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

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

October 13, 2014

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)
```

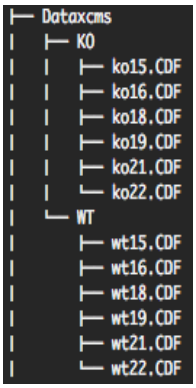

Figure 1: Example of the correct sample distribution for MAIT package use. Each sample file has to be saved under a folder with its class name.

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

215:366 230:680 245:1014 260:1392 275:1766 290:2120 305:2468 320:2804 335:3150 350:3468
215:344 230:662 245:1018 260:1378 275:1728 290:2090 305:2434 320:2722 335:3030 350:3352
215:274 230:544 245:850 260:1186 275:1498 290:1830 305:2162 320:2442 335:2726 350:2976 3
215:224 230:478 245:758 260:1058 275:1388 290:1698 305:1998 320:2302 335:2592 350:2870 3
215:292 230:588 245:914 260:1240 275:1574 290:1872 305:2176 320:2472 335:2792 350:3088 3
215:266 230:512 245:816 260:1114 275:1424 290:1686 305:1972 320:2268 335:2562 350:2850 3
215:348 230:684 245:1016 260:1384 275:1716 290:2100 305:2480 320:2776 335:3118 350:3440
215:324 230:608 245:954 260:1308 275:1638 290:1964 305:2316 320:2618 335:2958 350:3256 3
215:268 230:534 245:822 260:1158 275:1458 290:1760 305:2058 320:2370 335:2654 350:2968 3
215:266 230:514 245:832 260:1170 275:1500 290:1834 305:2136 320:2424 335:2720 350:2982 3
215:316 230:600 245:942 260:1276 275:1620 290:1894 305:2204 320:2492 335:2818 350:3128 3
215:304 230:568 245:872 260:1202 275:1536 290:1838 305:2150 320:2444 335:2758 350:3030 3
Peak detection done
262 325 387 450 512 575
Retention Time Correction Groups: 7
Retention time correction done
262 325 387 450 512 575
Peak grouping after samples done
/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf/KO/ko15.CDF
/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf/KO/ko16.CDF
/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf/KO/ko18.CDF
/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf/KO/ko19.CDF
/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf/KO/ko21.CDF
/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf/KO/ko22.CDF
/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf/WT/wt15.CDF
/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf/WT/wt16.CDF
```

```
/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf/WT/wt18.CDF
/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf/WT/wt19.CDF
/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf/WT/wt21.CDF
/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf/WT/wt22.CDF
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             "/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf"
snThres             "2"
Sigma               "2.12332257516562"
mzSlices            "0.3"
retcorrMethod       "loess"
groupMethod         "density"
bwGroup             "3"
mzWidGroup          "0.25"
filterMethod        "matchedFilter"
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 posi
Set adductTable equal to negAdducts to use the default MAIT table for negative polarity
Start grouping after retention time.
Created 1037 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: 15
Isotope annotation done
Start grouping after correlation.
Generating EIC's ..

Calculating peak correlations in 1037 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 1037 Groups...
 % finished: 10  20  30  40  50  60  70  80  90  100
Calculating graph cross linking in 1037 Groups...
 % finished: 10  20  30  40  50  60  70  80  90  100
New number of ps-groups:  2398
xsAnnotate has now 2398 groups, instead of 1037
```

```
Spectrum number increased after correlation done
Generating peak matrix for peak annotation!
Found and use user-defined ruleset!
Calculating possible adducts in 2398 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)

$xsaFA
An "xsAnnotate" object!
With 2398 groups (pseudospectra)
With 12 samples and 2640 peaks
Polarity mode is set to:  positive
Using automatic sample selection
Annotated isotopes: 15
Annotated adducts & fragments: 16
Memory usage: 9.2 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 2640 peaks. No peak aggregation technique has been
106 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                             "/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf"
snThres                             "2"
```

```
Sigma                                "2.12332257516562"
mzSlices                             "0.3"
retcorrMethod                        "loess"
groupMethod                          "density"
bwGroup                              "3"
mzWidGroup                           "0.25"
filterMethod                         "matchedFilter"
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 2640 peaks. No peak aggregation technique has been
106 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")
> PCAmodel <- model(MAIT, type = "PCA")
```

```
> PLSmodel
```

```
Partial Least Squares

 12 samples
106 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  Accuracy SD  Kappa SD
  1      1         1      0            0
  2      1         1      0            0
  3      1         1      0            0

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,] -8.758728  0.92480221 -6.1406083  0.2742129 -1.9537777  1.32707353
 [2,] -8.348530 -0.86569846  0.1783953 -0.7181633  2.3262804  1.86527975
 [3,] -7.570347  0.32825445 -1.6159867  0.6419965  0.9824302 -2.61644837
 [4,] -6.209758 -0.01281555  3.1104855  0.1755831  0.5704235 -2.12234290
 [5,] -4.632576 -0.80459247  5.6779015 -1.2113600 -1.0185012  1.53485915
 [6,] -5.757966 -0.47710433  0.8561668  0.8644558 -1.6144472  0.03039007
 [7,]  6.483476  7.10158291  0.9827710  2.2904732 -1.8845025 -2.25300205
 [8,]  6.508645  0.44504996 -1.2287543 -6.5380582 -0.9739469 -0.95720989
 [9,]  6.568818  3.66149693 -0.2422269 -1.0526199  3.4833462  3.40476542
[10,]  6.311563 -1.97819990 -0.8625683  3.4660218  3.5531811 -0.38694285
[11,]  7.518147 -5.26076372 -0.8812214 -0.5652639  0.2818974 -2.81704547
[12,]  7.887257 -3.06201203  0.1656458  2.3727220 -3.7523834  2.99062362
             PC7         PC8        PC9        PC10        PC11          PC12
 [1,] -0.94201123 -0.05164849 -0.8474687  2.17078213  1.58640788  1.193490e-15
 [2,]  3.72329244  2.69873472 -1.8907127 -0.51715855 -1.77999111 -4.653396e-16
 [3,]  1.23681723 -0.88123850  3.1775709 -3.25569369  1.34150545 -1.713907e-15
 [4,]  0.04524509 -0.84738531  2.4080804  3.89254447 -1.53783030 -1.186551e-15
 [5,] -1.08614106  0.54573295 -0.9055858 -0.32789136  3.46510005 -9.298118e-16
 [6,] -3.42132405 -2.03957261 -1.9095001 -2.06677976 -2.93497431  5.256212e-16
 [7,]  1.42165727  1.14371311 -1.6573867 -0.06536094  0.08519736  5.467848e-15
 [8,] -1.38904649  2.07408555  1.0680453 -0.37548338 -0.80926550 -1.734723e-15
 [9,]  0.10519927 -3.49752203  0.4989633  0.10649305  0.12756537 -1.096345e-15
[10,] -3.27829656  2.94920817  0.2049729  0.18273224  0.39493774 -3.885781e-16
[11,]  1.90528076 -2.41561220 -2.6916697  0.54595160  0.82921982 -2.151057e-16
[12,]  1.67932734  0.32150464  2.5446908 -0.29013581 -0.76787245 -6.800116e-16
```

