Package 'MetCirc'

April 14, 2017

Type Package

Title A workflow to analyse and visualise metabolomics data

Version 1.0.1

Date 2016-12-20

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 (>= 0.3.5), graphics (>= 3.3), grDevices (>= 3.3), methods (>= 3.3), scales (>= 0.3.0), shiny (>= 0.13.1), stats (>= 3.3)

Suggests BiocGenerics, knitr (>= 1.11)

biocViews Metabolomics, MassSpectrometry, Visualization

Description MetCirc comprises a workflow to interactively explore metabolomics data: create MSP, bin m/z values, calculate similarity between precursors and visualise similarities.

License GPL-2

RoxygenNote 5.0.1

NeedsCompilation no

R topics documented:

allocatePrecursor2mz 2
binnedMSP 4
binning 4
cart2Polar 5
circosLegend 6
combine 7
compartmentTissue 7
convert2MSP 8
convertExampleDF 9
createLink0Matrix 10
createLinkMatrix 11
createOrderedSimMat 11
createSimilarityMatrix 12

cutLinkMatrix . . . 13
cutUniquePreMZ . . . 14
getBegEndIndMSP . . . 15
getLinkMatrixIndices . . . 15
getMetaboliteClass . . . 16
getMetaboliteName . . . 17
getMSP . . . 18
getName . . . 18
getPrecursorMZ . . . 19
getRT . . . 20
highlight . . . 20
idMSMStoMSP-data . . . 22
length . . . 22
minFragCart2Polar . . . 23
MSP . . . 24
msp2FunctionalLossesMSP . . . 24
NDP . . . 25
plotCircos . . . 26
printInformationHover . . . 27
sd01_outputXCMS . . . 28
sd02_deconvoluted . . . 29
setMetaboliteClass . . . 29
setMetaboliteName . . . 30
setName . . . 31
shinyCircos . . . 31
show . . . 32
similarityMat . . . 33
thresholdLinkMatrix . . . 33
tissue . . . 34
truncateName . . . 35
[. . . 35

Index **37**

`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	is a data.frame with idMS/MS deconvoluted spectra with fragment ions (m/z, retention time, relative intensity in %) and the corresponding principal component group with the precursor ion. sd02 has four columns, the first column contains the m/z value, the second column the rt, the third column the intensity, the fourth column the pcgroup_precursorMZ
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 (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

internal

`binning` *Bin m/z values*

Description

Bin m/z values

Usage

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

Arguments

msp	data.frame in msp format, 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"

Details

The functions 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.

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("idMSMStoMSP", package = "MetCirc")
##group <- sample(c("yl", "ol", "s","r"), size = length(finalMSP), replace=TRUE)
binning(msp = finalMSP, tol = 0.01, group = NULL, method = "median")
```

---

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 = c(TRUE, FALSE), colour = NULL)
```

Arguments

groupname	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

Details

Internal use for shiny app or outside of shiny 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)
```

combine *combine method for MSP class*

Description

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 MSP objects

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 format*

Description

Convert deconvoluted matrix into MSP format

Usage

```
convert2MSP(mm, splitPattern = "_", splitIndMZ = 1, splitIndRT = 2,
 names = FALSE, metNames = FALSE, class = FALSE)
```

Arguments

`mm`	matrix, mm has to have four columns with colnames mz, rt, intensity (order is not important). In the fourth column there has to information about the precursor ion which will be assessed by splitPattern and splitInd. Optionally, mm can have colnames names, metNames, class.
`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 in the pcgroup
`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
`metNames`	logical, should names of metabolites be retrieved? If set to TRUE, convert2MSP will access the column "metNames" in mm which contains the names of the metabolites
`class`	logical, should classes of metabolites be retrieved? If set to TRUE, convert2MSP will access the column "class" in mm which contains the names of the metabolites

Details

Creates a data entry for each precursor ion. Each entry in the return object has the following information: NAME, RETENTIONTIME, PRECURSORMZ, METABOLITENAME, ADDUCTIONNAME, Num Peaks and a list of fragments together with their intensities. convert2MSP will access the column name 'name', 'metNames' and 'class', respectively, if arguments are set to TRUE. In the fourth column there has to information about the precursor ion which will be assessed by splitPattern and splitInd. E.g. items in the fourth column 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).

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 = 1,
 splitIndRT = 2, names = FALSE, metNames = FALSE, class = 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

`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 similarityMatrix which includes all connections between features in the similarityMatrix, 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)
```

