Package ‘MAIT’

July 17, 2025

Type Package

Title Statistical Analysis of Metabolomic Data

Version 1.43.0

Date 09-06-2020

Author Francesc Fernandez-Albert,
Rafael Llorach, Cristina
Andres-LaCueva, Alexandre Perera

Maintainer Pol Sola-Santos
`<pol.soladelossantos@gmail.com>`

Description The MAIT package contains functions to perform end-to-end statistical analysis of LC/MS Metabolomic Data. Special emphasis is put on peak annotation and in modular function design of the functions.

biocViews ImmunoOncology, MassSpectrometry, Metabolomics, Software

License GPL-2

LazyLoad yes

Depends R (>= 2.10), CAMERA, Rcpp, pls

Imports gplots,e1071,class,MASS,plsgenomics,agricolae,xcms,methods,caret

Suggests faahKO

Enhances rgl

git_url https://git.bioconductor.org/packages/MAIT

git_branch devel

git_last_commit c489a34

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-07-16

Contents

`annotateBiotransf` *Single Biotransformation Annotator*

Description

Function `annotateBiotransf` annotates a spectrum with an already detected Biotransformation.

Usage

```
annotateBiotransf(biotransf, diffIndex, spectrum, sigPeaksTable,biotransformationsTable)
```

Arguments

`biotransf`	The already detected biotransformation using the `inBetween` function
`diffIndex`	A numeric pointer to the other peak involved in the biotransformation inside the spectrum.
`spectrum`	The spectrum to be annotated.
`sigPeaksTable`	A dataframe obtained from running the `sigPeaksTable` function.
`biotransformationsTable`	Table of biotransformations either read from the bioTable argument or the default MAIT table.

Value

A vector containg the masses of the peaks involved in the biotransformation, their retention time, their annotation and their indices in the signPeaksTable.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`Biotransformations`

`Biotransformations` *Biotransformations of the significant data contained in a MAIT object are identified.*

Description

This function takes a MAIT-class object having significant features already detected and looks up for biotransformations between them. **MAIT** has a default biotransformation table that will be used if no other table is specified via the bioTable input parameter.

Usage

```
Biotransformations(MAIT.object = NULL,
                   peakPrecision = 0.005,
                   bioTable = NULL,
                   adductTable = NULL,
                   adductAnnotation = FALSE)
```

Arguments

`MAIT.object`	A MAIT-class object where significant features have already been found.
`peakPrecision`	Maximum difference between the peak masses differences and the values shown in bioTable to be considered as a biotransformation. As default the value is 0.005 Da.
`bioTable`	Table containing the biotransformations to be looked for in the signData input. By default it is taken the **MAIT-class** biotransformations table.
`adductTable`	Table containing the adducts to be looked for in the signData input. By default it is taken the **MAIT-class** positive adducts table. If this argument is set to "negAdducts", then the default table for negative adducts is taken instead. It is possible to use a user-defined adduct table
`adductAnnotation`	If it is set to TRUE, both adduct and Biotransformations annotation stages are performed.

Value

A MAIT-class object with the updated biotransformations slot

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`spectralAnova` `spectralTStudent`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-Biotransformations(MAIT.object = MAIT, peakPrecision = 0.005)
MAIT@FeatureInfo@biotransformations; #Detected Biotransformations
```

`biotransformationsTable` *biotransformationsTable*

Description

This table contains the biotransformations to be looked up for.

Value

A table having the fields:

- NAME: The name of the biotransformation
- MASSDIFF: The mass difference of the fragment caused by the biotransformation

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`Biotransformations`

`classes` *Class names extractor from a MAIT object*

Description

Function `classes` extracts the class names of a linkMAIT-class object as a vector.

Usage

```
classes(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A character vector containing the class names of the MAIT-class object

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class`

Examples

```
data(MAIT_sample)
MAIT
classes(MAIT)
```

`classifRatioClasses` *Class classification ratio extractor from a MAIT object*

Description

Function `classifRatioClasses` extracts the class classification ratio of a MAIT object as a matrix.

Usage

```
classifRatioClasses(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object where function Validation has already been launched successfully.

Value

A matrix containing the classification ratio for each class, classifier and iteration.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT <- Validation(Iterations = 20, trainSamples= 15, MAIT.object = MAIT)
classifRatioClasses(MAIT)
```

`classNum` *Sample number extractor for each class from a MAIT object*

Description

Function `classNum` extracts the number of samples belonging to each class of a MAIT object as a vector.

Usage

```
classNum(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A numeric vector containing the number of samples for each class of the MAIT object. The order of the classes correspond to that of the output of function classes.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class`

Examples

```
data(MAIT_sample)
MAIT
classNum(MAIT)
```

`Database` *Human Metabolome Database*

Description

The Human Metabolome Database is saved in this dataframe.

Value

A table having the fields:

- ENTRY: HMDB entry
- NAME: The compound name
- FORMULA: The chemical formula of the compound
- MASS: Mass of the fragment
- Biofluid: Where the compound can be found

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`identifyMetabolites` `metaboliteTable`

`featureID` *Feature ID extractor from a MAIT object*

Description

Function `featureID` extracts the feature IDs of a MAIT object as a vector.

Usage

```
featureID(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A numeric vector containing the feature IDs of the MAIT object.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
featureID(MAIT)
```

`featureInfo` *Feature Info extractor from a MAIT object*

Description

Function `featureInfo` extracts the slot MAIT.FeatureInfo of a MAIT object.

Usage

```
featureInfo(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

An object of the class MAIT.FeatureInfo. More Info at MAIT-class.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
featureInfo(MAIT)
```

`featureSigID` *Feature statistically significant ID extractor from a MAIT object*

Description

Function `featureSigID` extracts the vector index of the feature IDs of a MAIT object that have been found significant through function spectralSigFeatures.

Usage

```
featureSigID(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A numeric vector containing the statistically significant feature IDs of the MAIT object.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `spectralSigFeatures`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
featureSigID(MAIT)
featureID(MAIT)[featureSigID(MAIT)] #Significant spectra IDs
```

`FisherLSD` *Performs Fisher's LSD tests on the provided data*

Description

Function `FisherLSD` performs Fisher's LSD tests on the data using the package agricolae.

Usage

```
FisherLSD(data,
          classes,
          index,
          DFerror,
          MSerror,
          numClasses
          )
```

Arguments

`data`	A numerical matrix containing the data
`classes`	A character vector containing the class names of the samples present in the data. This vector must have the same length as the number of samples present in the argument data.
`index`	Numerical value to choose a subset of the data on which the LSD tests is going to be performed.
`DFerror`	Degrees of freedom of the model
`MSerror`	Means square error of the model
`numClasses`	Numerical parameter corresponding to the number of classes present in the data.

Value

A list containing the class names, the group where each class belongs according to the LSD test and the value of their means.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

`getScoresTable` *Returns a list with the peak scores, masses, retention time and other information*

Description

Function getScoresTable takes an `MAIT-class` object and returns a list with the scores of the features in the samples. Additionally, it returns the spectral ID of the peak and (optionally) a table containing the peak information (mass, retention time and annotation).

Usage

```
getScoresTable(MAIT.object = NULL,
               getSpectra = TRUE,
               getExtendedTable = FALSE)
```

Arguments

`MAIT.object`	A MAIT-class object where significant features have already been found.
`getSpectra`	If it is set to TRUE, an element of the returned list will contain the spectra ID of each feature.
`getExtendedTable`	If it is set to TRUE, an element of the returned list will contain a table with peak information (mass, retention time, annotation, intensity per sample)

Value

A list containing:

- scores: The intensity of each feature per sample
- spectraID: A numeric with the correspondence between peaks and spectral ID
- extendedTable: a data frame containg detailed peak information (mass, retention time, annotation, intensity per sample).

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`spectralTStudent` `spectralAnova`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
getScoresTable(MAIT,getExtendedTable=TRUE)
```

`identifyMetabolites` *Metabolite identifier*

Description

Takes a MAIT object and performs the metabolite search for the significant features

Usage

```
identifyMetabolites(MAIT.object=NULL,
      peakTolerance=0.005,
                    database=NULL,
                    polarity="positive",
                    printCSVfile=TRUE)
```

Arguments

`MAIT.object`	A MAIT-class object where significant features have already been found.
`peakTolerance`	Maximum difference between the peak masses differences and the values shown in the database to be considered as a match. As default the value is 0.005 Da.
`database`	User-defined input table. If it is set to NULL, the default **MAIT** database is selected to perform the metabolite identification.
`polarity`	Character parameter that can be set to "positive" or "negative" depending on the polarity in which the samples were taken.
`printCSVfile`	Set to TRUE if an output table has to be produced. The table should be found in (working directory)/Tables/SearchTable.csv.

