

An R package to process LC/MS metabolomic data: MAIT (Metabolite Automatic Identification Toolkit)

Francesc Fernández-Albert, Rafael Llorach,
Cristina Andrés-Lacueva, Alexandre Perera

September 12, 2024

1 Abstract

Processing metabolomic liquid chromatography and mass spectrometry (LC/MS) data files is time consuming. Currently available R tools allow for only a limited number of processing steps and online tools are hard to use in a programmable fashion. This paper introduces the metabolite automatic identification toolkit MAIT package, which allows users to perform end-to-end LC/MS metabolomic data analysis. The package is especially focused on improving the peak annotation stage and provides tools to validate the statistical results of the analysis. This validation stage consists of a repeated random sub-sampling cross-validation procedure evaluated through the classification ratio of the sample files. MAIT also includes functions that create a set of tables and plots, such as principal component analysis (PCA) score plots, cluster heat maps or boxplots. To identify which metabolites are related to statistically significant features, MAIT includes a metabolite database for a metabolite identification stage.

2 Introduction

Liquid Chromatography and Mass Spectrometry (LC/MS) is an analytical instrument widely used in metabolomics to detect molecules in biological samples. It breaks the molecules down into pieces, some of which are detected as peaks in the mass spectrometer. Metabolic profiling of LC/MS samples basically consists of a peak detection and signal normalisation step, followed by multivariate statistical analysis such as principal components analysis (PCA) and univariate statistical tests such as ANOVA .

As analysing metabolomic data is time consuming, a wide array of software tools are available, including commercial tools such as Analyst® software. There are programmatic R packages, such as XCMS to detect peaks or CAMERA package and AStream , which cover only peak annotation. Another category of free tools available consists of those having online access through a graphical user interface (GUI), such as XCMS Online (<http://xcmsonline.scripps.edu>) or MetaboAnalyst, both extensively used.

These online tools are difficult to use in a programmable fashion. They are also designed and programmed to be used step by step with user intervention, making it difficult to set

up metabolomic data analysis workflow. These R packages involve only a part of the entire metabolomic analysis process. Although there are specific R packages whose objective is peak annotation, this is still an issue in analysing LC/MS metabolomic data.

We introduce a new R package called metabolite automatic identification toolkit (MAIT) for automatic LC/MS analysis. The goal of the MAIT package is to provide an array of tools for programmable metabolomic end-to-end analysis. It consequently has special functions to improve peak annotation through the processes called biotransformations. Specifically, MAIT is designed to look for statistically significant metabolites that separate the classes in the data.

3 Methodology

The main processing steps for metabolomic LC/MS data include the following stages: peak detection, peak annotation and statistical analysis. In the peak detection stage, the objective is to detect the peaks in the LC/MS sample files. The peak annotation stage identifies the metabolites in the metabolomic samples better by increasing the chemical and biological information in the data set. A statistical analysis step is essential to obtain significant sample features. All these 3 steps are covered in the MAIT workflow.

3.1 Peak Detection

Peak detection in metabolomic LC/MS data sets is a complex issue for which several approaches have been developed. Two of the most well established techniques are matched filter and the centWave algorithm . MAIT can use both algorithms through the XCMS package.

3.2 Peak Annotation

The MAIT package uses 3 complementary steps in the peak annotation stage.

- The first annotation step uses a peak correlation distance approach and a retention time window to ascertain which peaks come from the same source metabolite, following the procedure defined in CAMERA package. The peaks within each peak group are annotated following a reference adduct/fragment table and a mass tolerance window.
- The second step uses a mass tolerance window inside the peak groups detected in the first step to look for more specific mass losses called biotransformations. To do this, MAIT uses a predefined biotransformation table where the biotransformations we want to find are saved. A user-defined biotransformation table can be set as an input following the procedure defined in Section (4.6).
- The third annotation step is the metabolite identification stage, in which a predefined metabolite database is mined to search for the significant masses, also using a tolerance window. This database is the Human Metabolome Database (HMDB), 2009/07 version.

