Package ‘MetCirc’

April 12, 2018

Type Package

Title Navigating mass spectral similarity in high-resolution MS/MS metabolomics data

Version 1.6.0

Date 2017-10-20

Author Thomas Naake <thomasnaake@googlemail.com> and Emmanuel Gaquerel <emmanuel.gaquerel@cos.uni-heidelberg.de>

Maintainer Thomas Naake <thomasnaake@googlemail.com>

VignetteBuilder knitr

Depends R (>= 3.3), amap (>= 0.8), circlize (>= 0.3.9), graphics (>= 3.3), grDevices (>= 3.3), methods (>= 3.3), scales (>= 0.3.0), shiny (>= 1.0.0), stats (>= 3.3)

Suggests BiocGenerics, knitr (>= 1.11)

biocViews Metabolomics, MassSpectrometry, Visualization

Description MetCirc comprises a workflow to interactively explore high-resolution MS/MS metabolomics data: create an MSP object, a format for MS/MS library data, bin m/z values of precursors, calculate similarity between precursors based on the normalised dot product and visualise similarities in a circular layout. Within the interactive framework the user can annotate MS/MS features based on their similarity to (known) related MS/MS features.

License GPL-2

RoxygenNote 6.0.1

NeedsCompilation no

R topics documented:

`adduct` *`adduct` returns adduct ion names of compounds in `MSP`-object*

Description

`adduct` returns adduct ion names of compounds in `MSP`-object.

Usage

```
adduct(x)
```

Arguments

x	object of class MSP

Format

An object of class NULL of length 0.

Value

character

Functions

- adduct: returns adduct ion names of compounds in MSP-objects

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                         splitIndMZ = 2, splitIndRT = NULL)
adduct(finalMSP)
```

adduct<-   adduct<- *sets adduct ion names in* MSP*-object*

Description

adduct<- sets adduct ion names in MSP-object

Arguments

x	object of class MSP, see ?convert2MSP for further information
value	character vector with new adduct ion names

Format

An object of class NULL of length 0.

Value

MSP-object

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                         splitIndMZ = 2, splitIndRT = NULL)
adduct(finalMSP) <- rep("Unknown")
```

allocatePrecursor2mz *allocatePrecursor2mz: Join two data sources*

Description

Allocates precursor ions to candidate m / z values based on minimal distance of m / z and deviance of rt based on an objective function

Usage

```
allocatePrecursor2mz(sd01, sd02, kNN = 10, mzCheck = 1, rtCheck = 30,
     mzVsRTbalance = 10000, splitPattern = "_", splitInd = 2)
```

Arguments

`sd01`	is the output of the `XCMS` and `CAMERA` processing and statistical analysis and `XCMS` and `CAMERA` scripts (see Li et al. 2015 and vignette for further information)
`sd02`	`data.frame` with idMS/MS deconvoluted spectra with fragment ions (m/z, retention time, relative intensity in %) and the corresponding peak correlation group of the precursor ion. `sd02` has to have at least four columns: a column 'mz', 'rt', 'intensity' and 'id'
`kNN`	numerical, number of k-nearest neighbours based on deviation from m/z (i.e. the k entries with the smallest deviation)
`mzCheck`	numerical, maximum tolerated distance for m/z (strong criterion here)
`rtCheck`	numerical, maximum tolerated distance for retention time
`mzVsRTbalance`	numerical, multiplicator for mz value before calculating the (euclidean) distance between two peaks, high value means that there is a strong weight on the deviation m/z value
`splitPattern`	character, character vector to use for splitting, see ?strsplit for further information
`splitInd`	numeric, extract precursor mz at position splitInd

Details

This function combines different data sources. `convertExampleDF` is a `data.frame` which comprises information on a specific metabolite per row stating the average retention time, average m/z, the name of the metabolite, the adduct ion name, the spectrum reference file name and additional information (here: TRIO/LVS). `allocatePrecursor2mz` uses `data.frames` of the kind of `sd01_outputXCMS` and `sd02_deconvoluted` to create a `data.frame` of the kind of `convertExampleDF`. Allocation of precursor ions to candidate m/z values is based on minimal distance of m/z and deviance of retention time based on an objective function. We can specify threshold values for m/z and retention time to be used in `allocatePrecursor2mz`, as well as the number of neighbours based on deviation from m/z values. Also, we can specify the weight to base the selection on the m/z compared to the retention time (`mzVsRTbalance`). This might be useful because m/z values might differ less than the retention time in `sd01_outputXCMS` and `sd02_deconvoluted`. Please note, that it might be problematic to compare `sd01_outputXCMS` and `sd02_deconvoluted` and allocate precursor ions therefrom, especially when data were acquired under different conditions.

Value

allocatePrecursor2mz returns a `data.frame` containing average retention time, average mz, metabolite name, adduct ion name, spectrum reference

Author(s)

Thomas Naake, `<thomasnaake@googlemail.com>`

References

Li et al. (2015): Navigating natural variation in herbivory-induced secondary metabolism in coyote tobacco populations using MS/MS structural analysis. PNAS, 112, E4147–E4155, 10.1073/pnas.1503106112.

Examples

```
data("sd01_outputXCMS", package = "MetCirc")
data("sd02_deconvoluted", package = "MetCirc")
data("convertExampleDF", package = "MetCirc")
allocatePrecursor2mz(sd01 = sd01_outputXCMS, sd02 = sd02_deconvoluted,
    kNN = 10, mzCheck = 1, rtCheck = 30, mzVsRTbalance = 10000, splitPattern = " _ ", splitInd = 2)
```

`binnedMSP` *Example data for* `MetCirc`*:* `binnedMSP`

Description

The object `binnedMSP` is a `matrix`, where rows are metabolites detected in the tissues sepal (SPL), limb (LIM), anther (ANT) and style (STY). The columns contain binned m/z values. Entries contain the intensity (in percent) of a certain metabolite at a certain m/z value. `binnedMSP` is derived from the object `tissue` and `compartmentTissue`.