Value

An output table is stored in the folder (working directory)/Tables/SearchTable.csv if `printCSVfile` is set to TRUE. More info at metaboliteTable

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`Biotransformations` `spectralSigFeatures`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-Biotransformations(MAIT.object = MAIT, peakPrecision = 0.005)
MAIT <- identifyMetabolites(MAIT.object = MAIT, peakTolerance = 0.005,polarity="positive")
```

`inBetween` *Checks if a peak mass value is in a certain mass allowance window.*

Description

Function `inBetween` extracts the mass peaks of a certain spectrum provided a dataframe where the spectrum labels are in a column called pcgroup.

Usage

```
inBetween(testValue,biotRange)
```

Arguments

`testValue`	The peak mass value to be checked
`biotRange`	A matrix containing two numerical columns and each row refers to a certain neutral mass loss. The first column should contain the lower value (neutral mass value minus the peak allowance window) and the second column should have the higher value (neutral mass value plus the peak allowance window)

Value

The rows of the biotRange table where possible neutral losses have been detected.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`Biotransformations`

`loadings` *Loadings extractor for either PCA or PLS models*

Description

Function `loadings` returns the loading vectors for either the PCA, PLS models when functions `plotPCA` or `plotPLS` have been already respectively launched. It also can be used to retrieve the peak aggregation models.

Usage

```
loadings(object, type = "none", ...)
```

Arguments

`object`	A MAIT-class object
`type`	A character whose value should be "PCA" or "PLS" depending on which loading vectors are wanted. If it is set to "none", the peak aggregation models are retrieved.
`...`	Other input

Value

A matrix with the loading vectors.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`plotPCA` or `plotPLS`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
loadings(MAIT)

MAIT<-plotPCA(MAIT,plot3d=FALSE)
loadings(MAIT,type="PCA")

MAIT<-plotPLS(MAIT,plot3d=FALSE)
loadings(MAIT,type="PLS")
```

`LSDResults` *Extractor of the Fisher's LSD tests from a MAIT object*

Description

Function `LSDResults` extracts the results of the LSD tests of a MAIT object as a matrix.

Usage

```
LSDResults(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A matrix containing the results of the Fisher's LSD tests. For each row, equal letters mean that the groups are found to be equal in the test.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `FisherLSD` `spectralSigFeatures`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
LSDResults(MAIT)
```

`MAIT` *MAIT*

Description

A MAIT-class object containing simulated LC/MS data

Value

`MAIT.object`	A MAIT-class object

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

`MAIT-class` *Class* `"MAIT"`

Description

MAIT class objects are used in the MAIT package to perform the analysis and statistical calculus of LC/MS data. It has 5 main slots: FeatureInfo, RawData, Validation, PhenoData and FeatureData

Slots

`FeatureInfo`: Object of class `MAIT.FeatureInfo-class`

`RawData`: Object of class `MAIT.RawData-class`

`Validation`: Object of class `MAIT.Validation-class`

`PhenoData`: Object of class `MAIT.PhenoData-class`

`FeatureData`: Object of class `MAIT.FeatureData-class`

Methods

summary `signature(object = "MAIT")`: This function show a summary of the workflow results performed so far including the classification results and the parameters used.

model `signature(object = "MAIT")`: returns the model for either the PCA, PLS models when functions `plotPCA` or `plotPLS` have been already respectively launched.

scores `signature(object = "MAIT")`: Retrieves the scores from a MAIT object

loadings `signature(object = "MAIT")`: Retrieves the loadings from a MAIT object

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`xsAnnotate`

`MAIT.FeatureData-class`

Class `"MAIT.FeatureData"`

Description

MAIT.FeatureData objects are used in the MAIT package to save the feature data.

Slots

`scores`**:** Here it is saved the dataset obtained after applying the peakAggregation function.

`featureID`**:** The ID numbers of all features are saved here.

`featureSigID`**:** The ID numbers of the significant features are saved here.

`LSDResults`**:** The results of performing a Fisher LSD test on each significant variable are saved in this slot.

`models`**:** The model for each feature used to obtain the scores are saved in this slot.

`pvalues`**:** In this slot are saved the pvalues of the features.

`pvaluesCorrection`**:** The pvalues corrected by multiple test correction are saved here.

`pcaModel`**:** PCA model generated using the function `plotPCA`.

`plsModel`**:** PCA model generated using the function `plotPLS`.

`masses`**:** Masses used as an input for the function `MAITbuilder`.

`rt`**:** Retention time values used as an input for the function `MAITbuilder`.

`extendedTable`**:** Dataframe containing the information regarding masses, retention time values, intensity and spectra IDs passes as an input for the function `MAITbuilder`

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `xsAnnotate`

`MAIT.FeatureInfo-class` *Class* `"MAIT.FeatureInfo"`

Description

MAIT.FeatureInfo objects are used in the MAIT package to save the data related to the information of the features.

Slots

It stores information related to the features. It contains three extra slots:

`biotransformations`**:** Biotransformations found when function Biotransformations is launched.

`peakAgMethod`**:** In this slot is stored the table created by the function identifyMetabolites. It can be retrieved quickly in R by using the function metaboliteTable

`metaboliteTable`**:** Peak Aggregation Method used when function peakAggregation is launched.

Methods

No methods defined with class "MAIT.FeatureInfo" in the signature.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `xsAnnotate`

`MAIT.Parameters-class` *Class* `"MAIT.Parameters"`

Description

This class contains all the parameters used in the MAIT run.

Slots

`sampleProcessing`**:** List containing the parameters of the function sampleProcessing

`peakAnnotation`**:** List containing the parameters of the function peakAnnotation

`peakAggregation`**:** List containing the parameters of the function peakAggregation

`sigFeatures`**:** List containing the parameters of the function spectralSigFeatures

`biotransformations`**:** List containing the parameters of the function Biotransformations

`identifyMetabolites`**:** List containing the parameters of the function identifyMetabolites

`classification`**:** List containing the parameters of the function Validation

`plotPCA`**:** List containing the parameters of the function plotPCA

`plotPLS`**:** List containing the parameters of the function plotPLS

`plotHeatmap`**:** List containing the parameters of the function plotHeatmap

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class`

`MAIT.PhenoData-class` *Class* `"MAIT.PhenoData"`

Description

MAIT.PhenoData objects are used in the MAIT package to save the phenotype data.

Objects from the Class

Objects can be created by calls of the form `new("MAIT.PhenoData", ...)`.

