Package 'MetCirc'

August 12, 2022

Type Package

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

Version 1.27.0

Date 2021-11-21

Author Thomas Naake <thomasnaake@googlemail.com>, Johannes Rainer <johannes.rainer@eurac.edu> and Emmanuel Gaquerel <emmanuel.gaquerel@ibmp-cnrs.unistra.fr>

Maintainer Thomas Naake <thomasnaake@googlemail.com>

VignetteBuilder knitr

Depends R (>= 3.5), amap (>= 0.8), circlize (>= 0.3.9), scales (>= 0.3.0), shiny (>= 1.0.0), MSnbase (>= 2.15.3),

Imports ggplot2 (>= 3.2.1), S4Vectors (>= 0.22.0)

Suggests BiocGenerics, graphics (>= 3.5), grDevices (>= 3.5), knitr (>= 1.11), methods (>= 3.5), stats (>= 3.5), testthat (>= 2.2.1)

biocViews ImmunoOncology, Metabolomics, MassSpectrometry, Visualization

Description MetCirc comprises a workflow to interactively explore high-resolution MS/MS metabolomics data. MetCirc uses the Spectrum2 and MSpectra infrastructure defined in the package MSnbase that stores MS/MS spectra. MetCirc offers functionality to calculate similarity between precursors based on the normalised dot product, neutral losses or user-defined functions 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 (>= 3)

RoxygenNote 7.1.0

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

git_branch master

git_last_commit 3f8617e

git_last_commit_date 2022-04-26

Date/Publication 2022-08-12

R topics documented:

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

```
data("spectra", package = "MetCirc")
similarityMat <- compare_Spectra(spectra_tissue[1:10],
    fun = normalizeddotproduct, binSize = 0.01)
linkDf <- createLinkDf(similarityMatrix = similarityMat,
    spectra = spectra_tissue[1:10],
    condition = c("SPL", "LIM", "ANT", "STY"), lower = 0.5, upper = 1)
## cut link data.frame (here: only display links between groups)
linkDf_cut <- cutLinkDf(linkDf, type = "inter")
groupname <- c(as.character(linkDf_cut[, "spectrum1"]),
            as.character(linkDf_cut[, "spectrum2"]))
groupname <- unique(groupname)
## plot legend
circosLegend(groupname, highlight = TRUE, colour = NULL, cex = 1)
```

---

compare_Spectra *Create similarity matrix from ‘MSnbase::MSpectra‘ object*

Description

‘compare_Spectra“ creates a similarity matrix of all Spectrum objects in ‘object‘

Usage

```
compare_Spectra(object, fun, ...)
```

Arguments

object	'MSpectra'
fun	'function' or 'character', see '?MSnbase::compareSpectra' for further information
...	arguments passed to 'compareSpectra'

Details

Function inspired by 'compareSpectra.OnDiskMSnExp'. Possibly transfer to 'MSnbase'. "

Author(s)

Thomas Naake (inspired by 'compareSpectra.OnDiskMSnExp')

Examples

```
data("spectra", package = "MetCirc")
compare_Spectra(spectra_tissue[1:10], fun = "dotproduct")
```

---

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.

Format

'data.frame'

Value

'data.frame'

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Source

internal

`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.frame's 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.

Format

'data.frame'

Value

'data.frame'

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Source

internal

`convertMsp2Spectra` *Convert MSP data frame into object of class 'MSpectra'*

Description

Convert msp data frame into object of class [MSpectra()]

Usage

```
convertMsp2Spectra(msp)
```

Arguments

`msp`	'data.frame' that mimicks the .msp file format, 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:"), "RETENTIONTIME:", Num Peaks:" and information on fragments and peak areas/intensities and will extract the respective information in the second column.

Value

'convertMsp2Spectra' returns an object of class 'MSpectra'

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("convertMsp2Spectra", package = "MetCirc")
convertMsp2Spectra(msp = msp2spectra)
```

---

createLink0df *Create a link matrix*

Description

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

Usage

```
createLink0df(similarityMatrix, spectra, condition)
```

Arguments

similarityMatrix	'matrix', a similarity matrix that contains the NDP similarity measure between all precursors in the data set
spectra	'MSpectra' object
condition	'character', which conditions should be included?

Details

createLink0df 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

createLink0df returns a 'matrix' that gives per each row information on linked features

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("spectra", package = "MetCirc")
data("similarityMat", package = "MetCirc")
link0df <- createLink0df(similarityMatrix = similarityMat,
    spectra_tissue, condition = c("SPL", "LIM", "ANT", "STY"))
```

---

createLinkDf *Create a data frame which contains features to link (indices)*

Description

Create a data frame which contains features to link (indices).