Usage

```
binnedMSP
```

Format

`matrix`

Value

`matrix`

Author(s)

Thomas Naake, `<thomasnaake@googlemail.com>`

Source

data("idMSMStissueproject", package = "MetCirc") data("idMSMStoMSP", package = "MetCirc") tissueSPL <- compartmentTissue[compartmentTissue[,"SPL"] == TRUE, 1] tissueLIM <- compartmentTissue[compartmentTissue[,"LIM"] == TRUE, 1] tissueANT <- compartmentTissue[compartmentTissue[,"ANT"] == TRUE, 1] tissueSTY <- compartmentTissue[compartmentTissue[,"STY"] == TRUE, 1]

## truncate tissue tissueSPL <- tissue[tissue[,4] tissueLIM <- tissue[tissue[,4] tissueANT <- tissue[tissue[,4] tissueSTY <- tissue[tissue[,4]

## create msp and combine msp objects of different tissues finalMSP <- convert2MSP(tissueSPL, splitIndMZ = 1, splitIndRT = 2, rt = TRUE) finalMSP <- combine(finalMSP, convert2MSP(tissueLIM, splitIndRT = 2, rt = TRUE)) finalMSP <- combine(finalMSP, convert2MSP(tissueANT, splitIndRT = 2, rt = TRUE)) finalMSP <- combine(finalMSP, convert2MSP(tissueSTY, splitIndRT = 2, rt = TRUE))

## create vector with compartments compSPL <- rep("SPL", length(convert2MSP(tissueSPL))) compLIM <- rep("LIM", length(convert2MSP(tissueLIM))) compANT <- rep("ANT", length(convert2MSP(tissueANT)) compSTY <- rep("STY", length(convert2MSP(tissueSTY)))

compartment <- c(compSPL, compLIM, compANT, compSTY) binnedMSP <- binning(msp = finalMSP, tol = 0.01, group = compartment, method = "median") save(binnedMSP, file = "binnedMSP.RData", compress = "bzip2")

`binning` *Bin m/z values*

Description

Bin m/z values

Usage

```
binning(msp, tol = 0.01, group = NULL, method = c("median", "mean"), verbose = FALSE)
```

Arguments

`msp` `MSP`-object, see ?convert2MSP for further information

`tol` `numerical`, boundary value until which neighboured peaks will be joined together

`group` `character` vector, to which group does the entry belong to

`method` `character` vector, method has to be `median` or `mean`

`verbose` `logical` vector, if set to TRUE information will be printed if groups were not detected

Details

The functions `binning` bins fragments together by obtaining bins via calculating either mean or medians of fragments which were put in intervals according to the `tol` parameter.

Value

`binning` returns a `matrix` where rownames are precursor ions (m/z / retention time) and colnames are newly calculated m/z values which were binned. Entires are intensity values in

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("idMSMStoMSP", package = "MetCirc")
binning(msp = finalMSP, tol = 0.01, group = NULL, method = "median", verbose = FALSE)
```

cart2Polar *Calculate polar coordinates from cartesian coordinates*

Description

cart2Polar calculates polar coordinates from cartesian coordinates

Usage

```
cart2Polar(x, y)
```

Arguments

x	cartesian x coordinate
y	cartesian y coordinate

Details

cart2Polar is employed to translate cartesian coordinates into polar coordinates especially in interactive shiny applications when using hovering and clicking features.

Value

cart2Polar returns a list of colar coordinates r and theta

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
x <- 1; y <- 1
cart2Polar(x, y)
```

circosLegend *Plot a legend for circos plot*

Description

circosLegend plots a legend for circos plot using group names.

Usage

```
circosLegend(groupname, highlight = TRUE, colour = NULL, cex = 1)
```

Arguments

groupname	character vector containing "group" and "name" to display, that is a unique identifier of the features, "group" and "name" have to be separated by "_" where "group" is the first and "name" is the last element
highlight	logical, should colours be adjusted to highlight settings?
colour	NULL or character, colour defines the colours which are used for plotting, if NULL default colours are used
cex	numeric, parameter that controls size of the legend in the plot

Details

Internal use in shinyCircos or outside of shinyCircos to reproduce figures.

Value

The function will open a new plot and display colours together with labels.

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
## load binnedMSP
data("binnedMSP", package = "MetCirc")
## use only a selection
binnedMSP <- binnedMSP[c(1:20, 29:48, 113:132, 240:259),]
similarityMat <- createSimilarityMatrix(binnedMSP)
groupname <- rownames(similarityMat)
## plot legend
circosLegend(groupname, highlight = TRUE, colour = NULL, cex = 1)
```

classes — `classes` *returns class names of compounds in* `MSP`*-object*

Description

`classes` returns class names of compounds in `MSP`-object.

Usage

```
classes(x)
```

Arguments

`x`	object of class `MSP`

Format

An object of class `NULL` of length 0.

Value

`character`

Functions

- `classes`: returns class names of metabolites in `MSP`-object

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                        splitIndMZ = 2, splitIndRT = NULL)
classes(finalMSP)
```

classes<- — `classes<-` *sets information in* `MSP`*-object*

Description

`classes<-` sets information in `MSP`-object.

Arguments

`x`	object of class `MSP`, see `?convert2MSP` for further information
`value`	character vector with new classes

Format

An object of class `NULL` of length 0.

Value

MSP-object

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                         splitIndMZ = 2, splitIndRT = NULL)
classes(finalMSP) <- rep("Unknown")
```

---

combine *combine method for MSP-class*

Description

combine combines two objects of class MSP.

Usage

```
combine(object1, object2)

## S4 method for signature 'MSP,MSP'
combine(object1, object2)
```

Arguments

object1	object of class MSP
object2	object of class MSP

Value

MSP-object

Methods (by class)

- object1 = MSP,object2 = MSP: combines two object of class MSP

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP1 <- convert2MSP(sd02_deconvoluted, split = " _ ",
                         splitIndMZ = 2, splitIndRT = NULL)
finalMSP2 <- convert2MSP(sd02_deconvoluted, split = " _ ",
                         splitIndMZ = 2, splitIndRT = NULL)
combine(finalMSP1, finalMSP2)
```

compartmentTissue *Example data for* MetCirc*:* compartmentTissue

Description

The data.frame compartmentTissue is used in the subsection 'Preparing the tissue data set for analysis' in the vignette of MetCirc. In compartmentTissue, information on the organ-localisation of each MS/MS spectrum is stored.

