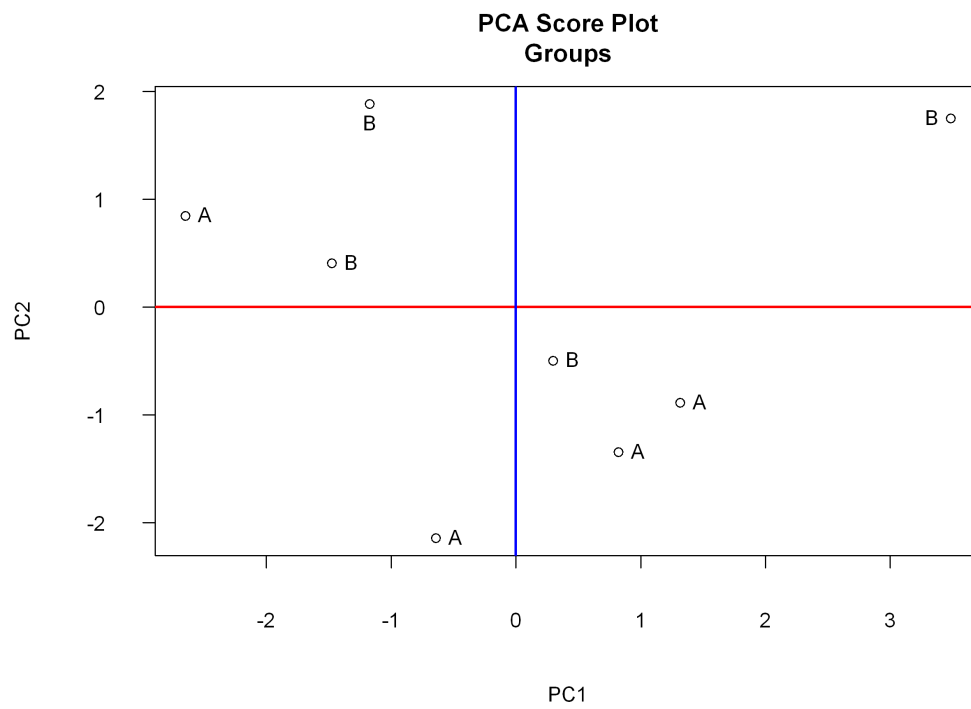


Figure 13: Scores plot with group names as labels.

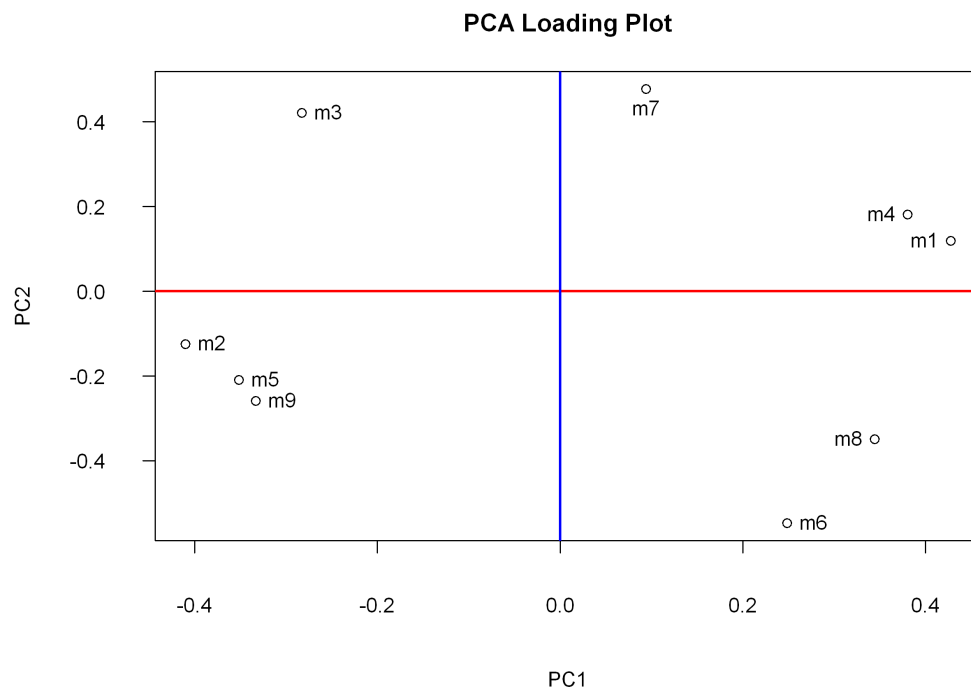


Figure 14: Loadings plot. Indicates which variables contributed to the separation seen in the scores plot.