Usage

```
createLinkDf(similarityMatrix, spectra, condition, lower, upper)
```

Arguments

similarityMatrix	'matrix', a similarity matrix that contains the similarity measure between all precursors in the data set
spectra	MSpectra object containing spectra of similarityMatrix
condition	'character', vector containing the conditions/samples for which a linkDf is created
lower	'numeric', threshold value for similarity values, below this value linked features will not be included
upper	'numeric', threshold value for similarity values, above this value linked features will not be included

Details

'lower' and 'upper' are numerical values and truncate similar spectra. The function createLinkDf is a wrapper for the functions 'createLink0df' and 'thresholdLinkDf'.

Value

'createLinkDf' returns a data frame that gives per each row information on linked features

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("spectra", package = "MetCirc")
data("similarityMat", package = "MetCirc")
link0df <- createLink0df(similarityMatrix = similarityMat,
    spectra_tissue, condition = c("SPL", "LIM", "ANT", "STY"))
createLinkDf(similarityMatrix = similarityMat, spectra = spectra_tissue,
    condition = c("SPL", "LIM", "ANT", "STY"), lower = 0.5, upper = 1)
```

`cutLinkDf` *Create a cut data frame with information on links*

Description

Create a cut link data frame

Usage

```
cutLinkDf(linkDf, type = c("all", "inter", "intra"))
```

Arguments

`linkDf`	'data.frame', that gives per each row information on linked features
`type`	'character', one of "all", "inter" or "intra"

Details

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

Value

cutLinkDf returns a data.frame that gives per each row information on linked features

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("spectra", package = "MetCirc")
data("similarityMat", package = "MetCirc")
linkDf <- createLinkDf(similarityMatrix = similarityMat,
    spectra = spectra_tissue, condition = c("SPL", "LIM", "ANT", "STY"),
    lower = 0.75, upper = 1)
cutLinkDf(linkDf = linkDf, type = "all")
```

`getLinkDfIndices` *Get indices in linkDf of feature*

Description

Gets indices in linkDf of feature

Usage

```
getLinkDfIndices(groupnameselected, linkDf)
```

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
`linkDf`	‘data.frame‘, in each row there is information about features to be connected

Details

Internal use for function ‘highlight‘

Value

‘getLinkDfIndices‘ returns indices concerning ‘linkDf‘ to which ‘groupnameselected‘ connects

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

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

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,
  linkDf,
  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
linkDf	‘data.frame‘, 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

```
data("spectra", package = "MetCirc")
similarityMat <- compare_Spectra(spectra_tissue[1:10],
    fun = normalizeddotproduct, binSize = 0.01)
 ## order similarityMat according to retentionTime and update rownames
 simM <- orderSimilarityMatrix(similarityMat, spectra = spectra_tissue[1:10],
             type = "retentionTime")
 ## create link matrix
 linkDf <- createLinkDf(similarityMatrix = simM, spectra = spectra_tissue,
     condition = c("SPL", "LIM", "ANT", "STY"), lower = 0.5, upper = 1)
 ## cut link matrix (here: only display links between groups)
 linkDf_cut <- cutLinkDf(linkDf, type = "inter")
 ## set circlize parameters
 circos.clear()
 circos.par(gap.degree = 0, cell.padding = c(0.0, 0, 0.0, 0),
         track.margin = c(0.0, 0))
 groupname <- c(as.character(linkDf_cut[, "spectrum1"]),
                 as.character(linkDf_cut[, "spectrum2"]))
 groupname <- unique(groupname)
 ## here: set indSelected arbitrarily
 indSelected <- c(2,3)
 ## actual plotting
 plotCircos(groupname, linkDf_cut, initialize = TRUE,
     featureNames = TRUE, cexFeatureNames = 0.2, groupSector = TRUE,
     groupName = FALSE, links = FALSE, highlight = TRUE)
 ## highlight
 highlight(groupname = groupname, ind = indSelected, linkDf = linkDf_cut,
     colour = NULL, transparency = 0.4, links = TRUE)
```