Usage

```
tissue
```

Format

data.frame

Value

data.frame

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Source

internal

convert2MSP *Convert deconvoluted matrix into* MSP*-object*

Description

Convert deconvoluted matrix into MSP-object

Usage

```
convert2MSP(mm, splitPattern = "_", splitIndMZ = 1, splitIndRT = NULL,
 rt = FALSE, names = FALSE, information = FALSE, classes = FALSE, adduct = FALSE)
```

Arguments

mm	matrix, mm has to have three columns with colnames "mz", "intensity" and "id" (order is not important). The column comprises information about the precursor ion which will be assessed by splitPattern and splitInd. Optionally, mm can have colnames "rt", "names", "information", "classes" and "adduct".
splitPattern	character, splitPattern is the pattern which separates elements and precursor m/z

splitIndMZ numeric, the position of the precursor m/z in the character string concerning separation by splitPattern

splitIndRT numeric or NULL, the position of the retention time in the character string concerning separation by splitPattern, if NULL the retention time will be the mean of all retention time values of the MS/MS feature fragments

rt logical, should retention times be retrieved? If set to TRUE, convert2MSP will access the column "rt" in mm which contains the retention time values for each fragment when splitIndRT is NULL, if rt is set to TRUE and splitIndRT is numeric, convert2MSP will access the column "id" to get the retention time at position splitIndRT when splitting with splitPattern

names logical, should names be retrieved? If set to TRUE, convert2MSP will access the column "names" in mm which contains the names of the metabolites

information logical, should further information of metabolites be retrieved? If set to TRUE, convert2MSP will access the column "information" in mm which contains information about the metabolites

classes logical, should classes of metabolites be retrieved? If set to TRUE, convert2MSP will access the column "classes" in mm which contains the names of the metabolites

adduct logical, should adduct ion names of metabolites be retrieved? If set to TRUE, convert2MSP will access the column "adduct" in mm which contains the adduct ion names of the metabolites

Details

The function convert2MSP creates a data entry for each precursor ion. Each entry in the return object has the following information: Num Peaks and a list of fragments together with their intensities; it will further contain information on m/z values of the precursor ion, the retention time, metabolite names, classes, adduct ion name and further information. convert2MSP will access the columns "rt", "names", "information", "classes" and "adduct", respectively, if arguments are set to TRUE. The column "id" has to contain a unique identifier for each MS/MS feature. It is obligatory that each element in the column "id" contains the precursor m/z value, but may contain furhter elements (e.g. peak correlation value or retention time of the precursor ion). Information about the m/z value will be assessed by splitPattern and splitInd. E.g. items in the column "id" can be in the form of "1_163.23", which has to be accessed by setting splitPattern = "_" and splitInd = 2 to access the m/z value of the precursor ion (here: 162.23). If rt is set to TRUE and splitIndRT is NULL, convert2MSP will access the column "rt" to get the retention time values corresponding to each fragment and calculate the mean value, if rt is set to TRUE and splitIndRT numeric, convert2MSP will retrieve the retention time value from column "id".

Value

convert2MSP returns an object of class MSP

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
convert2MSP(mm = sd02_deconvoluted, splitPattern = " _ ", splitIndMZ = 2,
 splitIndRT = NULL, rt = FALSE, names = FALSE, information = FALSE,
 classes = FALSE, adduct = FALSE)
```

`convertExampleDF` *Example data for* `MetCirc`*: convertExampleDF*

Description

`convertExampleDF` is a `data.frame` which comprises information on a specific metabolite per row stating the average retention time, average m/z, the name of the metabolite, the adduct ion name and the spectrum reference file name. The function `allocatePrecursor2mz` uses `data.frames` of the kind of `sd01_outputXCMS` and `sd02_deconvoluted` to create a `data.frame` of the kind of `convertExampleDF`. Allocation of precursor ions to candidate m/z values is based on minimal distance of m/z and deviance of retention time based on an objective function. See `?allocatePrecursor2mz` for further information.

Usage

```
convertExampleDF
```

Format

`data.frame`

Value

`data.frame`

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Source

internal

`convertMSP2MSP` *Convert MSP data frame into object of* `MSP`*-class*

Description

Convert msp data frame into object of `MSP`-class

Usage

```
convertMSP2MSP(msp)
```

Arguments

`msp`	`data.frame`, see `Details` for further information.

Details

msp is a data frame of a .MSP file, a typical data file for MS/MS libraries. The data frame has two columns and contains in the first column the entries "NAME:", "PRECURSORMZ:" (or "EXACTMASS:"), "Num Peaks:" and information on fragments and peak areas/intensities. It may additionally contain row entries: convertMSP2MSP will try to find the row entries "RETENTIONTIME:", "ADDUCTIONNAME:" (or "PRECURSORTYPE:"), "CLASS:" and "INFORMATION:" and extract the respective information in the second column.

Value

convertMSP2MSP returns an object of class MSP.

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("convertMSP2MSP", package = "MetCirc")
convertMSP2MSP(msp = msp2msp)
```

createLink0Matrix *Create a link matrix*

Description

Create a link matrix which links every feature in similarity matrix with another.

Usage

```
createLink0Matrix(similarityMatrix)
```

Arguments

similarityMatrix
: matrix, a similarity matrix that contains the NDP similarity measure between all precursors in the data set

Details

createLink0Matrix creates a matrix from a similarity matrix which includes all connections between features in the similarity matrix, but exclude links which have a similarity of exactly 0.