Slots

The information related to the classes present in the data is stored in this slot. It has three different extra slots:

- **`classes`:** It contains the name of the classes in the data. It can be quickly accessed by using the function `classes`
- **`classNum`:** Vector showing the number of samples belonging to each class. It can be quickly accessed by using the function `classNum`
- **`resultsPath`:** In this slot is saved the direction where the project is saved. This means that all the output tables and files of the MAIT object are going to be stored in that directory. It can be quickly accessed by using the function `resultsPath`

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `xsAnnotate`

`MAIT.RawData-class` *Class* `"MAIT.RawData"`

Description

MAIT.RawData objects are used in the MAIT package to save the data related to the information of the features.

Slots

This class contains information of the raw data and the parameters of the whole analysis. It has two slots:

`parameters`**:** All the parameters of the analysis are saved in this slot. It can be obtained as a matrix in R by typing summary(parameters(MAIT.object))

`data`**:** This slot contains either the xcmsSet-class or the xsAnnotate object, depending if the function peakAnnotation has already been launched

Methods

No methods defined with class "MAIT.RawData" in the signature.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `xsAnnotate`

`MAIT.Validation-class` *Class* `"MAIT.Validation"`

Description

MAIT.Validation objects are used in the MAIT package to save the validation results obtained from the classification run.

Slots

The information related to the run of the function Validation is saved here. It contains three lots:

`ovClassifRatioTable`**:** Summary table showing the overall classification ratios for each of the three classifiers. It can be quickly gathered by using the function `ovClassifRatioTable`

`ovClassifRatioData`**:** All the data corresponding to the overall classification ratios. It can be quickly gathered by using the function `ovClassifRatioData`

`classifRatioClasses`**:** All the data corresponding to the classification ratios per class. It can be quickly gathered by using the function `classifRatioClasses`

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `xsAnnotate`

`MAITbuilder` *MAIT constructor function when using external peak data*

Description

Function MAITbuilder creates a MAIT-class object for a given external data. This process allows the user to analyse external peak data through all the MAIT processing steps.

Usage

```
MAITbuilder(data = NULL,
            spectraID = NULL,
            masses = NULL,
            rt = NULL,
            classes = NULL,
            significantFeatures = FALSE,
            spectraEstimation = FALSE,
            rtRange = 0.2,
            corThresh = 0.7)
```

Arguments

`data`	Matrix containing the peak intensity values for each sample. Each row should correspond to a peak and each column to a sample.
`spectraID`	Numeric corresponding to the peak spectral grouping IDs.Two peaks having the same spectraID means that they correspond to the same spectrum.
`masses`	Numeric that contains the masses of the peaks. It should be as long as the number of rows in the argument data.
`rt`	Numeric that contains the retention time of the peaks. It should be as long as the number of rows in the argument data.
`classes`	Character with the class labels for each sample. It should be as long as the number of columns in the argument data.
`significantFeatures`	If it is set to TRUE, all the features set as an input are considered to be significant. Funcions Biotransformations, identifyMetabolites, Validation, plotPCA, plotPLS, plotHeatmap, plotBoxplot are computed on the significant features only. If it is only wanted to perform an annotation process on the external peak data, this flag should be set to TRUE.
`spectraEstimation`	If it is set to TRUE, an estimation of the peak grouping into spectra is performed. This computation is based on a retention time window (set by the argument rtRange) and a correlation threshold (defined by the parameter corThresh).

`rtRange`	Retention time parameter used to build a window to perform an estimation of the peak grouping into spectra.
`corThresh`	Peak correlation value used to define a threshhol to perform an estimation of the peak grouping into spectra.

Value

All the imput values are stored in a new MAIT object.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

Examples

```
data(MAIT_sample)
peaks<-scores(MAIT)
aux<-getScoresTable(MAIT)
masses<-aux$extendedTable$mz
rt <- aux$extendedTable$rt
classFactor <- rep(classes(MAIT),classNum(MAIT))
importMAIT <- MAITbuilder (data=peaks,masses=masses,rt=rt,
significantFeatures=TRUE, spectraEstimation=TRUE, rtRange=0.2,
corThresh=0.7,classes=classFactor)

importMAIT
```

`metaboliteTable` *Metabolite table generator*

Description

Takes a `MAIT-class` object and builds a table with the information related to the significant features and their possible identifications.

Usage

```
metaboliteTable(MAIT.object,
                  printCSVfile = FALSE)
```

Arguments

`MAIT.object`	A MAIT-class object where significant features have already been found.
`printCSVfile`	A boolean parameter. Set to TRUE if a csv file should be written with the metabolite table.

Value

An output table is stored in the folder (working directory)/Tables/SearchTable.csv having the fields:

- First column: search ID number.
- Second column (mz): Peak mass.
- Third column(rt): Peak retention time (in minutes).
- The columns from the third to the column labeled "p.adj" contain number of class samples where the peak has been detected and the intensities of the peak among samples.
- The P.adjust column contains the corrected peak p-value using bonferroni.
- The p column shows the peak p-value with no multiple test correction.
- The Fisher column shows the FIsher test results for the peak. Each of the letters separated by the character "_" corresponds to a class value. Classes having the same letters are indistinguible whereas those having different letters are statistically different clases.
- The isotopes column shows if the peak has been identified as a possible isotope.
- The adduct column shows which kind of adduct or biotransformation could the peak be.
- Column Name contains the name of the possible metabolite identification for the peak.
- The column labeled spectra contains the spectral ID of the peak.
- Column Biofluid shows if the identified search is stored as a biofluid in the input database or not.
- The column ENTRY shows the database name of the entry for the metabolite.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`identifyMetabolites` `spectralAnova` `spectralTStudent`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-Biotransformations(MAIT.object = MAIT, peakPrecision = 0.005)
MAIT <- identifyMetabolites(MAIT.object = MAIT, peakTolerance = 0.005,polarity="positive")
head(metaboliteTable(MAIT))
```

`method` *Peak Aggregation Method Used*

Description

Function `method` returns the name of the peak aggregation method used on a MAIT-class object.

Usage

```
method(object)
```

Arguments

`object`	A MAIT-class object

Value

A character with the peak aggregation method

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`peakAggregation`

Examples

```
data(MAIT_sample)
method(MAIT)
```

`model` *Model extractor for either PCA or PLS models*

Description

Function `model` returns the model for either the PCA, PLS models when functions `plotPCA` or `plotPLS` have been already respectively launched.

Usage

```
model(x,type)
```

Arguments

`x`	A MAIT-class object
`type`	A character whose value should be "PCA" or "PLS" depending on which loading vectors are wanted.

Value

The PCA or PLS model

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`plotPCA` or `plotPLS`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-plotPCA(MAIT,plot3d=FALSE)
model(MAIT,type="PCA")

MAIT<-plotPLS(MAIT,plot3d=FALSE)
model(MAIT,type="PLS")
```

`models` *Model extractor from a MAIT object*

Description

Function `models` extracts the models of a MAIT object as a list.

Usage

```
models(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A list containing the models of the MAIT object.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
models(MAIT)
```

`negAdducts` *Negative adducts table*

Description

This table contains the adducts to be looked up for when the LC/MS polarisation mode was set to Negative. The layout of the table is that of the `CAMERA` adduct table.