Arguments

similarityMatrix	matrix, a similarity matrix that contains the NDP similarity measure between all precursors in the data set
threshold	numerical, threshold value for NDP values, below this value linked features will not be included

Details

threshold is a numerical value and filters linked precursor ions; filtering 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 = 0.5)
```

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 timem or clustering in each group. createOrderSimMat is used 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)
```

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 the mass

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.

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)
```

---

cutLinkMatrix *Create a cut LinkMatrix*

Description

Create a cut LinkMatrix

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 = 0.5)
cutLinkMatrix(LinkMatrix = linkMat, type = "all")
```

---

cutUniquePreMZ *Get unique precursor ions*

Description

Get unique precursor ions

Usage

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

Arguments

precursor,	character with 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

Internal function.

Value

cutUniquePreMZ returns character as specified by parameters

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
## Not run: cutUniquePreMZ(precursor, splitPattern = splitPattern,
     splitInd = splitInd, returnCharacter = TRUE)
## End(Not run)
```

getBegEndIndMSP *Get beginning and end indices of each entry in a data.frame in msp format*

Description

Get beginning and end indices of each entry in a data.frame in msp format

Usage

```
getBegEndIndMSP(msp)
```

Arguments

msp	data.frame in msp format, see ?convert2MSP for further information

Details

Internal use to retrieve indices when fragments start and end.

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 <- getMSP(finalMSP)
getBegEndIndMSP(finalMSPdf)
```

getLinkMatrixIndices *Get indices in LinkMatrix of feature*

Description

Gets indices in LinkMatrix of feature

Usage

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

Arguments

groupnameselected	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)
```

---

getMetaboliteClass *getMetaboliteClass returns names of compounds in MSP object*

Description

getMetaboliteClass returns names of compounds in MSP object.

Usage

```
getMetaboliteClass(object)
```

Arguments

object	object of class MSP

Format

An object of class NULL of length 0.

Value

character

Functions

- getMetaboliteClass: returns class names of compounds in MSP objects

Examples

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

`getMetaboliteName` *getMetaboliteName returns names of metabolites in MSP object*

Description

getMetaboliteName returns names in MSP object.

Usage

```
getMetaboliteName(object)
```

Arguments

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

Format

An object of class `NULL` of length 0.

Value

character

Functions

- `getMetaboliteName`: returns names of metabolites in MSP objects

Examples

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

getMSP *getMSP method for MSP class*

Description

Returns the data.frame entry of an MSP object.

Usage

```
getMSP(object)

## S4 method for signature 'MSP'
getMSP(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)
getMSP(finalMSP)
```

getName *getName returns names in MSP object*

Description

getName returns names in MSP object.

Usage

```
getName(object)
```

Arguments

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

Format

An object of class NULL of length 0.

Value

character

Functions

- getName: returns names in MSP objects

Examples

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

---

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

Description

getPrecursorMZ returns a numeric vector with precursor m/z values

Usage

```
getPrecursorMZ(object)
```

Arguments

object	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(object)
```

Arguments

object	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)
```

Arguments

groupname	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	numerical, 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	numerical, defines the transparency of the colours

Details

Internal use for shiny app.

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=0.95)
## 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)
```

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

internal

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

numerical

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

MSP class for msp data.frame. Allows easy computation of length of entries by entering length(msp), where msp is of class MSP.

Arguments

`msp`	a data.frame in msp format

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

`msp2FunctionalLossesMSP`
Convert MSP to MSP with functional losses

Description

msp2FunctionalLossesMSP converts a data.frame in msp format (with fragments) into a data.frame in msp format (with neutral losses)

Usage

```
msp2FunctionalLossesMSP(msp)
```

Arguments

`msp`	data.frame, a data.frame in msp format (with fragments)

Details

msp2FunctionalLosses can be used when you want to calculate the similarity based on neutral losses instead of fragments

Value

msp2FunctionalLossesMSP returns a data.frame in msp format (with neutral losses).