Value

createLink0Matrix returns a matrix that gives per each row information on linked features

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("binnedMSP", package = "MetCirc")
## truncate binnedMSP
binnedMSP <- binnedMSP[1:28,]
namesPrec <- rownames(binnedMSP)
similarityMat <- createSimilarityMatrix(binnedMSP)
link0Mat <- createLink0Matrix(similarityMatrix = similarityMat)
```

---

createLinkMatrix *Create a matrix which contains features to link (indices)*

Description

Create a matrix which contains features to link (indices)

Usage

```
createLinkMatrix(similarityMatrix, threshold_low, threshold_high)
```

Arguments

similarityMatrix
: matrix, a similarity matrix that contains the NDP similarity measure between all precursors in the data set

threshold_low
: numeric, threshold value for NDP values, below this value linked features will not be included

threshold_high
: numeric, threshold value for NDP values, above this value linked features will not be included

Details

threshold_low and threshold_high are numerical values and truncate similar/identical precursor ions; similarity is currently based on the normalised dot product.

Value

createLinkMatrix returns a matrix that gives per each row information on linked features

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("binnedMSP", package = "MetCirc")
## use only a selection
binnedMSP <- binnedMSP[c(c(1:20, 29:48, 113:132, 240:259)),]
similarityMat <- createSimilarityMatrix(binnedMSP)
createLinkMatrix(similarityMatrix = similarityMat,
     threshold_low = 0.5, threshold_high=1)
```

createOrderedSimMat — *Update colnames and rownames of a similarity matrix according to order m/z, retention time and clustering*

Description

Internal function for shiny application. May also be used outside of shiny to reconstruct figures.

Usage

```
createOrderedSimMat(similarityMatrix, order = c("retentionTime", "mz", "clustering"))
```

Arguments

similarityMatrix
: `matrix`, `similarityMatrix` contains pair-wise similarity coefficients which give information about the similarity between precursors

order
: `character`, one of "retentionTime", "mz" or "clustering"

Details

`createOrderSimMat` takes a similarity matrix and a `character` vector as arguments. It will then reorder rows and columns of the similarityMatrix object such, that it orders rows and columns of similarityMatrix according to m/z, retention time or clustering in each group. `createOrderSimMat` is employed in the shinyCircos function to create `similarityMatrix` objects which will allow to switch between different types of ordering in between groups (sectors) in the circos plot. It may be used as well externally, to reproduce plots outside of the reactive environment (see vignette for a workflow).

Value

`createOrderedSimMat` returns a similarity matrix with ordered rownames according to the `character` vector given to order

Author(s)

Thomas Naake, `<thomasnaake@googlemail.com>`

Examples

```
data("binnedMSP", package = "MetCirc")
data("similarityMat", package = "MetCirc")
## order according to retention time
createOrderedSimMat(similarityMatrix = similarityMat, order = "retentionTime")
```

`createSimilarityMatrix` *Create similarity matrix*

Description

Creates the similarity matrix by calculating the normalised dot product (NDP) between precursors

Usage

```
createSimilarityMatrix(mm, m = 0.5, n = 2)
```

Arguments

`mm`	`matrix`, colnames are all fragments which occur in the dataset, rownames are m/z / rt values, entries of `mm` are intensity values corresponding to their m/z values
`m`	`numeric`, see `?NDP` for further details
`n`	`numeric`, see `?NDP` for further details

Details

createSimilarityMatrix calls a function to calculate the NDP between all precursors in the data set. For further information on how the NDP is calculated see `?NDP` and Li et al. (2015): Navigating natural variation in herbivory-induced secondary metabolism in coyote tobacco populations using MS/MS structural analysis. PNAS, E4147–E4155. Currently `m = 0.5` and `n = 2` are set as default.

Value

`createSimilarityMatrix` returns a similarity matrix that contains the NDP similarity measure between all precursors in the data set

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("binnedMSP", package = "MetCirc")
## truncate binnedMSP
binnedMSP <- binnedMSP[1:28,]
createSimilarityMatrix(binnedMSP, m = 0.5, n = 2)
```

cutLinkMatrix *Create a cut link matrix*

Description

Create a cut link matrix

Usage

```
cutLinkMatrix(LinkMatrix, type = c("all", "inter", "intra"))
```

Arguments

LinkMatrix	matrix, that gives per each row information on linked features
type	character, one of "all", "inter" or "intra"

Details

This function is used to cut features from LinkMatrix. If type = "all", LinkMatrix will not be changed; if type = "inter" the cut LinkMatrix will only contain entries of links which are between groups and not inside groups; contrary to that, if type = "intra" the cut LinkMatrix will only contain entries of links which are inside groups and not between groups.

Value

cutLinkMatrix returns a matrix that gives per each row information on linked features

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("binnedMSP", package = "MetCirc")
## use only a selection
binnedMSP <- binnedMSP[c(c(1:20, 29:48, 113:132, 240:259)),]
similarityMat <- createSimilarityMatrix(binnedMSP)
linkMat <- createLinkMatrix(similarityMatrix = similarityMat, threshold_low = 0.75, threshold_high = 1)
cutLinkMatrix(LinkMatrix = linkMat, type = "all")
```

cutUniquePrecursor *Get unique precursor ions*

Description

Get unique precursor ions

Usage

```
cutUniquePrecursor(precursor, splitPattern = splitPattern,
     splitInd = splitInd, returnCharacter = TRUE)
```

Arguments

precursor	character where features are separated by splitPattern
splitPattern	character, character vector to use for splitting, see ?strsplit for further information
splitInd	numeric, extract precursor mz at position splitInd
returnCharacter	logical, if TRUE return character, if FALSE return numeric

Details

Function for internal usage.

Value

The function cutUniquePrecursor returns character or numeric as specified by parameters.