3.3 Statistical Analysis

The objective of analysing metabolomic profiling data is to obtain the statistically significant features that contain the highest amount of class-related information. To gather these features, MAIT applies standard univariate statistical tests (ANOVA or Student's t-test) to every feature and selects the significant set of features by setting up a user-defined threshold P-value. Bonferroni multiple test correction can be applied to the resulting P-values. We propose a validation test to quantify how well the data classes are separated by the statistically significant features. The separation is validated through a repeated random sub-sampling cross-validation using partial least squares and discriminant analysis (PLS-DA), support vector machine (SVM) with a radial Kernel and K-nearest neighbours (KNN). Overall and class-related classification ratios are obtained to evaluate the class-related information of the significant features.

4 Using MAIT

The data files for this example are a subset of the data used in reference , which are freely distributed through the XCMS package. In these data there are 2 classes of mice: a group where the fatty acid amide hydrolase gene has been suppressed (class knockout or KO) and a group of wild type mice (class wild type or WT). There are 6 spinal cord samples in each class. In the following, the MAIT package will be used to read and analyse these samples using the main functions discussed in Section ?? . The significant features related to each class will be found using statistical tests and analysed through the different plots that MAIT produces.

4.1 Data Import

Each sample class file should be placed in a directory with the class name. All the class folders should be placed under a directory containing only the folders with the files to be analysed. In this case, 2 classes are present in the data. An example of correct file distribution using the example data files is shown in Figure 1.

4.2 Peak Detection

Once the data is placed in 2 subdirectories of a single folder, the function `sampleProcessing()` is run to detect the peaks, group the peaks across samples, perform the retention time correction and carry out the peak filling process. As function `sampleProcessing()` uses the XCMS package to perform these 4 processing steps, this function exposes XCMS parameters that might be modified to improve the peak detection step. A project name should be defined because all the tables and plots will be saved in a folder using that name. For example, typing `project = "project_Test"`, the output result folder will be `"Results_project_Test"`.

By choosing `"MAIT_Demo"` as the project name, the peak detection stage can be launched by typing:

```
> library(MAIT)
```

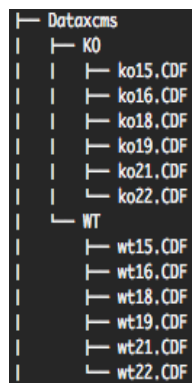


Figure 1: Example of the correct sample distribution for MAIT package use. Each sample file has to be saved under a folder with its class name.

```

> library(faahKO)
> cdfFiles<-system.file("cdf", package="faahKO", mustWork=TRUE)
> MAIT <- sampleProcessing(dataDir = cdfFiles, project = "MAIT_Demo",
+ snThres=2,rtStep=0.03)

```

```

Peak detection done
Retention time correction done
Peak grouping after samples done
Missing Peak integration done

```

After having launched the sampleProcessing function, peaks are detected, they are grouped across samples and their retention time values are corrected. A short summary in the R session can be retrieved by typing the name of the MAIT-class object.

```

> MAIT

A MAIT object built of 12 samples
The object contains 6 samples of class KO
The object contains 6 samples of class WT

```

The result is a MAIT-class object that contains information about the peaks detected, their class names and how many files each class contains. A longer summary of the data is retrieved by performing a summary of a MAIT-class object. In this longer summary version, further information related to the input parameters of the whole analysis is displayed. This functionality is especially useful in terms of traceability of the analysis.

```

> summary(MAIT)

A MAIT object built of 12 samples
The object contains 6 samples of class KO
The object contains 6 samples of class WT

```

Parameters of the analysis:

	Value
dataDir	"/github/workspace/pkglib/faahK0/cdf"
snThres	"2"
Sigma	"2.12332257516562"
mzSlices	"0.3"
retcorrMethod	"loess"
groupMethod	"density"
bwGroup	"3"
mzWidGroup	"0.25"
filterMethod	"centWave"
rtStep	"0.03"
nSlaves	"0"
project	"MAIT_Demo"
ppm	"10"
minfrac	"0.5"
fwhm	"30"
family1	"gaussian"
family2	"symmetric"
span	"0.2"
centWave peakwidth1	"5"
centWave peakwidth2	"20"