`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

```
data("spectra", package = "MetCirc")
similarityMat <- compare_Spectra(spectra_tissue[1:10],
    fun = normalizeddotproduct, binSize = 0.01)
linkDf <- createLinkDf(similarityMatrix = similarityMat,
    spectra = spectra_tissue[1:10],
    condition = c("SPL", "LIM", "ANT", "STY"), lower = 0.5, upper = 1)
## cut link data.frame (here: only display links between groups)
linkDf_cut <- cutLinkDf(linkDf, type = "inter")
groupname <- c(as.character(linkDf_cut[, "spectrum1"]),
                as.character(linkDf_cut[, "spectrum2"]))
groupname <- unique(groupname)
## set circlize parameters
circos.clear()
circos.par(gap.degree = 0, cell.padding = c(0.0, 0, 0.0, 0),
    track.margin = c(0.0, 0))
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)
```

---

msp2spectra *Example data for 'MetCirc': 'msp2spectra'*

Description

'convertMsp2Spectra' contains the object 'msp2spectra' 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 ‘convertMsp2Spectra‘ the ‘data.frame‘ ‘msp2spectra‘ is used to construct an object of class ‘MSpectra‘.

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

---

neutralloss *Calculate similarity based on neutral losses*

Description

Calculate similarity based on neutral losses (NLS)

Usage

```
neutralloss(x, y, m = 0.5, n = 2, ...)
```

Arguments

x	‘Spectrum2‘ object from ‘MSnbase‘ containing intensity and m/z values, first MS/MS spectrum
y	‘Spectrum2‘ object from ‘MSnbase‘ containing intensity and m/z values, second MS/MS spectrum
m	‘numeric‘, exponent to calculate peak intensity-based weights
n	‘numeric‘, exponent to calculate m/z-based weights
...	further arguments passed to ‘MSnbase:::bin_Spectra‘

Details

Similarities of spectra based on neutral losses are calculated according to the following formula:

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

, with $W = [peakintensity]^m \cdot [NL]^n$ and $NL = |m/z - precursorm/z|$. 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. 'neutralloss' 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. Arguments can be passed to the function 'MSnbase:::bin_Spectra', e.g. to set the width of bins (binSize). Prior to calculating

$$W_{S1}$$

or

$$W_{S2}$$

, all intensity values are divided by the maximum intensity value.

Value

'neutralloss' returns a numeric similarity coefficient between 0 and 1

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("spectra", package = "MetCirc")
x <- spectra_tissue[[1]]
y <- spectra_tissue[[2]]
neutralloss(x, y, m = 0.5, n = 2, binSize = 0.01)
```

`normalizeddotproduct` *Calculate the normalized dot product*

Description

Calculate the normalized dot product (NDP)

Usage

```
normalizeddotproduct(x, y, m = 0.5, n = 2, ...)
```

Arguments

x	'Spectrum2' object from 'MSnbase' containing intensity and m/z values, first MS/MS spectrum
y	'Spectrum2' object from 'MSnbase' containing intensity and m/z values, second MS/MS spectrum
m	'numeric', exponent to calculate peak intensity-based weights
n	'numeric', exponent to calculate m/z-based weights
...	further arguments passed to MSnbase:::bin_Spectra

Details

The normalized dot product is calculated according to the following formula:

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

, 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. 'normalizeddotproduct' 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. Arguments can be passed to the function 'MSnbase:::bin_Spectra', e.g. to set the width of bins (binSize). Prior to calculating

$$W_{S1}$$

or

$$W_{S2}$$

, all intensity values are divided by the maximum intensity value.

Value

'normalizeddotproduct' returns a numeric similarity coefficient between 0 and 1

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("spectra", package = "MetCirc")
x <- spectra_tissue[[1]]
y <- spectra_tissue[[2]]
normalizeddotproduct(x, y, m = 0.5, n = 2, binSize = 0.01)
```

`orderSimilarityMatrix` *Order columns and rows of a similarity matrix according to m/z, retention time and clustering*

Description

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

Usage

```
orderSimilarityMatrix(
  similarityMatrix,
  spectra,
  type = c("retentionTime", "mz", "clustering"),
  group = FALSE
)
```

Arguments

`similarityMatrix`	'matrix', 'similarityMatrix' contains pair-wise similarity coefficients which give information about the similarity between precursors
`spectra`	'MSpectra' object containing spectra that are compared in 'similarityMatrix'
`type`	'character', one of "retentionTime", "mz" or "clustering"
`group`	'logical', if TRUE group separated by "_" will be cleaved from rownames/colnames of similarityMatrix and matched against names of spectra, if FALSE rownames/colnames of similarityMatrix are taken as are and matched against names of spectra