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
precursor <- "A_269.0455469_-1"
splitPattern <- "_"
splitInd <- 2
cutUniquePrecursor(precursor, splitPattern = splitPattern,
     splitInd = splitInd, returnCharacter = TRUE)
```

getBegEndIndMSP *Get beginning and end indices of each entry in a* data.frame *in* peaks(MSP)*-objects*

Description

Get beginning and end indices of each entry in a data.frame in a peaks(MSP)-object

Usage

```
getBegEndIndMSP(msp)
```

Arguments

msp	data.frame in peaks(MSP)-object, see ?convert2MSP for further information

Details

Internal use to retrieve start and end row indices for fragments of MS/MS features.

Value

getBegEndIndMSP returns a list of length 2 where the first entry contains the start indices and the second the end indices

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                        splitIndMZ = 2, splitIndRT = 3)
finalMSPdf <- peaks(finalMSP)
getBegEndIndMSP(finalMSPdf)
```

---

getLinkMatrixIndices *Get indices in LinkMatrix of feature*

Description

Gets indices in LinkMatrix of feature

Usage

```
getLinkMatrixIndices(groupnameselected, linkMatrix)
```

Arguments

`groupnameselected`
: `character` vector with groupname of selected feature, vector containing "group" and "name" to display, that is a unique identifier of the features, "group" and "name" have to be separated by `"_"` where "group" is the first and "name" is the last element

`linkMatrix`
: `matrix`, in each row there is information about features to be connected

Details

Internal use for function `highlight`.

Value

`getLinkMatrixIndices` returns indices concerning `linkMatrix` to which `groupnameselected` connects

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
## Not run: getLinkMatrixIndices(groupnameselected, linkMatrix)
```

getPrecursorMZ — getPrecursorMZ *returns precursor m/z values of an* MSP*-object*

Description

getPrecursorMZ returns a numeric vector with precursor m/z values

Usage

```
getPrecursorMZ(x)
```

Arguments

x	object of class MSP

Format

An object of class NULL of length 0.

Value

numeric

Functions

- getPrecursorMZ: returns precursor m/z values of an MSP object

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                        splitIndMZ = 2, splitIndRT = NULL)
getPrecursorMZ(finalMSP)
```

getRT — getRT *returns precursor RT values of an* MSP*-object*

Description

getRT returns a numeric vector with all retention time values

Usage

```
getRT(x)
```

Arguments

x	object of class MSP

Format

An object of class NULL of length 0.

Value

`numeric`

Functions

- `getRT`: returns precursor RT values of an `MSP`-object

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                     splitIndMZ = 2, splitIndRT = NULL)
getRT(finalMSP)
```

`highlight` *Add links and highlight sectors*

Description

A function to add links and highlight sectors to an initialised and plotted `circlize` plot with one track.

Usage

```
highlight(groupname, ind, LinkMatrix, colour = NULL, transparency = 0.4, links = TRUE)
```

Arguments

`groupname`	`character` vector containing "group" and "name" to display, that is a unique identifier of the features, "group" and "name" have to be separated by `"_"` where "group" is the first and "name" is the last element
`ind`	`numeric`, indices which will be highlighted
`LinkMatrix`	`matrix`, in each row there is information about features to be connected
`colour`	`NULL` or `character`, colour defines the colours which are used for plotting, if `NULL` default colours are used
`transparency`	`numeric`, defines the transparency of the colours
`links`	`logical`, should links of unselected features be plotted

Details

Internal use for `shinyCircos` or outside of `shinyCircos` to reproduce the figure.

Value

The function will update an existing plot by highlighting a specified sector and connected links.

Author(s)

Thomas Naake, `<thomasnaake@googlemail.com>`

Examples

```
## load binnedMSP
data("binnedMSP", package = "MetCirc")
## use only a selection
binnedMSP <- binnedMSP[c(1:20, 29:48, 113:132, 240:259),]
similarityMat <- createSimilarityMatrix(binnedMSP)
## order similarityMat according to retentionTime and update rownames
simM <- createOrderedSimMat(similarityMat, order = "retentionTime")
## create link matrix
linkMat <- createLinkMatrix(similarityMatrix = simM,
    threshold_low = 0.95, threshold_high = 1)
## cut link matrix (here: only display links between groups)
linkMat_cut <- cutLinkMatrix(linkMat, type = "inter")
## set circlize parameters
circos.par(gap.degree = 0, cell.padding = c(0.0, 0, 0.0, 0),
        track.margin = c(0.0, 0))
groupname <- rownames(simM)
## here: set selectedFeatures arbitrarily
indSelected <- c(2,23,42,62)
selectedFeatures <- groupname[indSelected]
## actual plotting
plotCircos(groupname, linkMat_cut, initialize = TRUE,
    featureNames = TRUE, cexFeatureNames = 0.2, groupSector = TRUE,
    groupName = FALSE, links = FALSE, highlight = TRUE)
## highlight
highlight(groupname = groupname, ind = indSelected, LinkMatrix =
        linkMat_cut, colour = NULL, transparency = 0.4, links = TRUE)
```

idMSMStoMSP-data *Example data for* MetCirc*:* finalMSP

Description

finalMSP is of instance MSP, a container for MS/MS data. finalMSP is derived from the object tissue and compartmentTissue.

Usage

```
finalMSP
```

Format

object of class MSP

Value

object of class MSP

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Source

data("idMSMStissueproject", package = "MetCirc") ## create vectors with precursor names present in tissue tissueSPL <- compartmentTissue[compartmentTissue[,"SPL"] == TRUE, 1] tissueLIM <- compartmentTissue[compartmentTissue[,"LIM"] == TRUE, 1] tissueANT <- compartmentTissue[compartmentTissue[,"ANT"] == TRUE, 1] tissueSTY <- compartmentTissue[compartmentTissue[,"STY"] == TRUE, 1]

## truncate tissue tissueSPL <- tissue[tissue[,4] tissueLIM <- tissue[tissue[,4] tissueANT <- tissue[tissue[,4] tissueSTY <- tissue[tissue[,4]

## create msp and combine msp objects of different tissues finalMSP <- convert2MSP(tissueSPL, rt = TRUE) finalMSP <- combine(finalMSP, convert2MSP(tissueLIM), rt = TRUE) finalMSP <- combine(finalMSP, convert2MSP(tissueANT), rt = TRUE) finalMSP <- combine(finalMSP, convert2MSP(tissueSTY), rt = TRUE)

## write finalMSP to idMSMStoMSP.RData save(finalMSP, file = "idMSMStoMSP.RData", compress = "xz")

---

information — information *returns information of metabolites in* MSP*-object*

Description

information returns information in MSP-object.

Usage

```
information(x)
```

Arguments

x	object of class MSP, see ?convert2MSP for further information

Format

An object of class NULL of length 0.

Value

character

Functions

- information: returns information of metabolites in MSP-object

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                        splitIndMZ = 2, splitIndRT = NULL)
information(finalMSP)
```

information<-   information<- *sets information in* MSP*-object*

Description

information<- sets information in MSP-object

Arguments

x	object of class MSP, see ?convert2MSP for further information
value	character vector with new information

Format

An object of class NULL of length 0.