4.3 Peak Annotation

The next step in the data processing is the first peak annotation step, which is performed through the `peakAnnotation()`. If the input parameter `adductTable` is not set, then the default MAIT table for positive polarisation will be selected. However, if the `adductTable` parameter is set to "negAdducts", the default MAIT table for negative fragments will be chosen instead. `peakAnnotation` function also creates an output table (see Table ??) containing the peak mass (in charge/mass units), the retention time (in minutes) and the spectral ID number for all the peaks detected. A call of the function `peakAnnotation` may be:

```
> MAIT <- peakAnnotation(MAIT.object = MAIT,corrWithSamp = 0.7,  
+ corrBetSamp = 0.75,perfwhm = 0.6)
```

```
WARNING: No input adduct/fragment table was given. Selecting default MAIT table for posit  
Set adductTable equal to negAdducts to use the default MAIT table for negative polarity  
Start grouping after retention time.  
Created 332 pseudospectra.  
Spectrum build after retention time done  
Generating peak matrix!
```

```

Run isotope peak annotation
% finished: 10 20 30 40 50 60 70 80 90 100
Found isotopes: 104
Isotope annotation done
Start grouping after correlation.
Generating EIC's ..

Calculating peak correlations in 332 Groups...
% finished: 10 20 30 40 50 60 70 80 90 100

Calculating peak correlations across samples.
% finished: 10 20 30 40 50 60 70 80 90 100

Calculating isotope assignments in 332 Groups...
% finished: 10 20 30 40 50 60 70 80 90 100
Calculating graph cross linking in 332 Groups...
% finished: 10 20 30 40 50 60 70 80 90 100
New number of ps-groups: 735
xsAnnotate has now 735 groups, instead of 332
Spectrum number increased after correlation done
Generating peak matrix for peak annotation!
Found and use user-defined ruleset!
Calculating possible adducts in 735 Groups...
% finished: 10 20 30 40 50 60 70 80 90 100
Adduct/fragment annotation done

```

Because the parameter `adductTable` was not set in the `peakAnnotation` call, a warning was shown informing that the default MAIT table for positive polarisation mode was selected. The `xsAnnotated` object that contains all the information related to peaks, spectra and their annotation is stored in the MAIT object. It can be retrieved by typing:

```

> rawData(MAIT)

$xaFA
An "xsAnnotate" object!
With 735 groups (pseudospectra)
With 12 samples and 1261 peaks
Polarity mode is set to: positive
Using automatic sample selection
Annotated isotopes: 104
Annotated adducts & fragments: 72
Memory usage: 4.43 MB

```

4.4 Statistical Analysis

Following the first peak annotation stage, we want to know which features are different between classes. Consequently, we run the function `spectralSigFeatures()`.

```
> MAIT<- spectralSigFeatures(MAIT.object = MAIT,pvalue=0.05,  
+ p.adj="none",scale=FALSE)
```

Skipping peak aggregation step...

```
> summary(MAIT)
```

A MAIT object built of 12 samples and 1261 peaks. No peak aggregation technique has been
56 of these peaks are statistically significant
The object contains 6 samples of class KO
The object contains 6 samples of class WT

Parameters of the analysis:

	Value
dataDir	"/github/workspace/pkglib/faahKO/cdf"
snThres	"2"
Sigma	"2.12332257516562"
mzSlices	"0.3"
retcorrMethod	"loess"
groupMethod	"density"
bwGroup	"3"
mzWidGroup	"0.25"
filterMethod	"centWave"
rtStep	"0.03"
nSlaves	"0"
project	"MAIT_Demo"
ppm	"10"
minfrac	"0.5"
fwhm	"30"
family1	"gaussian"
family2	"symmetric"
span	"0.2"
centWave peakwidth1	"5"
centWave peakwidth2	"20"
corrWithSamp	"0.7"
corrBetSamp	"0.75"
perfwhm	"0.6"
sigma	"6"
peakAnnotation pvalue	"0.05"
calcIso	"TRUE"
calcCiS	"TRUE"
calcCaS	"TRUE"
graphMethod	"hcs"
annotateAdducts	"TRUE"
peakAggregation method	"None"

```

peakAggregation PCAscale          "FALSE"
peakAggregation PCAcenter         "FALSE"
peakAggregation scale             "FALSE"
peakAggregation RemoveOnePeakSpectra "FALSE"
Welch pvalue                      "0.05"
Welch p.adj                       "none"