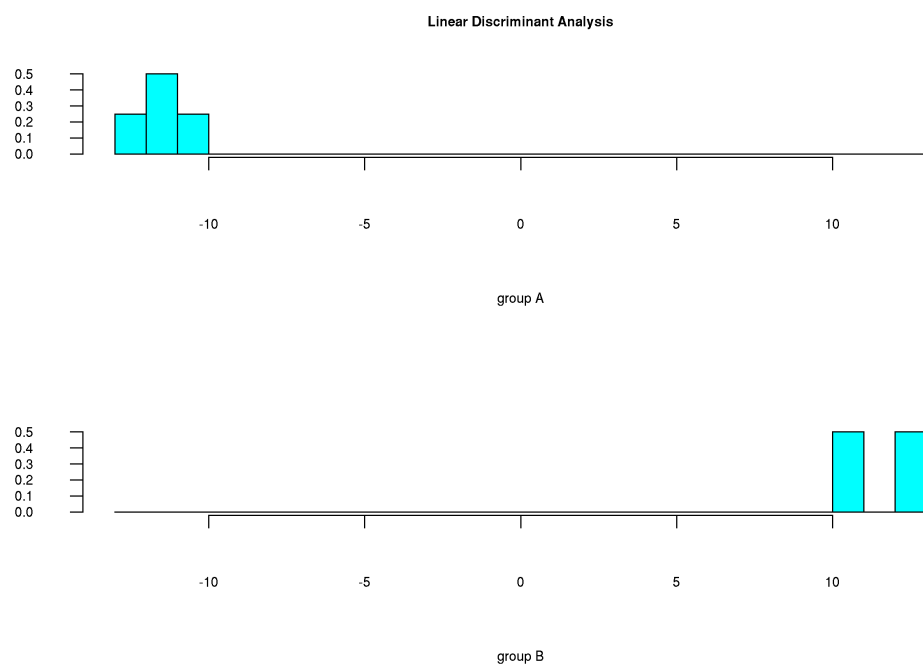


Figure 15: Linear Discriminant Analysis for two groups. This example shows a very clear separation between the groups.

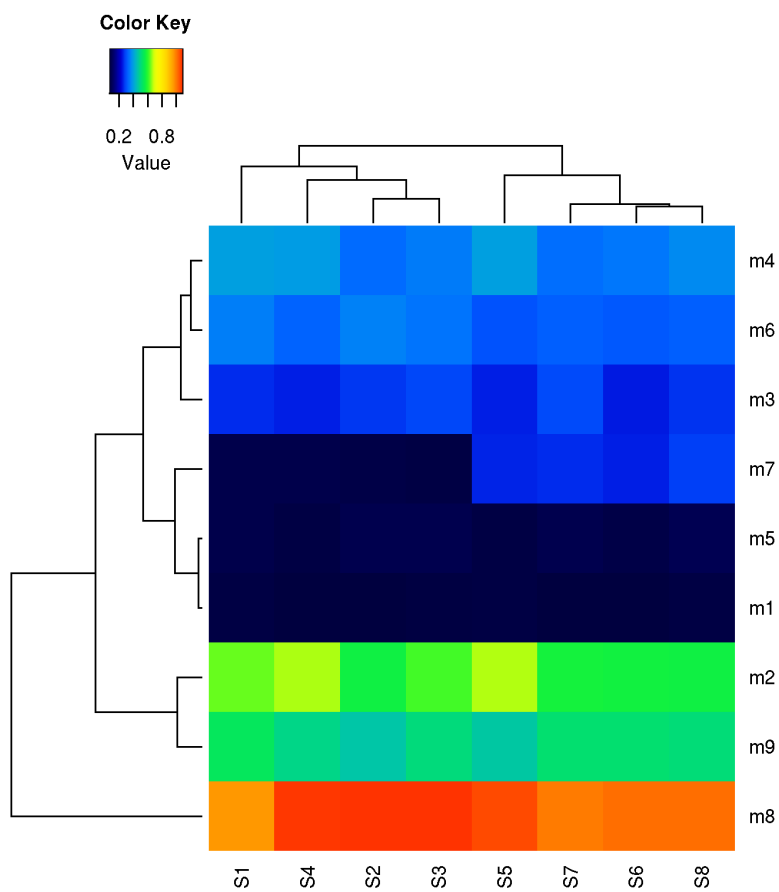


Figure 16: Heatmap showing sample clustering along the x -axis and metabolites along the y -axis. The Color Key shows the value represented by the colours in the body of the plot.