Value

MSP-object

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                        splitIndMZ = 2, splitIndRT = NULL)
information(finalMSP) <- rep("Unknown")
```

length   length *method for* MSP*-class*

Description

Gives the number of entries in the MSP object.

Usage

```
## S4 method for signature 'MSP'
length(x)
```

Arguments

x	object of class MSP

Value

numeric

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                        splitIndMZ = 2, splitIndRT = NULL)
length(finalMSP)
```

minFragCart2Polar *Calculate the nearest feature in polar coordinates given cartesian coordinates*

Description

Calculates the nearest feature in polar coordinates given cartesian coordinates

Usage

```
minFragCart2Polar(x, y, degreeOfFeatures)
```

Arguments

`x`	cartesian x coordinate
`y`	cartesian y coordinate
`degreeOfFeatures`	list of positions of features

Details

`minFragCart2Polar` is employed to find the feature with the smallest distance from given cartesian coordinates.

Value

`minFragCart2Polar` returns the index of the feature that has the smallest distance to the given coordinates. As `minFragCart2Polar` is used in `shinyCircos` for the track 1 only polar r coordinates between 0.8 and 1 will be used to find the feature with smallest distance.

Author(s)

Thomas Naake, `<thomasnaake@googlemail.com>`

Examples

```
## load binnedMSP
data("binnedMSP", package = "MetCirc")
## use only a selection
binnedMSP <- binnedMSP[c(1:20, 29:48, 113:132, 240:259),]
simM <- createSimilarityMatrix(binnedMSP)
groupname <- rownames(simM)
plotCircos(groupname, NULL, initialize = TRUE, featureNames = FALSE,
     groupName = FALSE, groupSector = FALSE, links = FALSE, highlight = FALSE)
x <- 1
y <- 0
degreeFeatures <- lapply(groupname,
 function(x) mean(circlize:::get.sector.data(x)[c("start.degree", "end.degree")]))
minFragCart2Polar(x, y, degreeOfFeatures = degreeFeatures)
```

MSP *MSP-class*

Description

Definiton of MSP-class in MetCirc. Entries are MS/MS features including their spectra. Allows easy computation of number of entries by entering length(msp), where msp is of class MSP. The MSP-class incorporates accessors for auxiliary information of MS/MS features (names, classes, information, adduct ion name).

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

msp2FunctionalLossesMSP *Convert MSP to MSP with functional losses*

Description

msp2FunctionalLossesMSP converts a MSP-object (with fragments) into a MSP-object with neutral losses

Usage

```
msp2FunctionalLossesMSP(msp)
```

Arguments

msp	MSP-object

Details

The function msp2FunctionalLosses can be used when calculating the similarity based on neutral losses instead of fragments.

Value

msp2FunctionalLossesMSP returns a MSP-object (with neutral losses)

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                        splitIndMZ = 2, splitIndRT = NULL)
finalMSPNL <- msp2FunctionalLossesMSP(msp = finalMSP)
```

msp2msp *Example data for* MetCirc*:* msp2msp

Description

convertMSP2MSP contains the object msp2msp that is a data frame in .MSP format, a typical format for MS/MS library building. Each entry consists of the metabolite name (NAME), the precursor mz (PRECURSORMZ), the retention time (RETENTIONTIME), number of peaks (Num Peaks), together with fragments and their intensity values. In the example used in the function convertMSP2MSP the matrix msp2msp is used to construct an object of class MSP.

Usage

```
msp2msp
```

Format

data.frame

Value

data.frame

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Source

http://prime.psc.riken.jp/Metabolomics_Software/MS-DIAL/, truncated .MSP file of GNPS MS/MS Negative (contains 22 entries): http://prime.psc.riken.jp/Metabolomics_Software/MS-DIAL/MSMS-GNPS-Curated-Neg.msp

names names *returns names in* MSP*-object*

Description

names returns names in MSP-object.

Usage

```
## S4 method for signature 'MSP'
names(x)
```

Arguments

x	object of class MSP, see ?convert2MSP for further information

Value

character

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                        splitIndMZ = 2, splitIndRT = NULL)
names(finalMSP)
```

names<- *names<- sets names in MSP-object*

Description

names<- sets names in MSP-object

Usage

```
## S4 replacement method for signature 'MSP,character'
names(x) <- value
```

Arguments

x	object of class MSP, see ?convert2MSP for further information
value	character vector with new names

Value

MSP-object

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                        splitIndMZ = 2, splitIndRT = NULL)
names(finalMSP) <- rep("Unknown")
```

NDP *Calculate the normalised dot product*

Description

Calculate the normalised dot product (NDP)

Usage