```

It is worth mentioning that by setting the scale parameter to TRUE, the data will be scaled to have unit variance. A summary of the statistically significant features is created and saved in a table called significantFeatures.csv (see Table ??). It is placed inside the Tables subfolder located in the project folder. This table shows characteristics of the statistically significant features, such as their P-value, the peak annotation or the expression of the peaks across samples. This table can be retrieved at any time from the MAIT-class objects by typing the instruction:

```

> signTable <- sigPeaksTable(MAIT.object = MAIT, printCSVfile = FALSE)

head(signTable)

```

The number of significant features can be retrieved from the MAIT-class object as follows:

```

> MAIT

```

```

A MAIT object built of 12 samples and 1261 peaks. No peak aggregation technique has been
56 of these peaks are statistically significant
The object contains 6 samples of class KO
The object contains 6 samples of class WT

```

4.5 Statistical Plots

Out of 2,402 features, 106 were found to be statistically significant. At this point, several MAIT functions can be used to extract and visualise the results of the analysis. Functions plotBoxplot, plotHeatmap, plotPCA and plotPLS automatically generate boxplots, heat maps and PCA/PLS score plot files in the project folder when they are applied to a MAIT object (see Table ??).

```

> plotBoxplot(MAIT)
> plotHeatmap(MAIT)

> MAIT<-plotPCA(MAIT,plot3d=FALSE)
> MAIT<-plotPLS(MAIT,plot3d=FALSE)
> PLSmodel <- model(MAIT, type = "PLS")
> PCAmode1 <- model(MAIT, type = "PCA")

> PLSmodel

```


Partial Least Squares

12 samples
56 predictors
2 classes: 'KO', 'WT'

No pre-processing

Resampling: Bootstrapped (25 reps)

Summary of sample sizes: 12, 12, 12, 12, 12, 12, ...

Resampling results across tuning parameters:

ncomp	Accuracy	Kappa
1	1	1
2	1	1
3	1	1

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was ncomp = 1.

> *pcaScores(MAIT)*

	PC1	PC2	PC3	PC4	PC5	PC6
[1,]	-8.103931	5.51253909	0.4560729	2.6304313	-0.001166111	-0.14024723
[2,]	-9.018332	-0.22222743	-3.7010516	-2.2074370	-1.944051226	0.65548080
[3,]	-5.511187	0.01698315	1.2817414	-2.5893508	3.427569605	-1.98128060
[4,]	-2.804242	-1.27431307	2.6576958	-0.8357823	0.453653596	3.36170251
[5,]	-2.964179	-4.19515061	0.7141187	2.3295230	-0.351542125	-1.13274686
[6,]	-3.879094	-2.88873255	0.1555561	1.3969922	-0.732631538	-0.30855890
[7,]	5.064055	1.00278256	0.6686535	0.3382944	0.542810579	1.22282372
[8,]	4.665604	0.94689331	1.7533831	-1.3360892	-2.774729880	-1.59218199
[9,]	5.978577	0.83212328	-0.4127802	-0.9160021	-0.177627419	0.26137007
[10,]	5.042800	0.59354948	0.9711509	-0.5728575	-1.200077540	-0.47892310
[11,]	5.587120	-0.03508374	-1.5934350	1.3148452	0.960806146	-0.08086651
[12,]	5.942808	-0.28936346	-2.9511057	0.4474328	1.796985912	0.21342810