Value

A table having the fields:

- ID: An ID number
- name: The adduct name
- nmol: Number of fragments in the adduct
- charge: Electric charge of the adduct
- massdiff: Mass difference in the fragment caused by the adduct
- oidscore: Numeric relating the related clusters of ions
- quasi: Binary value showing the validness of the annotation group
- ips: Four values are possible (0.25,0.5,0.75,1) depending on the likelihood of the rule

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`peakAnnotation`

`ovClassifRatioData` *Overall classification ratio extractor for MAIT objects*

Description

Function `ovClassifRatioData` extracts the overall classification ratio for a MAIT-class object

Usage

```
ovClassifRatioData(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A list containing the overall classification ratio of the MAIT-class object for each classifier.

Author(s)

Francesc Fernandez, <`francesc.fernandez.albert@upc.edu`>

See Also

`MAIT-class` `Validation`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT <- Validation(Iterations = 20, trainSamples= 15, MAIT.object = MAIT)
ovClassifRatioData(MAIT)
```

`ovClassifRatioTable` *Overall classification table extractor for MAIT objects*

Description

Function `ovClassifRatioData` extracts the overall classification table for a `MAIT-class` object

Usage

```
ovClassifRatioTable(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A list containing the overall classification table of the MAIT object for each classifier showing the mean value and their standard error.

Author(s)

Francesc Fernandez, <`francesc.fernandez.albert@upc.edu`>

See Also

`MAIT-class` `Validation`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT <- Validation(Iterations = 20, trainSamples= 15, MAIT.object = MAIT)
ovClassifRatioTable(MAIT)
```

---

parameters *Extractor of the parameters used in the whole run from a MAIT object*

Description

Function `parameters` extracts the slot linkMAIT.Parameters-class of a `MAIT-class` object. This class contains all the parameters that have been used in the previous functions. Typing a summary of this object, a matrix version of the parameters is obtained.

Usage

```
parameters(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

An object of the class MAIT.Parameters.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT <- Validation(Iterations = 20, trainSamples= 15, MAIT.object = MAIT)
parameters(MAIT)
```

---

pcaLoadings *Loadings extractor for the PCA model*

Description

Function `pcaLoadings` returns the loading vectors for the PCA model when function `plotPCA`

Usage

```
pcaLoadings(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A matrix with the PCA loading vectors.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`plotPCA`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-plotPCA(MAIT,plot3d=FALSE)
pcaLoadings(MAIT)
```

`pcaModel` *Model extractor for either PCA*

Description

Function `model` returns the model for PCA when function `plotPCA` have been already respectively launched.

Usage

```
pcaModel(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

The PCA model of the MAIT.object

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`plotPCA`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-plotPCA(MAIT,plot3d=FALSE)
pcaModel(MAIT)
```

`PCAplot3d` *3D PCA scoreplots*

Description

This function takes three sets of coordinates and builds a 3D scoreplot using the package **rgl**

Usage

```
PCAplot3d (z,
           x,
           y,
           cols,
           axes=TRUE,
           new=TRUE)
```

Arguments

`z`	A numerical vector containing the values for the z-axis.
`y`	A numerical vector containing the values for the y-axis.
`x`	A numerical vector containing the values for the x-axis.
`axes`	Boolean parameter. Set to TRUE if axes should be plotted.
`new`	Boolean parameter. Set to TRUE if a new rgl plot should be created.
`cols`	Character vector containing the colors for each sample.

Value

A 3D interactive plot is created using the package **rgl**

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`plotPCA`

`pcaScores` *Scores extractor for the PCA model*

Description

Function `pcaScores` returns the loading vectors for the PCA model when function `plotPCA`

Usage

```
pcaScores(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A matrix with the PCA loading vectors.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`plotPCA`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-plotPCA(MAIT,plot3d=FALSE)
pcaScores(MAIT)
```

`peakAggregation` *Performs a peak aggregation procedure to the rawData of a MAIT object*

Description

peakAggregation function applies a peak aggregation technique to the data of a MAIT-class object. Several aggregation techniques are available (see methods below).

Usage

```
peakAggregation(MAIT.object=NULL,
                method="None",
                clases=NULL,
                samples=NULL,
                PCAscale=FALSE,
                PCAcenter=FALSE,
                scale=FALSE,
                signVariables=NULL,
                RemoveOnePeakSpectra=FALSE,
                printCSVfile=TRUE)
```

Arguments

`MAIT.object`	A MAIT-class object where function peakAnnotation has already been applied. The output of the function is going to be an update of the same MAIT-class object.
`method`	Chosen method to perform the dimensionality reduction using the non-free **pagR** package: - If it is set to "None", no reduction is performed and the spectral peaks are taken as variables. This is the default method. - If it is set to "Mean", the intensity mean value over each sample is taken and used as spectral intensity. - If it is set to "PCA", the first scores vector of a principal components analysis (PCA) decomposition is used as spectral intensity. - If it is set to "NMF", the first scores vector of a non-negative matrix factorization (NMF) is used as spectral intensity. - If it is set to "Single", the spectral peak having the highest intensity mean value over samples among all the spectral peaks is used as spectral intensity.
`clases`	Parameter to explicitly define the clases of the future spectralData object. If it is set to NULL this value is taken from the annotatedPeaks input.
`samples`	If the spectralData object has to include just a subset of the annotatedPeaks' samples, this input must be the vector having the wanted sample's IDs.
`PCAscale`	If method="PCA" and PCAscale is set to TRUE, then the data is scaled following the `prcomp` function. If it is set to TRUE, scale input is ignored.
`PCAcenter`	If method="PCA" and PCAscale is set to TRUE, then the data is centered following the `prcomp` function. If it is set to TRUE, scale input is ignored.
`scale`	If it is set to TRUE, the data is scaled through the spectral mean value. Set to FALSE by default.
`signVariables`	If this input field is a numeric vector, only the spectra/peaks having an ID number present in such vector are used as input data. If it is set to NULL, all the variables are taken into account to build the input data.
`RemoveOnePeakSpectra`	If it is set to TRUE, all the one-peak spectra are deleted from the dataSet and the resulting spectralData object will only contain spectra with more than one peak.
`printCSVfile`	If it is set to TRUE, an output matrix showing the spectral/peak intensity is build, where each column is a sample and each row is a variable (spectra or peak depending on the method used).

Value

An `MAIT-class` object.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

Examples

```
data(MAIT_sample)
peakAggregation(MAIT)
```

`peakAnnotation` *Spectra constructor and peak annotator*

Description

peakAnnotation function performs spectra building and peak annotation using the CAMERA package on a MAIT-class object, after applying the sampleProcessing function. The resultant xsAnnotate object is stored in a MAIT-class object.

Usage

```
peakAnnotation(MAIT.object = NULL,
               corrWithSamp = 0.7,
               perfwhm = 0.6,
               sigma = 6,
               adductTable = NULL,
               printSpectraTable = TRUE,
               corrBetSamp = 0.75,
               pval = 0.05,
               calcIso = TRUE,
               calcCiS = TRUE,
               calcCaS = TRUE,
               graphMethod = "hcs",
               annotateAdducts = TRUE)
```

Arguments

`MAIT.object`	A MAIT-class object where function sampleProcessing has already been applied. The output of the function is going to be an update of the same MAIT-class object.
`corrWithSamp`	Correlation threshold value within samples
`perfwhm`	This parameter is used to group two peaks depending on their retention time. Two peaks are considered to be coeluted if their retention time falls in a range defined as Rt_med +/- FWHM * perfwhm. Where Rt_med is the retention time median and FWHM is the Full Width at Half Maximum. Defined this way, perfwhm is the percentage of the width of the FWHM (Full Width at Half Maximum)
`sigma`	Defining the coelution range as defined in the perfwhm variable, the FWHM is obtained by the expression FWHM=SD*sigma, where SD is calculated considering the peak as normally distributed.

`adductTable`	User-defined input table to annotate the peaks. If it is set to NULL, the default MAIT table for adducts in positive polarization is selected. If its value is "negAdducts", the default MAIT table for fragments in negative polarization is chosen. By default it is set to NULL.
`printSpectraTable`	If it is set to TRUE, a three-column table is build as a csv file, where the first column shows the peak mass, the second column its retention time and the third one shows its spectral ID number. This file is saved under the project directory, in the subfolder named Tables.
`corrBetSamp`	Correlation threshold value between samples
`pval`	See groupCorr function in the **CAMERA** package
`calcIso`	See groupCorr function in the **CAMERA** package
`calcCiS`	See groupCorr function in the **CAMERA** package
`calcCaS`	See groupCorr function in the **CAMERA** package
`graphMethod`	See groupCorr function in the **CAMERA** package
`annotateAdducts`	If it is set to TRUE, the function will perform an adduct annotation stage.