```
NDP(matrow1, matrow2, m = 0.5, n = 2, mass)
```

Arguments

`matrow1` — `character` or `numeric` vector, the entries correspond to the mass vector and contain corresponding intensities to the masses, it is the first feature to compare

`matrow2` — `character` or `numeric` vector, the entries correspond to the mass vector and contain corresponding intensities to the masses, it is the second feature to compare

`m` — `numeric`, exponent to calculate peak intensity-based weights

`n` — `numeric`, exponent to calculate peak intensity-based weights

`mass` — `character` or `numeric` vector, vector with all masses which occur in the data set

Details

The NDP is calculated according to the following formula:

$$NDP = \frac{\sum(W_{S1,i} \cdot W_{S2,i})^2}{\sum(W_{S1,i}^2) * \sum(W_{S2,i}^2)}$$

, with $W = [peakintensity]^m \cdot [m/z]^n$. For further information see Li et al. (2015): Navigating natural variation in herbivory-induced secondary metabolism in coyote tobacco populations using MS/MS structural analysis. PNAS, E4147–E4155. NDP returns a numeric value ranging between 0 and 1, where 0 indicates no similarity between the two MS/MS features, while 1 indicates that the MS/MS features are identical. For the calculation of the NDP only the elements of S1 and S2 that are not equal to 0 will be used.

Value

NDP returns a numeric similarity coefficient between 0 and 1

Author(s)

Thomas Naake, `<thomasnaake@googlemail.com>`

Examples

```
data("binnedMSP", package = "MetCirc")
NDP(matrow1 = binnedMSP[1,], matrow2 = binnedMSP[2,], m = 0.5, n = 2,
 mass = colnames(binnedMSP))
```

peaks — `peaks` *method for* `MSP`*-class*

Description

`peaks` returns the `data.frame` entry with peak information of an `MSP` object.

Usage

```
peaks(object)

## S4 method for signature 'MSP'
peaks(object)
```

Arguments

object	object of class MSP

Value

data.frame

Methods (by class)

- MSP: returns the data.frame of an MSP-object

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                        splitIndMZ = 2, splitIndRT = NULL)
peaks(finalMSP)
```

---

plotCircos *Circular plot to visualise similarity*

Description

Circular plot to visualise similarity

Usage

```
plotCircos(groupname, linkMat, initialize = c(TRUE, FALSE),
     featureNames = c(TRUE, FALSE), cexFeatureNames = 0.3,
     groupSector = c(TRUE, FALSE), groupName = c(TRUE, FALSE),
     links = c(TRUE, FALSE), highlight = c(TRUE, FALSE), colour = NULL,
     transparency = 0.2)
```

Arguments

groupname	character vector containing "group" and "name" to display, that is a unique identifier of the features, "group" and "name" have to be separated by "_" where "group" is the first and "name" is the last element
linkMat	data.frame containing linked features in each row, has five columns (group1, name1, group2, name2, NDP)
initialize	logical, should plot be initialized?
featureNames	logical, should feature names be displayed?
cexFeatureNames	numeric, size of feature names
groupSector	logical, should groups be displayed with background colours?
groupName	logical, should group names (e.g. compartment names or individual names) be displayed?
links	logical, should links be plotted?
highlight	logical, highlight is set to TRUE
colour	NULL or character, colour defines the colours which are used for plotting, if NULL default colours are used
transparency	numeric, defines the transparency of the colours

Details

Internal use for `shinyCircos` or used outside of `shinyCircos` to reproduce figure

Value

The function will initialize a circlize plot and/or will plot features of a circlize plot.

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
## load binnedMSP
data("binnedMSP", package = "MetCirc")
## use only a selection
binnedMSP <- binnedMSP[c(1:20, 29:48, 113:132, 240:259),]
similarityMat <- createSimilarityMatrix(binnedMSP)
## order similarityMat according to retentionTime
simM <- createOrderedSimMat(similarityMat, order = "retentionTime")
## create link matrix
linkMat <- createLinkMatrix(similarityMatrix = simM,
     threshold_low=0.8, threshold_high=1)
## cut link matrix (here: only display links between groups)
linkMat_cut <- cutLinkMatrix(linkMat, type = "inter")
## set circlize paramters
circos.par(gap.degree = 0, cell.padding = c(0.0, 0, 0.0, 0),
         track.margin = c(0.0, 0))
groupname <- rownames(simM)
## actual plotting
plotCircos(groupname, linkMat_cut, initialize = TRUE,
    featureNames = TRUE, cexFeatureNames = 0.3, groupSector = TRUE,
     groupName = FALSE, links = FALSE, highlight = FALSE, colour = NULL,
     transparency = 0.2)
```

---

printInformationSelect *Display information on connected features of selected features*

Description

Displays information on connected features of selected features.

Usage

```
printInformationSelect(groupname, msp = NULL, ind,
 lMatInd, linkMatrixThreshold, similarityMatrix, roundDigits = 2)
```

Arguments

groupname
: character vector with groupname of selected feature, vector containing "group" and "name" to display, that is a unique identifier of the features, "group" and "name" have to be separated by "_" where "group" is the first and "name" is the last element

msp
: MSP, an S4 object of class MSP for information about the selected feature

ind
: numeric

lMatInd
: numeric indices of selected features

linkMatrixThreshold
: matrix that contains information of linked features for given thresholds

similarityMatrix
: matrix that is used to get information on the degree of similarity, similarityMat is an ordered version of a similarity matrix, see ?createOrderedSimMat

roundDigits
: numeric, how many digits should be displayed?

Details

printInformationSelect is for internal use.

Value

character that is in HTML format

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("idMSMStoMSP", package = "MetCirc")
data("binnedMSP", package = "MetCirc")
## use only a selection
binnedMSP <- binnedMSP[c(1:20, 29:48, 113:132, 240:259),]
similarityMat <- createSimilarityMatrix(binnedMSP)
groupname <- rownames(similarityMat)
## order similarityMat according to mz
simMat <- createOrderedSimMat(similarityMat, order = "mz")
groupnameMZ <- rownames(simMat)
linkMat_thr <- createLinkMatrix(simMat, 0.8, 1)
ind <- 2
indMZ <- which(groupname[ind] == truncateName(groupnameMZ, NULL, group = TRUE))
linkMatInds <- getLinkMatrixIndices(groupnameMZ[indMZ], linkMat_thr)
MetCirc:::printInformationSelect(groupname = groupname,
 msp = NULL, ind = ind, lMatInd = linkMatInds,
 linkMatrixThreshold = linkMat_thr,
 similarityMatrix = similarityMat, roundDigits = 2)
```

sd01_outputXCMS *Example data for* MetCirc*:* sd01_outputXCMS

Description

sd01_outputXCMS is the output file from the package XCMS using the data from Li et al. (2015). See Li et al. (2015) for further details.