	PC7	PC8	PC9	PC10	PC11	PC12
[1,]	-0.18634269	-0.38998060	-0.2168213	-0.06935821	-0.10492904	5.265613e-16
[2,]	-0.42507303	0.44230455	0.2571789	0.23286357	0.22619221	-3.787978e-16
[3,]	0.16476943	0.22085542	0.1370778	0.07101086	0.01383750	-2.809772e-15
[4,]	-0.29989390	0.03301313	-0.9984575	-0.09242844	-0.21597347	-8.913647e-17
[5,]	-2.29389799	-0.25563439	0.2625380	0.19442059	0.14482426	-8.366219e-16
[6,]	3.02775049	-0.34453075	0.4009623	-0.43706048	-0.29884966	-5.022980e-16
[7,]	0.01389960	0.92107983	1.9245152	-0.34297762	1.06501598	8.723169e-16
[8,]	-0.27625241	0.78002973	-0.7304979	-1.19197489	0.04518825	7.430038e-16
[9,]	-0.61402817	-1.31497376	1.1199112	0.04205945	-1.54954265	1.208101e-15
[10,]	0.70310509	-1.12306466	-0.4338590	1.78998490	0.72362460	3.935402e-16

```
[11,] 0.25951393 2.42089320 -0.6828096 0.73329844 -0.65932865 4.679457e-16
[12,] -0.07355034 -1.38999168 -1.0397381 -0.92983818 0.60994068 1.284873e-15
```

The `plotPCA` and `plotPLS` functions produce MAIT objects with the corresponding PCA and PLS models saved inside. The models, loadings and scores can be retrieved from the MAIT objects by using the functions `model`, `loadings` and `scores`:

All the output figures are saved in their corresponding subfolders contained in the project folder. The names of the folders for the boxplots, heat maps and score plots are `Boxplots`, `Heatmaps`, `PCA_Scoreplots` and `PLS_Scoreplots` respectively. Inside the R session, the project folder is recovered by typing:

```
> resultsPath(MAIT)
[1] "/tmp/RtmpRqeZ4S/Rbuild222718d37648/MAIT/vignettes/Results_MAIT_Demo"
```

4.6 Biotransformations

Before identifying the metabolites, peak annotation can be improved using the function `Biotransformations` to make interpreting the results easier. The MAIT package uses a default biotransformations table, but another table can be defined by the user and introduced by using the `bioTable` function input variable. The biotransformations table that MAIT uses is saved inside the file `MAITtables.RData`, under the name `biotransformationsTable`.

```
> Biotransformations(MAIT.object = MAIT, peakPrecision = 0.005)
% Annotation in progress: 10 20 30 40 50 60 70 80 90 100 A MAIT object built o
56 of these peaks are statistically significant
The object contains 6 samples of class KO
The object contains 6 samples of class WT
```

Building a user-defined biotransformations table from the MAIT default table or adding a new biotransformation is straightforward. For example, let's say we want to add a new adduct called `"custom_biotrans"` whose mass loss is 105.

```
> data(MAITtables)
> myBiotransformation<-c("custom_biotrans",105.0)
> myBiotable<-biotransformationsTable
> myBiotable[,1]<-as.character(myBiotable[,1])
> myBiotable<-rbind(myBiotable,myBiotransformation)
> myBiotable[,1]<-as.factor(myBiotable[,1])
> tail(myBiotable)
```

	NAME	MASSDIFF
45	glucuronide conjugation	176.0321
46	hydroxylation + glucuronide	192.027
47	GSH conjugation	305.0682
48	2x glucuronide conjugation	352.0642
49	[C13]	1.0034
50	custom_biotrans	105

To build an entire new biotransformations table, you only need to follow the format of the `biotransformationsTable`, which means writing the name of the biotransformations as factors in the `NAME` field of the data frame and their corresponding mass losses in the `MASSDIFF` field.