Value

A MAIT-class object containing the xsAnnotate-class in the rawData slot.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`xsAnnotate`,`xsAnnotate-class`

Examples

```
#Provided that the data files are saved accordingly
#in subfolders under a folder named "data" (see vignette):
#MAIT<-sampleProcessing(dataDir = "data", project = "Results", snThres=2,rtStep=0.02)
#MAIT<-peakAnnotation(MAIT.object = MAIT,corrWithSamp = 0.7, corrBetSamp = 0.7,perfwhm = 0.6)
```

`plotBoxplot` *Prints a png file for each of the significant peak/spectra present in the input*

Description

This function takes a MAIT-class object containing information related to the significant features and plots a boxplot for each significant feature (peak or spectra).

Usage

```
plotBoxplot(MAIT.object=NULL)
```

Arguments

`MAIT.object`	A MAIT-class object where significant features have already been found.

Value

A boxplot is stored as a png file for each of the significant features (peak or spectra). The files will be stored in the directory (working directory)/Boxplots

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`spectralAnova` `spectralTStudent`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-plotBoxplot(MAIT)
```

`plotHeatmap` *Builds ten heatmaps with different p-values and clustering distances*

Description

This function takes a MAIT object containing information of the significant features in the data and plots 10 heatmaps. 5 diferent p-values (0.05, 0.01, 0.001, 1e-4 and 1e-5) and two clustering distances (euclidean and pearson) are used.

Usage

```
plotHeatmap(MAIT.object=NULL)
```

Arguments

`MAIT.object`	A MAIT-class object where significant features have already been found.

Value

10 different heatmaps using 5 p-values (0.05, 0.01, 0.001, 1e-4 and 1e-5) and two clustering distances (euclidean and pearson) are created. The plots will be stored as png files in a folder called (working directory)/Heatmaps

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`spectralAnova` `spectralTStudent`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-plotHeatmap(MAIT)
```

`plotPCA` *2D and 3D PCA scoreplots from a MAIT object*

Description

This function takes a MAIT-class object containing information of the significant features in the data and performs 2D scoreplots (PC1 vs PC2, PC2 vs PC3 and PC1 vs PC3) saved as png files. Additionally it also performs an interactive 3D PCA scoreplot.

Usage

```
plotPCA (MAIT.object=NULL,
         Log=FALSE,
  center=TRUE,
  scale=TRUE,
  plot3d=TRUE)
```

Arguments

`MAIT.object`	A MAIT-class object where significant features have already been found.
`Log`	Set to TRUE if the data should be plotted using the logarithm of the intensity.
`center`	Set to TRUE if the data should be centered around its mean. See `scale`.
`scale`	Set to TRUE if the data should be scaled. See `scale`.
`plot3d`	Boolean set to TRUE if a 3D PCA scoreplot should be plot.

Value

Three different PCA scoreplots are printed in three png files. One using PC1 vs PC2, another with PC1 vs PC3 and the last one with PC2 vs PC3. The files will be stored in the directory (working directory)/PCA_Scoreplots. Moreover, an interactive 3D PCA scoreplot is also generated through function PCAplot3d.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`spectralAnova` `spectralTStudent`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-plotPCA(MAIT,plot3d=FALSE)
```

`plotPLS` *2D and 3D PLS scoreplots from a MAIT object*

Description

This function takes a MAIT-class object containing information of the significant features in the data and performs 2D scoreplots (PC1 vs PC2, PC2 vs PC3 and PC1 vs PC3) saved as png files. Additionally it also performs an interactive 3D PLS scoreplot.

Usage

```
plotPLS (MAIT.object=NULL,
         Log=FALSE,
  center=TRUE,
  scale=TRUE,
  plot3d=TRUE)
```

Arguments

`MAIT.object`	A MAIT-class object where significant features have already been found.
`Log`	Set to TRUE if the data should be plotted using the logarithm of the intensity.
`center`	Set to TRUE if the data should be centered around its mean. See `scale`.
`scale`	Set to TRUE if the data should be scaled. See `scale`.
`plot3d`	Boolean set to TRUE if a 3D PCA scoreplot should be plot.

Value

If the number of components in the PLS is found to be three or more, three different PLS scoreplots are printed in three png files. One using PC1 vs PC2, another with PC1 vs PC3 and the last one with PC2 vs PC3. If the number of components is less than three, all the possible plots of these three are created. The files will be stored in the directory (working directory)/PLS_Scoreplots. Moreover, an interactive 3D PLS scoreplot is also generated through function PCAplot3d.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`spectralAnova` `spectralTStudent`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-plotPLS(MAIT,plot3d=FALSE)
```

`PLSDA` *Applies PLSDA to the provided data*

Description

Function `PLSDA` performs Fisher's LSD tests on the data using the package plsgenomics

Usage

```
PLSDA(Xtrain,
      Ytrain,
      Xtest = NULL,
      ncomp,
      nruncv = 0,
      alpha = 2/3,
      priors = NULL)
```

Arguments

`Xtrain`	A numerical matrix containing the data
`Ytrain`	A factor vector containing the class labels of the samples
`Xtest`	A numerical matrix containing the data whose class is to be predicted.
`ncomp`	Number of components to build the PCA model
`nruncv`	Number of cross-validation iterations to be performed for the choice of the number of latent components
`alpha`	The proportion of Obesrvations to be included in the training set at each cross-validation iteration
`priors`	The class priors to be used for linear discriminant analysis. If unspecified, the class proportions in the training set are used.

Value

A list containing the output of function pls.regression, the predicted class for the Xtest dataset and the number of components used.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

`plsLoadings` *Loadings extractor for the PLS model*

Description

Function `plsLoadings` returns the loading vectors for the PLS model when function `plotPLS`

Usage

```
plsLoadings(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A matrix with the PLS loading vectors.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`plotPLS`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-plotPLS(MAIT,plot3d=FALSE)
plsLoadings(MAIT)
```

`plsModel` *Model extractor for either PLS*

Description

Function `model` returns the model for PLS when function `plotPLS` have been already respectively launched.

Usage

```
plsModel(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

The PLS model of the MAIT.object

Author(s)

Francesc Fernandez, <`francesc.fernandez.albert@upc.edu`>

See Also

`plotPLS`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-plotPLS(MAIT,plot3d=FALSE)
plsModel(MAIT)
```

`plsScores` *Scores extractor for the PLS model*

Description

Function `plsScores` returns the scores vectors for the PLS model when function `plotPLS`

Usage

```
plsScores(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A matrix with the PLS loading vectors.

Author(s)

Francesc Fernandez, <`francesc.fernandez.albert@upc.edu`>

See Also

`plotPLS`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT<-plotPLS(MAIT,plot3d=FALSE)
plsScores(MAIT)
```

`posAdducts` *Positive adducts table*

Description

This table contains the adducts to be looked up for when the LC/MS polarisation mode was set to positive. The layout of the table is that of the `CAMERA` adduct table.

Value

A table having the fields:

- ID: An ID number
- name: The adduct name
- nmol: Number of fragments in the adduct
- charge: Electric charge of the adduct
- massdiff: Mass difference in the fragment caused by the adduct
- oidscore: Numeric relating the related clusters of ions
- quasi: Binary value showing the validness of the annotation group
- ips: Four values are possible (0.25,0.5,0.75,1) depending on the likelihood of the rule

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`peakAnnotation`

`project` *Change the basis of the MAIT data*

Description

Function `project` is used to project the data of a MAIT object to the subspace of the models generated by another MAIT object.