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/Rtmp9v8oGq/Rbuild32e41f66a3e8/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
106 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
1      300.2                     Unknown 56.36
2      588.2                     Unknown 46.65
3      537.4                     Unknown 64.41
4      451.2                  450.193634 61.88
5      325.2                     Unknown 60.95
6      395.1                     Unknown 51.19
                  Name spectra      Biofluid      ENTRY       p.adj           p
1              Unknown      27       unknown    unknown 0.017482939 0.017482939
2              Unknown      91       unknown    unknown 0.193607894 0.193607894
3              Unknown    1869       unknown    unknown 0.024657677 0.024657677
4  Geranylgeranyl-PP     1891 Not Available HMDB04486  0.003172073 0.003172073
5              Unknown    1901       unknown    unknown 0.019582285 0.019582285
6              Unknown    1921       unknown    unknown 0.025496645 0.025496645
  Fisher.Test Mean Class KO Mean Class WT Median Class KO Median Class WT KO WT
1          NA  2258350.1365    128461.054    2769931.3564     115642.2922  6  3
2          NA     1998.5050     28919.323          0.0000      10033.2150  2  4
3          NA      521.9275      3261.594          0.0000       3751.3050  1  3
4          NA     8853.1464      1629.177       9644.3125        835.6261  5  0
5          NA     7781.1248     16818.493       7676.3250      17783.4658  5  6
6          NA     1463.7786      6408.485        900.5959       6702.1125  0  4
         ko15        ko16        ko18       ko19      ko21       ko22
1 4005711.400 3115027.656 2726906.080 2812956.63 57169.450 832329.600
2       0.000       0.000       0.000       0.00  2837.345   9153.685
3       0.000       0.000       0.000       0.00     0.000   3131.565
4   10878.315    1943.378   12670.240    9634.14  8338.320   9654.485
5    9563.384    7485.395    3538.465   11418.24  6814.010   7867.255
6       0.000    1801.192    3595.172       0.00  3386.308      0.000
        wt15      wt16      wt18       wt19       wt21       wt22
```

```
1 192385.450 94036.332 48410.145 137248.252 213368.607  85317.540
2  40378.565     0.000     0.000   6696.635  13369.795 113070.941
3   3306.845     0.000  4255.525   1844.086   4195.765   5967.345
4   1671.252  3877.383     0.000      0.000   4226.428      0.000
5  17009.985 18556.947 27223.175   7555.820  11949.359  18615.675
6   4895.743  9045.700 11105.240   5371.080      0.000   8033.145
```

This table provides useful results about the analysis of the samples, such as the P-value of the statistical test, its adduct or isotope annotation and the name of any possible hit in the database. Note that if no metabolite has been found in the database for a certain feature, it is labelled as "unknown" in the table.

4.8 Validation

Finally, we will use the function Validation() to check the predictive value of the significant features. All the information related to the output of the Validation() function is saved in the project directory in a folder called "Validation". Two boxplots showing the overall and per class classification ratios are created, along with every confusion matrix corresponding to each iteration (see Table **??**).

```
>  MAIT <- Validation(Iterations = 20, trainSamples= 3,
+ MAIT.object = MAIT)

Iteration 1 done
Iteration 2 done
Iteration 3 done
Iteration 4 done
Iteration 5 done
Iteration 6 done
Iteration 7 done
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 2640 peaks. No peak aggregation technique has been
106 of these peaks are statistically significant
The object contains 6 samples of class KO
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                             "/home/biocbuild/bbs-3.0-bioc/R/library/faahKO/cdf"
snThres                             "2"
Sigma                               "2.12332257516562"
mzSlices                            "0.3"
retcorrMethod                       "loess"
groupMethod                         "density"
bwGroup                             "3"
mzWidGroup                          "0.25"
filterMethod                        "matchedFilter"
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 MAITbuilder 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
```