Usage

```
sd01_outputXCMS
```

Format

data.frame

Value

data.frame

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Source

Li et al. (2015)

sd02_deconvoluted *Example data for* MetCirc*: sd02_deconvoluted*

Description

sd02_deconvoluted contains MS/MS data from Li et al. (2015). It is a data.frame which hosts m/z values, retention time, intensity and the respective precursor m/z values. sd02_deconvoluted originates from Li et al. (2015). See Li et al. (2015) for further information.

Usage

```
sd02_deconvoluted
```

Format

data.frame

Value

data.frame

Author(s)

Thomas Naake, `<thomasnaake@googlemail.com>`

Source

Li et al. (2015)

`shinyCircos` *Interactive visualisation of similarity and navigation of MS/MS features*

Description

Visualise the similarity of MS/MS features in a reactive context. See `Details` the vignette for further descriptions on how to use `shinyCircos`.

Usage

```
shinyCircos(similarityMatrix, msp = NULL, ...)
```

Arguments

`similarityMatrix`
: `matrix`, `similarityMatrix` contains pair-wise similarity coefficients which give information about the similarity between MS/MS features

`msp`
: `MSP`, an S4 object of class `MSP`, the `MSP`-object will be used to display information about the selected feature

`...`
: further arguments passed to `shinyCircos`, e.g. `cexFeatureNames` to pass to `plotCircos` to set font size in `plotCircos` of feature names

Details

The function is based on the `shiny` and `circlize` package. The user can choose interactively thresholds, type of links (between or within groups), display information about MS/MS features, permanently select MS/MS features and export selected precursors. When running `shinyCircos` with the object of class `MSP`, annotation data of selected MS/MS features will be displayed.

Value

`shinyCircos` returns a `character` vector with the (permanently) selected precursors or an object with the entries `msp` and `selectedFeatures` if a `MSP`-object was passed to `shinyCircos`

Author(s)

Thomas Naake, `<thomasnaake@googlemail.com>`

Examples

```
data("idMSMStoMSP", package = "MetCirc")
## truncate files
finalMSP <- finalMSP[c(1:20, 29:48, 113:132, 240:259)]
data("binnedMSP", package = "MetCirc")
binnedMSP <- binnedMSP[c(1:20, 29:48, 113:132, 240:259),]
similarityMat <- createSimilarityMatrix(binnedMSP)
## Not run: shinyCircos(similarityMatrix = similarityMat, msp = finalMSP)
```

---

show — show *method for* MSP*-class*

Description

show prints information on the MSP-object (number of entries).

Usage

```
## S4 method for signature 'MSP'
show(object)
```

Arguments

object	object of class MSP

Value

character

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                     splitIndMZ = 2, splitIndRT = NULL)
show(finalMSP)
```

---

similarityMat — *Example data for* MetCirc*:* similarityMat

Description

similarityMat is a matrix containing the pair-wise similarity scores derived from the idMSMStissueproject data set. See the vignette for a workflow to reproduce the object similarityMat.

Usage

```
similarityMat
```

Format

matrix

Value

`matrix`

Author(s)

Thomas Naake, <`thomasnaake@googlemail.com`>

Source

data("binnedMSP", package = "MetCirc") similarityMat <- createSimilarityMatrix(binnedMSP) save(similarityMat, file = "similarityMat.RData", compress = "xz")

`thresholdLinkMatrix` *Threshold a link matrix*

Description

Threshold a link matrix

Usage

```
thresholdLinkMatrix(linkMatrix, threshold_low, threshold_high)
```

Arguments

`linkMatrix`	`matrix`, a link matrix that gives per each row information on linked features
`threshold_low`	`numeric`, threshold value for NDP values, below this value linked features will not be returned
`threshold_high`	`numeric`, threshold value for NDP values, above this value linked features will not be returned

Details

`threshold_low` and `threshold_high` are numerical values and truncates similar/identical precursor ions; similarity is momentarily based on the normalised dot product.

Value

`thresholdLinkMatrix` returns a matrix that gives per each row information on linked features which are linked above a certain threshold

Author(s)

Thomas Naake, <`thomasnaake@googlemail.com`>

Examples

```
data("binnedMSP", package = "MetCirc")
## use only a selection
binnedMSP <- binnedMSP[c(c(1:20, 29:48, 113:132, 240:259)),]
similarityMat <- createSimilarityMatrix(binnedMSP)
linkMatrix <- createLink0Matrix(similarityMatrix = similarityMat)
thresholdLinkMatrix(linkMatrix = linkMatrix,
     threshold_low = 0.5, threshold_high=1)
```

tissue *Example data for* MetCirc*:* tissue

Description

The data.frame tissue is used in the subsection 'Preparing the tissue data set for analysis' in the vignette of MetCirc. MS/MS data are merged across floral organs in this data.frame.

Usage

```
tissue
```

Format

data.frame

Value

data.frame

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Source

internal

truncateName *Truncate names*

Description

A function to truncate names

Usage

```
truncateName(groupname, roundDigits = 2, group = FALSE)
```

Arguments

groupname	character vector with group and unique idenfier (name)
roundDigits	numeric, how many digits should be displayed?
group	logical, should groups be returned?

Details

groupname is a vector of character strings consisting of a group, retention time and m/z value, separated by "_". It is cumbersome to display such long strings. truncateName truncates these strings by rounding retention time and m/z values by digits given by roundDigits. truncateName is an internal function.

Value

truncateName returns groupname with truncated names without group)

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
    groupname <- "a_100.12345/10.12345"
    truncateName(groupname, roundDigits = 2, group = FALSE)
```

[*Extract parts of a* MSP*-object*

Description

[operator acting on an MSP-object to extract parts.

Usage

```
## S4 method for signature 'MSP,numeric'
x[i]
```

Arguments

x	object of class MSP
i	numeric

Value

MSP-object

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                        splitIndMZ = 2, splitIndRT = NULL)
finalMSP[1]
```

Index