Usage

```
project(modelData,
        projectData)
```

Arguments

`modelData`	The `MAIT-class` object where the models to which the new data is to be projected are saved.
`projectData`	The `MAIT-class` containing the data to be projected.

Value

A matrix containing the data contained in the `projectData` parameter already projected into the `modelData` model subspace.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

`pvalues` *Pvalues extractor from a MAIT object*

Description

Function `pvalues` extracts the pvalues contained in a `MAIT-class` object.

Usage

```
pvalues(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A numeric vector containing the pvalues of a MAIT-class object.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `spectralSigFeatures`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
pvalues(MAIT)
```

`pvaluesCorrection` *P-values correction extractor from a MAIT object*

Description

Function `pvaluesCorrection` returns a character showing wheter some multiple testing correction has been perfomed on the p-values.

Usage

```
pvaluesCorrection(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

The output is a character whose values could be "None" if no p-value correction has been performed or "Bonferroni" if Bonferroni multiple test correction was selected when function spectralSigFeatures was applied.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `spectralSigFeatures`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
pvaluesCorrection(MAIT)
```

`rawData` *Raw data extractor from a MAIT object*

Description

Function `rawData` extracts the raw data used to build the `MAIT-class` object

Usage

```
rawData(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A list containing either a xcmsSet or a xsAnnotate object.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`xsAnnotate-class`

Examples

```
data(MAIT_sample)
MAIT
rawData(MAIT)
```

`removeOnePeakSpectra` *Removes those spectra having just one peak*

Description

Function `removeOnePeakSpectra` removes the spectra having just one peak

Usage

```
removeOnePeakSpectra(data,
      idGroup)
```

Arguments

`data`	A numerical matrix containing the peak data
`idGroup`	A numeric vector containing the spectra id number of the peaks

Value

A peak data set without the one-peak spectra.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`peakAggregation`

`resultsPath` *Retrieves the folder where the results are saved for a MAIT object*

Description

Function `resultsPath` returns the folder where the plots and tables are saved for a `MAIT-class` object

Usage

```
resultsPath(MAIT.object)
```

Arguments

`MAIT.object`	A MAIT-class object

Value

A character showing where the plots and tables have been stored.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
resultsPath(MAIT)
```

`retrieveSpectrum` *Extractor of the mass peaks corresponding to a certain spectrum*

Description

Function `retrieveSpectrum` extracts the mass peaks of a certain spectrum provided a dataframe where the spectrum labels are in a column called pcgroup.

Usage

```
retrieveSpectrum(spectrumNumber,sigPeaksTable)
```

Arguments

`spectrumNumber`	The spectrum ID number whose peaks we want to retrieve.
`sigPeaksTable`	A dataframe containing the peak data in rows. There should be a column called pcgroup containing the spectra correspondence for all the peaks and the firts column should contain the peak masses.

Value

A numeric vector containing the peak masses of the queried spectrum.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`Biotransformations`

`sampleProcessing` *Peak detector of netCDF samples using xcms package*

Description

sampleProcessing takes a set of netCDF files containing LC/MS sample data and performs a peak detection, retention time correction and peak grouping steps using the package xcms. A MAIT-class object is created and all the informated is saved in it.

Usage

```
sampleProcessing(dataDir = NULL,
                 snThres = 5,
                 Sigma = 5/2.3548,
   mzSlices = 0.3,
   retcorrMethod = "loess",
   groupMethod = "density",
   bwGroup = 3,
   mzWidGroup = 0.25,
   filterMethod = "centWave",
   prefilter = c(3,3000),
     rtStep = 0.03,
   nSlaves = 0,
   minfrac = 0.5,
                 minsamp = 1,
   peakwidth = c(5, 20),
   project = NULL,
   ppm = 10,
                 family = c("gaussian", "symmetric"),
                 span = 0.2,
                 fwhm = 30)
```

Arguments

`dataDir`	Folder where the netCDF files are stored. The samples files must be classified in subdirectories according to their classes.
`snThres`	Signal to noise ratio. Setting a high value of this parameter will lead to a higher number of features although they will be more noisy.
`Sigma`	Standard deviation (width) of matched filtration model peak.
`mzSlices`	Minimum difference in m/z for peaks with overlapping retention times.
`retcorrMethod`	Method used to correct the retention times values of the variables.
`groupMethod`	Method used to build the group peaks of variables.
`bwGroup`	Bandwidth (standard deviation or half width at half maximum) of gaussian smoothing kernel to apply to the peak density chromatogram.
`mzWidGroup`	Width of overlapping m/z slices to use for creating peak density chromatograms and grouping peaks across samples.
`filterMethod`	Filtering method applied in the peak detection step.
`prefilter`	c(k, I)specifying the prefilter step for the first analysis step(ROI detection). Mass traces are only retained if they contain at least k peakswith intensity>= I.
`rtStep`	Step size to use for profile generation.
`nSlaves`	Number of slaves for parallel calculus.
`project`	Project folder name under which the results will be saved. This folder will be created in the working directory.
`minfrac`	minimum fraction of samples necessary in at least one of the sample groups for it to be a valid group. See group.density in package **xcms** for details.
`minsamp`	minimum number of samples necessary in at least one of the sample groups for it to be a valid group. See group.density in package **xcms** for details.
`ppm`	maxmial tolerated m/z deviation in consecutive scans, in ppm (parts per million). See findPeaks.centWave in package **xcms** for details.
`peakwidth`	Chromatographic peak width, given as range (min,max) in seconds.
`fwhm`	See fwhm argument in xcmsSet function.
`span`	See span argument in xcmsSet function.
`family`	See family argument in xcmsSet function.

Value

A MAIT-class object containing the data of the netCDF files. The xcmsSet-class object can be retrieved using the function rawData.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

Examples

```
#Provided that the data files are saved accordingly
#in subfolders under a folder named "data" (see vignette):
#MAIT<-sampleProcessing(dataDir = "data", project = "Results", snThres=2,rtStep=0.02)
```

`scores` *Retrieves the scores from a MAIT object*

Description

Function `scores` extracts the scores MAIT object

Usage

```
scores(object,
       type,...)
```

Arguments

`object`	A MAIT-class object
`type`	If it is set to "none", the peak aggregation models are returned. If it is set to "PCA", the PCA model is returned. If it is set to "PLS", the PCA model is returned.
`...`	Other input

Value

A numeric matrix containing the scores saved in the MAIT object

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
scores(MAIT)

MAIT<-plotPCA(MAIT,plot3d=FALSE)
scores(MAIT,type="PCA")

MAIT<-plotPLS(MAIT,plot3d=FALSE)
scores(MAIT,type="PLS")
```

`SearchCand` *Peak search function into a database*

Description

Function `SearchCand` looks up for a peak into a database

Usage

```
SearchCand(candidate,
           dataBase,
           peakTolerance)
```

Arguments

`candidate`	The mass of the peak to be looked up into the database
`dataBase`	The table where the database to be used is saved. The function is build to use databases with the same layout as the MAIT's database. This database can be accessed by typing data(MAITtables) and Database.
`peakTolerance`	Maximum difference between the peak masses differences and the values shown in the database to be considered as a match.

Value

A matrix containing all the possible hits for that peak candidate

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `identifyMetabolites`

`selectK` *Looks for the optimum number of nearest neighbours to be considered for the KNN*

Description

Function `selectK` finds the optimum number of nearest neighbours for the K-Nearest Neighbours (KNN) algorithm.

Usage