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
## Not run: msp2FunctionalLossesMSP(msp)
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = " _ ",
                    splitIndMZ = 2, splitIndRT = NULL)
finalMSPNL <- msp2FunctionalLossesMSP(msp = finalMSP)
```

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 vector or numerical vector, the entries correspond to the mass vector and contain corresponding intensities to the masses, it is the first feature to compare
matrow2	character vector or numerical 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 vector or numerical 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 precursors, while 1 indicates a strong similarity between the two precursors.

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))
```

`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 = 0.2,
     groupSector = c(TRUE, FALSE), groupName = c(TRUE, FALSE),
     links = c(TRUE, FALSE), highlight = c(TRUE, FALSE), colour = NULL,
     transparency = 0.2)
```

Arguments

`groupname`	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`	numerical, 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, are we in highlighting mode?
`colour`	NULL or character, colour defines the colours which are used for plotting, if NULL default colours are used
`transparency`	numerical, defines the transparency of the colours

Details

Internal use for shiny app

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=0.8)
## 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 = 0.2, groupSector = TRUE,
     groupName = FALSE, links = FALSE, highlight = FALSE, colour = NULL,
     transparency = 0.2)
```

---

printInformationHover *Display information on connected features of hovered features*

Description

Displays information on connected features of hovered features.

Usage

```
printInformationHover(groupname, msp = NULL, ind,
 lMatIndHover, linkMatrixThreshold, similarityMatrix)
```

Arguments

groupname	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 hovered feature
ind	numeric
lMatIndHover	numeric indices of connected features
linkMatrixThreshold	matrix that contains information of linked features of a threshold or greater
similarityMatrix	matrix that is used to get information on the degree of similarity, similarityMat is an ordered version of a similarity matrix

Details

printInformationHover 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)
## order similarityMat according to mz
simMat <- createOrderedSimMat(similarityMat, order = "mz")
groupname <- rownames(simMat)
linkMat_thr <- createLinkMatrix(simMat, 0.9)
ind <- 19
linkMatIndsHover <- getLinkMatrixIndices(groupname[ind], linkMat_thr)
MetCirc:::printInformationHover(groupname = groupname,
 msp = NULL, ind = ind, lMatIndHover = linkMatIndsHover,
 linkMatrixThreshold = linkMat_thr,
 similarityMatrix = simMat)
```

---

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)

`setMetaboliteClass` *setMetaboliteClass sets class names of compounds in MSP objects*

Description

setMetaboliteClass sets names of class names of compounds in MSP objects. To set names pass a vector with class names to the argument `class`.

Usage

```
setMetaboliteClass(object, class)
```

Arguments

`object`	object of class MSP
`class`	character, a vector with new class names

Format

An object of class NULL of length 0.

Value

MSP

Functions

- setMetaboliteClass: sets class names of compounds in MSP objects

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = "_ ",
                        splitIndMZ = 2, splitIndRT = NULL)
setMetaboliteClass(finalMSP, c(rep("unknown", 359), "class1"))
```

---

setMetaboliteName *setMetaboliteName sets metabolite names in MSP objects*

Description

setMetaboliteName sets metabolite names in MSP objects. To set metabolite names pass a vector with names to the argument class.

Usage

```
setMetaboliteName(object, metName)
```

Arguments

object	object of class MSP
metName	character, a vector with new metabolite names

Format

An object of class NULL of length 0.