4.7 Metabolite Identification

Once the biotransformations annotation step is finished, the significant features have been enriched with a more specific annotation. The annotation procedure performed by the `Biotransformations()` function never replaces the peak annotations already done by other functions. MAIT considers the peak annotations to be complementary; therefore, when new annotations are detected, they are added to the current peak annotation and the identification function may be launched to identify the metabolites corresponding to the statistically significant features in the data.

```
> MAIT <- identifyMetabolites(MAIT.object = MAIT, peakTolerance = 0.005)
```

```
WARNING: No input database table was given. Selecting default MAIT database...
Metabolite identification initiated
```

```
% Metabolite identification in progress: 10 20 30 40 50 60 70 80 90 100
Metabolite identification finished
```

By default, the function `identifyMetabolites()` looks for the peaks of the significant features in the MAIT default metabolite database. The input parameter `peakTolerance` defines the tolerance between the peak and a database compound to be considered a possible match. It is set to 0.005 mass/charge units by default. To check the results easily, function `identifyMetabolites` creates a table containing the significant feature characteristics and the possible metabolite identifications. Such a table is recovered from the MAIT-class object using the instruction:

```
> metTable<-metaboliteTable(MAIT)
> head(metTable)
```

	Query Mass	Database Mass	(neutral mass)	rt	Isotope	Adduct	Name	
1	328.20		Unknown	56.32			Unknown	
2	544.20		Unknown	56.62		[M+Na]+ 521.202	Unknown	
3	481.10		Unknown	61.42			Unknown	
4	570.15		Unknown	53.81			Unknown	
5	396.20		Unknown	64.39			Unknown	
6	449.10		Unknown	54.87			Unknown	
	spectra	Biofluid	ENTRY	p.adj	p	Fisher.Test	Mean Class	KO
1	4	unknown	unknown	0.0020024932	0.0020024932	NA	49713.15	
2	12	unknown	unknown	0.6602191037	0.6602191037	NA	150448.04	
3	429	unknown	unknown	0.0228596524	0.0228596524	NA	32493.60	
4	461	unknown	unknown	0.0219567137	0.0219567137	NA	71808.43	
5	469	unknown	unknown	0.0178739635	0.0178739635	NA	35365.04	


```

Iteration 8 done
Iteration 9 done
Iteration 10 done
Iteration 11 done
Iteration 12 done
Iteration 13 done
Iteration 14 done
Iteration 15 done
Iteration 16 done
Iteration 17 done
Iteration 18 done
Iteration 19 done
Iteration 20 done

```

A summary of a MAIT object, which includes the overall classification values, can be accessed:

```
> summary(MAIT)
```

```

A MAIT object built of 12 samples and 1261 peaks. No peak aggregation technique has been
56 of these peaks are statistically significant
The object contains 6 samples of class K0
The object contains 6 samples of class WT
The Classification using 3 training samples and 20 Iterations gave the results:

```

	KNN	PLSDA	SVM
mean	1	1	1
standard error	0	0	0

Parameters of the analysis:

	Value
dataDir	"/github/workspace/pkglib/faahK0/cdf"
snThres	"2"
Sigma	"2.12332257516562"
mzSlices	"0.3"
retcorrMethod	"loess"
groupMethod	"density"
bwGroup	"3"
mzWidGroup	"0.25"
filterMethod	"centWave"
rtStep	"0.03"
nSlaves	"0"
project	"MAIT_Demo"
ppm	"10"
minfrac	"0.5"

```

fwhm "30"
family1 "gaussian"
family2 "symmetric"
span "0.2"
centWave peakwidth1 "5"
centWave peakwidth2 "20"
corrWithSamp "0.7"
corrBetSamp "0.75"
perfwhm "0.6"
sigma "6"
peakAnnotation pvalue "0.05"
calcIso "TRUE"
calcCiS "TRUE"
calcCaS "TRUE"
graphMethod "hcs"
annotateAdducts "TRUE"
peakAggregation method "None"
peakAggregation PCAscale "FALSE"
peakAggregation PCAcenter "FALSE"
peakAggregation scale "FALSE"
peakAggregation RemoveOnePeakSpectra "FALSE"
Welch pvalue "0.05"
Welch p.adj "none"
peakTolerance "0.005"
polarity "positive"
Validation Iterations "20"
Validation trainSamples "3"
Validation PCAscale "0"
Validation PCAcenter "1"
Validation RemoveOnePeakSpectra "0"
Validation tuneSVM "0"
Validation scale "1"
PCA data logarithm "FALSE"
PCA data centered "TRUE"
PCA data scaled "TRUE"