```
selectK(data,
         class,
         max.k)
```

Arguments

`data`	A numerical matrix containing the data
`class`	Vector containing the class label of each sample.
`max.k`	Maximum number of nearest neighbours to be considered.

Value

A numeric value of the optimal number of neighbours to be considered in a KNN run.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`Validation`

`selectPLScomp` *Looks for the optimum number of components to be considered for the PLSDA*

Description

Function `selectPLScomp` finds the optimum number of components to be used by the Partial Least Squares and linear Discriminant Algorithm (PLSDA).

Usage

```
selectPLScomp(data,
         class,
         max.comp)
```

Arguments

`data`	A numerical matrix containing the data
`class`	Vector containing the class label of each sample.
`max.comp`	Maximum number of components to be considered.

Value

A numeric value of the optimal number of components to be considered in a PLSDA run.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`Validation`

`sigPeaksTable` *Build a table of the information related to the significant features contained in a MAIT object*

Description

Function sigPeaksTable takes an `MAIT-class` object containing significant feature information and builds a table with the information related to these features.

Usage

```
sigPeaksTable(MAIT.object=NULL,
              printCSVfile=FALSE,
              extendedTable = TRUE,
              printAnnotation=TRUE)
```

Arguments

`MAIT.object`	A MAIT-class object where significant features have already been found.
`printCSVfile`	Set to TRUE if an output table has to be produced. The table should be found in (working directory)/(project directory)Tables/significativeFeatures.csv.
`extendedTable`	Set to TRUE the table created by the peak external data is used.
`printAnnotation`	Set to TRUE The peak annotation is provided in the output table

Value

A table containing:

- First column (mz): Peak mass
- Second column(mzmin): Minimum peak mass of the peak group.
- Third column(mzmax): Maximum peak mass of the peak group.
- Fourth column(rt): Peak retention time (in minutes).
- Fifth column(rtmin): Minimum peak retention time of the peak group.
- Sixth column(rtmax): Maximum peak retention time of the peak group.
- Seventh column(npeaks): Number of samples where the peak has been detected.
- The columns from the nineth to the column labeled "isotopes" contain number of class samples where the peak has been detected and the intensities of the peak among samples.
- The isotopes column shows if the peak has been identified as a possible isotope.
- The adduct column shows which kind of adduct could the peak be.
- The column labeled pcgroup contains the spectral ID of the peak.
- The P.adjust column contains the corrected peak p-value using post-hoc methods.
- The p column shows the peak p-value with no multiple test correction.
- The Fisher column shows the Fisher test results for the peak. Each of the letters separated by the character "_" corresponds to a class value. Classes having the same letters are indistinguible whereas those having different letters are statistically different clases.
- The last columns contain the mean and median values for each feature

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`spectralTStudent` `spectralAnova`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
head(sigPeaksTable(MAIT))
```

`spectralAnova` *Extract significant features from a MAIT object*

Description

Function spectralAnova takes an `MAIT-class` object and obtains which of the variables are significant given a p-value threshold. The parameters of the significant features can ve printed to an output table (TRUE by default).

Usage

```
spectralAnova(pvalue = 0.05,
              p.adj="none",
              MAIT.object = NULL,
              printCSVfile = TRUE)
```

Arguments

`MAIT.object`	A MAIT-class object where function peakAggregation has already been applied. The output of the function is going to be an update of the same MAIT-class object.
`pvalue`	P-value threshold. Variables having a p-value lower than this value is considered as a significant variable.
`p.adj`	Post-hoc method to be used to correct the p-values.
`printCSVfile`	Set to TRUE if an output table has to be produced. See function sigPeaksTable for more information.

Value

A MAIT-class object containing the significant features of the scores slot of MAIT-class object used as an input.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `peakAggregation` `sigPeaksTable`

`spectralFUN` *Extract significant features from a MAIT object using a user-defined test*

Description

Function spectralFUN takes an `MAIT-class` object and obtains which of the variables are significant given a p-value threshold following a user-defined statistical test. The parameters of the significant features can ve printed to an output table (TRUE by default).

Usage

```
spectralFUN(pvalue=0.05,
            p.adj="none",
            MAIT.object=NULL,
            printCSVfile=TRUE,
            test.fun=NULL,
            namefun=NULL)
```

Arguments

`pvalue`	P-value threshold. Variables having a p-value lower than this value is considered as a significant variable.
`p.adj`	Post-hoc method to be used to correct the p-values.
`MAIT.object`	A MAIT-class object where function peakAggregation has already been applied. The output of the function is going to be an update of the same MAIT-class object.
`printCSVfile`	Set to TRUE if an output table has to be produced. See function sigPeaksTable for more information.
`test.fun`	Function containing the statistical test to be applied on each feature. The function should be designed to correct just one feature as the function will aplly this correction to all the features in the MAIT.object.
`namefun`	Character with the name of the test. This name will appear in the MAITparameters table and in the summary of the MAIT object.

Value

A MAIT-class object containing the significant features of the scores slot of MAIT-class object used as an input.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `peakAggregation` `sigPeaksTable`

`spectralKruskal` *Extract significant features from a MAIT object*

Description

Function spectralKruskal takes an `MAIT-class` object and obtains which of the variables are significant given a p-value threshold following a Kruskal-Wallis test. The parameters of the significant features can ve printed to an output table (TRUE by default).

Usage

```
spectralKruskal(pvalue = 0.05,
              p.adj="none",
              MAIT.object = NULL,
              printCSVfile = TRUE)
```

Arguments

`MAIT.object`	A MAIT-class object where function peakAggregation has already been applied. The output of the function is going to be an update of the same MAIT-class object.
`pvalue`	P-value threshold. Variables having a p-value lower than this value is considered as a significant variable.
`p.adj`	Post-hoc method to be used to correct the p-values.
`printCSVfile`	Set to TRUE if an output table has to be produced. See function sigPeaksTable for more information.

Value

A MAIT-class object containing the significant features of the scores slot of MAIT-class object used as an input.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `peakAggregation` `sigPeaksTable`

`spectralSigFeatures` *Extract significant features from a MAIT object*

Description

Function spectralSigFeatures takes a `MAIT-class` object and obtains which of the variables are significant given a p-value threshold. The parameters of the significant features can ve printed to an output table (TRUE by default). Depending on the number of classes in the data, the function chooses between using ANOVA tests through function spectralAnova, or T-Student tests by using function spectralTStudent.

Usage

```
spectralSigFeatures(MAIT.object = NULL,
                    pvalue = 0.05,
                    p.adj = "none",
                    printCSVfile = FALSE,
                    scale = FALSE,
                    parametric = TRUE,
                    var.equal = FALSE,
                    test.fun = NULL,
                    jitter = FALSE,
                    jitter.factor = 1,
                    jitter.amount = 0,
                    namefun = NULL)
```

Arguments

`MAIT.object`	A MAIT-class object where function peakAggregation has already been applied. The output of the function is going to be an update of the same MAIT-class object.
`pvalue`	P-value threshold. Variables having a p-value lower than this value is considered as a significant variable.
`p.adj`	Character with the name of the posthoc method to be applied to correct the pvalues. The supported methods are that of the p.adjust function
`printCSVfile`	Set to TRUE if an output table has to be produced. See function sigPeaksTable for more information.
`scale`	Set to FALSE by default. When set to TRUE, a unit variance scaling of the data when no peak aggregation is performed. If a peak aggregation method is applied, this parameter is ignored.
`parametric`	If it is set to TRUE, the statistical tests to be applied will be parametrical tests (e.g. ANOVA, TStudent or Welch's tests). Non-parametrical tests (e.g. Kruskal-Wallis, Mann-Whitney tests) are applied otherwise.
`var.equal`	Set to FALSE by default. When set to TRUE, a Student's T-Test is applyied when having 2 classes in the data. If it is set to FALSE, a Welch's test is applyied instead.
`test.fun`	Function of the user-defined posthoc method to be applyied.
`jitter`	If it is set to TRUE, a jitter noise is added to the data. This is useful when applying Mann-Whitney tests with ties.