Value

MSP

Functions

- setMetaboliteName: sets metabolite names in MSP objects

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = "_ ",
                          splitIndMZ = 2, splitIndRT = NULL)
setMetaboliteName(finalMSP, c(rep("unknown", 358), "met1", "met2"))
```

setName *setName sets names in MSP objects*

Description

setName sets names in MSP objects. To set names pass a vector with names to the argument `class`.

Usage

```
setName(object, name)
```

Arguments

`object`	object of class MSP
`name`	character, a vector with new names

Format

An object of class `NULL` of length 0.

Value

MSP

Functions

- `setName`: sets names in MSP objects

Examples

```
data("sd02_deconvoluted", package = "MetCirc")
finalMSP <- convert2MSP(sd02_deconvoluted, split = "_ ",
                        splitIndMZ = 2, splitIndRT = NULL)
setMetaboliteName(finalMSP, c(rep("unknown", 358), "name1", "name2"))
```

shinyCircos *Interactive visualisation of similar precursors*

Description

Visualise similar precursors.

Usage

```
shinyCircos(similarityMatrix, msp, size = 400)
```

Arguments

similarityMatrix	matrix, similarityMatrix contains pair-wise similarity coefficients which give information about the similarity between precursors
msp	MSP, an S4 object of class 'MSP' for information about the hovered feature
size	numerical, image width/height in pixels

Details

The function is based on the shiny and circlize package. Choose interactively thresholds, type of links, hover over precursors, select precursors.

Value

shinyCircos returns a character vector with the selected precursors

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(similarityMat, finalMSP, size = 400)
```

---

show *show method for MSP class*

Description

Prints information on the MSP class (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

internal

`thresholdLinkMatrix` *Threshold a link matrix*

Description

Threshold a link matrix

Usage

```
thresholdLinkMatrix(linkMatrix, threshold)
```

Arguments

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

Details

threshold is a numerical value and filters linked precursor ions; filtering is currently 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 = 0.5)
```

---

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)
```

Arguments

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

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)
```

[*Extract parts of a MSP object*

Description

[operator acting on an MSP object to extract parts.

Usage

```
## S4 method for signature 'MSP,numeric,missing,missing'
x[i, j = "missing",
  drop = "missing"]
```

Arguments

x	object of class MSP
i	numeric
j	missing
drop	missing

Value

MSP object

Examples

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

Index

∗Topic **datasets**
    getMetaboliteClass, 16
    getMetaboliteName, 17
    getName, 18
    getPrecursorMZ, 19
    getRT, 20
    setMetaboliteClass, 29
    setMetaboliteName, 30
    setName, 31
[, 35
[,MSP,numeric,missing,missing-method *([)*, 35

allocatePrecursor2mz, 2

binnedMSP, 4
binning, 4

cart2Polar, 5
circosLegend, 6
combine, 7
combine,MSP,MSP-method *(combine)*, 7
combine,MSP-method *(combine)*, 7
compartmentTissue, 7
convert2MSP, 8
convertExampleDF, 9
createLink0Matrix, 10
createLinkMatrix, 11
createOrderedSimMat, 11
createSimilarityMatrix, 12
cutLinkMatrix, 13
cutUniquePreMZ, 14

finalMSP *(idMSMStoMSP-data)*, 22

getBegEndIndMSP, 15
getLinkMatrixIndices, 15
getMetaboliteClass, 16
getMetaboliteClass,MSP-method *(getMetaboliteClass)*, 16
getMetaboliteName, 17
getMetaboliteName,MSP-method *(getMetaboliteName)*, 17
getMSP, 18
getMSP,MSP-method *(getMSP)*, 18
getName, 18
getName,MSP-method *(getName)*, 18
getPrecursorMZ, 19
getPrecursorMZ,MSP-method *(getPrecursorMZ)*, 19
getRT, 20
getRT,MSP-method *(getRT)*, 20

highlight, 20

idMSMStoMSP-data, 22

length, 22
length,MSP-method *(length)*, 22

minFragCart2Polar, 23
MSP, 24
MSP-class *(MSP)*, 24
msp2FunctionalLossesMSP, 24

NDP, 25

plotCircos, 26
printInformationHover, 27

sd01_outputXCMS, 28
sd02_deconvoluted, 29
setMetaboliteClass, 29
setMetaboliteClass,character,MSP-method *(setMetaboliteClass)*, 29
setMetaboliteName, 30
setMetaboliteName,character,MSP-method *(setMetaboliteName)*, 30
setName, 31
setName,character,MSP-method *(setName)*, 31
shinyCircos, 31
show, 32
show,MSP-method *(show)*, 32
similarityMat, 33

thresholdLinkMatrix, 33
tissue, 34
truncateName, 35