```

It is also possible to gather the classification ratios per class, classifier used and iteration number by using the function `classifRatioClasses()`:

```

> classifRatioClasses(MAIT)

      KNN_Class_KO  PLSDA_Class_KO  SVM_Class_KO  KNN_Class_WT  PLSDA_Class_WT
[1,]             1                1             1              1              1
[2,]             1                1             1              1              1
[3,]             1                1             1              1              1
[4,]             1                1             1              1              1

```

[5,]	1	1	1	1	1
[6,]	1	1	1	1	1
[7,]	1	1	1	1	1
[8,]	1	1	1	1	1
[9,]	1	1	1	1	1
[10,]	1	1	1	1	1
[11,]	1	1	1	1	1
[12,]	1	1	1	1	1
[13,]	1	1	1	1	1
[14,]	1	1	1	1	1
[15,]	1	1	1	1	1
[16,]	1	1	1	1	1
[17,]	1	1	1	1	1
[18,]	1	1	1	1	1
[19,]	1	1	1	1	1
[20,]	1	1	1	1	1
SVM_Class_WT					
[1,]	1				
[2,]	1				
[3,]	1				
[4,]	1				
[5,]	1				
[6,]	1				
[7,]	1				
[8,]	1				
[9,]	1				
[10,]	1				
[11,]	1				
[12,]	1				
[13,]	1				
[14,]	1				
[15,]	1				
[16,]	1				
[17,]	1				
[18,]	1				
[19,]	1				
[20,]	1				

The classification ratios are 100%; the set of significant features separates the samples belonging to these classes.

4.9 Using External Peak Data

Taking advantage of the modularised design of MAIT, it is possible to use the function MAIT-builder to import peak data and analyse it using the MAIT statistical functions. As stated in

section ??, there are certain arguments that should be provided depending on which function is wanted to be launched. In this section we will show an example of this data importation procedure using the same data that we have been using in the tutorial so far. Let's say we have a peak table recorded in positive polarisation mode with the peak masses and retention time values such as:

```
> peaks <- scores(MAIT)
> masses <- getPeaklist(MAIT)$mz
> rt <- getPeaklist(MAIT)$rt/60
```

We want to perform an annotation stage and metabolite identification on these data. To that end, we can launch the function MAITbuilder to build a MAIT-class object with the data in the table:

```
> importMAIT <- MAITbuilder(data = peaks, masses = masses,
+                           rt = rt, significantFeatures = TRUE,
+                           spectraEstimation = TRUE, rtRange=0.2,
+                           corThresh=0.7)
```

We have selected the option spectraEstimation as TRUE because we do not know the grouping of the peaks into spectra. As we want to annotate and identify all the peaks in the data frame, we set the flag significantFeatures to TRUE. At this point, we can launch the Biotransformations function:

```
> importMAIT <- Biotransformations(MAIT.object = importMAIT,
+                                  adductAnnotation = TRUE,
+                                  peakPrecision = 0.005, adductTable = NULL)
```

Set adductTable equal to negAdducts to use the default MAIT table for negative polarity

```
% Annotation in progress: 0 10 20 30 40 50 60 70 80 90 100
```

We set the adductAnnotation flag to TRUE as we want to perform an adduct annotation step. The parameter adductTable set to NULL implies that a positive polarisation adduct annotation stage will be performed. To run a negative annotation, the argument should be set to negAdducts. The metabolite identification stage is launched as in the previous case:

```
> importMAIT <- identifyMetabolites(MAIT.object = importMAIT,
+                                   peakTolerance=0.005, polarity="positive")
```

```
WARNING: No input database table was given. Selecting default MAIT database...
Metabolite identification initiated
```

```
% Metabolite identification in progress: 0 10 20 30 40 50 60 70 80 90 100
Metabolite identification finished
```