`jitter.factor`	See argument factor of the function jitter.
`jitter.amount`	See argument amount of the function jitter.
`namefun`	Name of the user-defined posthoc test in the argument test.fun.

Value

A MAIT-class object containing the significant features of the scores slot of MAIT-class object used as an input.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `spectralTStudent` `spectralAnova` `sigPeaksTable`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
```

`spectralTStudent` *Extract significant features from a MAIT object for two classes*

Description

Function spectralTStudent takes a `MAIT-class` object and obtains which of the variables are significant given a p-value threshold when there only are two classes in the raw data. The parameters of the significant features can be printed to an output table (TRUE by default).

Usage

```
spectralTStudent(MAIT.object = NULL,
   pvalue = 0.05,
   p.adj = "none",
                 printCSVfile = TRUE)
```

Arguments

`MAIT.object`	A MAIT-class object where function peakAggregation has already been applied. The output of the function is going to be an update of the same MAIT-class object.
`pvalue`	P-value threshold. Variables having a p-value lower than this value is considered as a significant variable.
`p.adj`	Character with the name of the posthoc method to be applied to correct the pvalues. The supported methods are that of the p.adjust function
`printCSVfile`	Set to TRUE if an output table has to be produced. See function sigPeaksTable for more information.

Value

A MAIT-class object containing the significant features of the scores slot of MAIT-class object used as an input.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`spectralSigFeatures` `sigPeaksTable`

`spectralWelch` *Extract significant features from a MAIT object*

Description

Function spectralWelch takes an `MAIT-class` object and obtains which of the variables are significant given a p-value threshold following a Welch test. The parameters of the significant features can ve printed to an output table (TRUE by default).

Usage

```
spectralWelch(MAIT.object = NULL,
              pvalue = 0.05,
              p.adj="none",
              printCSVfile = TRUE)
```

Arguments

`MAIT.object`	A MAIT-class object where function peakAggregation has already been applied. The output of the function is going to be an update of the same MAIT-class object.
`pvalue`	P-value threshold. Variables having a p-value lower than this value is considered as a significant variable.
`p.adj`	Post-hoc method to be used to correct the p-values.
`printCSVfile`	Set to TRUE if an output table has to be produced. See function sigPeaksTable for more information.

Value

A MAIT-class object containing the significant features of the scores slot of MAIT-class object used as an input.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `peakAggregation` `sigPeaksTable`

`spectralWilcox` *Extract significant features from a MAIT object*

Description

Function spectralWilcox takes an `MAIT-class` object and obtains which of the variables are significant given a p-value threshold following a Mann-Witney-Wilcoxon test. The parameters of the significant features can ve printed to an output table (TRUE by default).

Usage

```
spectralWilcox(MAIT.object = NULL,
              pvalue = 0.05,
              p.adj="none",
              printCSVfile = TRUE,
              jitter = FALSE,
              jitter.factor = 1,
              jitter.amount = 0)
```

Arguments

`MAIT.object`	A MAIT-class object where function peakAggregation has already been applied. The output of the function is going to be an update of the same MAIT-class object.
`pvalue`	P-value threshold. Variables having a p-value lower than this value is considered as a significant variable.
`p.adj`	Post-hoc method to be used to correct the p-values.
`printCSVfile`	Set to TRUE if an output table has to be produced. See function sigPeaksTable for more information.
`jitter`	If it is set to TRUE, a jitter noise is added to the data. This is useful when applying Mann-Whitney tests with ties.
`jitter.factor`	See argument factor of the function jitter.
`jitter.amount`	See argument amount of the function jitter.

Value

A MAIT-class object containing the significant features of the scores slot of MAIT-class object used as an input.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`MAIT-class` `peakAggregation` `sigPeaksTable`

`successRatio` *Extracts the success ratio of a truth table*

Description

Function `successRatio` extracts the success ratio (weighted ratio of samples correctly classified vs total samples) for each class and overall. The value is weighted to take into account the possible different sample number between classes.

Usage

```
successRatio(classes,
             tt,
             ClassWeights)
```

Arguments

`tt`	The truth table from which the success ratio should have to be extracted.
`classes`	Vector containing the class label of each sample.
`ClassWeights`	Vector containing the weights of each class

Value

A numeric value showing the ratio of the samples that are correctly associated according to their real classes.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`Validation`

`Validation` *Cross validated classification over the output of the function* `signPeaksAnova` *or function* `TStudent2Clases`

Description

Function Validation performs a cross-validated classification using three different classifiers: KNN, PLSDA and SVM. The output comes in a table with the classification ratio and its standard error. The classification ratio is weighted to take into account the different sample number of each class.

Usage

```
Validation(Iterations=NULL,
               MAIT.object=NULL,
               trainSamples=NULL,
               PCAscale=FALSE,
               PCAcenter=TRUE,
               RemoveOnePeakSpectra=FALSE,
               tuneSVM=FALSE,
            scale=TRUE)
```

Arguments

`Iterations`	Number of iterations to be performed in the classifications. For each iteration a new training group is randomly chosen.
`MAIT.object`	A MAIT-class object where significant features have already been found.
`trainSamples`	Number of samples per class to construct the train dataset.
`PCAscale`	If method="PCA" and PCAscale is set to TRUE, then the data is scaled following the `prcomp` function. If it is set to TRUE, scale input is ignored.
`PCAcenter`	If method="PCA" and PCAscale is set to TRUE, then the data is centered following the `prcomp` function. If it is set to TRUE, scale input is ignored.
`RemoveOnePeakSpectra`	If it is set to TRUE, all the one-peak spectra are deleted from the dataSet and the resulting `spectralData` object will only contain spectra with more than one peak.
`tuneSVM`	If it is set to TRUE, a tune of parameters is performed before the SVM calculus.
`scale`	If it is set to TRUE, the data is scaled through the spectral mean value. Set to TRUE by default.

Value

The numerical results of the classification per class and per classifier are saved in a MAIT-class object. Additionally, a table is also included in the output both in the list (field table) and printed as a csv file in the folder (working directory)/Validation. A boxplot is also printed as a png in the same folder showing the differences between classifiers. The confusion matrices of each iteration and classifier are also stored as csv files.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

See Also

`peakAggregation` `spectralAnova` `spectralTStudent` `spectralSigFeatures`

Examples

```
data(MAIT_sample)
MAIT<-spectralSigFeatures(MAIT,p.adj="fdr",parametric=TRUE)
MAIT <- Validation(Iterations = 20, trainSamples= 15, MAIT.object = MAIT)
```

`writeExcelTable` *Writes a csv table*

Description

Function `writeExcelTable` writes a csv table with the input data.

Usage

```
writeExcelTable(file,
                file.name)
```

Arguments

`file`	The data to be saved in the csv file
`file.name`	The name of the csv file.

Value

A csv file containg the data provided as input in the file parameter.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

`writeParameterTable` *Writes a csv table containing the parameters launched in the MAIT analysis*

Description

Function `writeParameterTable` writes a csv table where all the provided input parameters in the whole MAIT analysis are saved.

Usage

```
writeParameterTable(listParameters,
         folder)
```

Arguments

`listParameters`	The list of parameters to be printed. This input should be an object of the class MAIT.Parameters
`folder`	The folder where the csv file is going to be saved

Value

A csv file containg the input parameters of the whole run.

Author(s)

Francesc Fernandez, `<francesc.fernandez.albert@upc.edu>`

Index