Details

'orderSimilarityMatrix' takes a similarity matrix, spectra (containing information on m/z and retentionTime 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. 'orderSimilarityMatrix' 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

'matrix', 'orderSimilarityMatrix' returns a similarity matrix with ordered rownames according to the 'character' vector given to order

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("spectra", package = "MetCirc")
similarityMat <- compare_Spectra(spectra_tissue[1:10],
    fun = normalizeddotproduct, binSize = 0.01)
## order according to retention time
orderSimilarityMatrix(similarityMatrix = similarityMat,
    spectra_tissue, type = "retentionTime", group = FALSE)
```

`plotCircos` *Circular plot to visualise similarity*

Description

Circular plot to visualise similarity.

Usage

```
plotCircos(
  groupname,
  linkDf,
  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
`linkDf`	‘data.frame‘ containing linked features in each row, has five columns (group1, spectrum1, group2, spectrum2, similarity)
`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

```
data("spectra", package = "MetCirc")
similarityMat <- compare_Spectra(spectra_tissue[1:10],
    fun = normalizeddotproduct, binSize = 0.01)
## order similarityMat according to retentionTime
simM <- orderSimilarityMatrix(similarityMat, spectra = spectra_tissue[1:10],
            type = "retentionTime")
## create link data.frame
linkDf <- createLinkDf(similarityMatrix = simM, spectra = spectra_tissue,
     condition = c("SPL", "LIM", "ANT", "STY"), lower = 0.5, upper = 1)
## cut link data.frame (here: only display links between groups)
linkDf_cut <- cutLinkDf(linkDf, type = "inter")
## set circlize paramters
circos.clear()
circos.par(gap.degree = 0, cell.padding = c(0.0, 0, 0.0, 0),
         track.margin = c(0.0, 0))
groupname <- c(as.character(linkDf_cut[, "spectrum1"]),
                as.character(linkDf_cut[, "spectrum2"]))
groupname <- unique(groupname)
## actual plotting
plotCircos(groupname, linkDf_cut, initialize = TRUE,
    featureNames = TRUE, cexFeatureNames = 0.3, groupSector = TRUE,
     groupName = FALSE, links = FALSE, highlight = FALSE, colour = NULL,
     transparency = 0.2)
```

plotSpectra *Plot pair-wise spectra*

Description

'plotSpectra' plots a spectra of a 'subject' and 'query' spectra. 'plotSpectra' uses 'ggplot' plotting functionality.

Usage

```
plotSpectra(spectra, subject, query)
```

Arguments

spectra	'MSpectra' object
subject	character, name of spectra that is aligned against, character with preceding sample name
query	character, name of spectra that is aligned to subject, character with preceding sample name

Details

Internally, all intensities are normalized to 100%.

Value

'ggplot2' plot

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("spectra", package = "MetCirc")
plotSpectra(spectra_tissue, subject = "SPL_1", query = "SPL_2")
```

printInformationSelect *Display information on connected features of selected features*

Description

Displays information on connected features of selected features.

Usage

```
printInformationSelect(
  select,
  spectra = NULL,
  linkDfInd,
  linkDf,
  similarityMatrix,
  roundDigits = 2
)
```

Arguments

select	'character', obtained from groupname, 'character' of selected feature
spectra	'MSpectra' object containing spectra that are compared in 'similarityMatrix'
linkDfInd	'numeric' indices of selected features
linkDf	'data.frame' 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 '?orderSimilarityMatrix'
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("spectra", package = "MetCirc")
similarityMat <- compare_Spectra(spectra_tissue[1:10],
    fun = normalizeddotproduct, binSize = 0.01)
linkDf <- createLinkDf(similarityMatrix = similarityMat,
    spectra = spectra_tissue[1:10],
    condition = c("SPL", "LIM", "ANT", "STY"), lower = 0.5, upper = 1)
## cut link data.frame (here: only display links between groups)
linkDf_cut <- cutLinkDf(linkDf, type = "inter")
groupname <- c(as.character(linkDf_cut[, "spectrum1"]),
            as.character(linkDf_cut[, "spectrum2"]))
groupname <- unique(groupname)
## arbitrarily select a feature
ind <- 2
linkDfInds <- getLinkDfIndices(groupname[ind], linkDf_cut)
MetCirc:::printInformationSelect(groupname[ind],
    spectra = spectra_tissue[1:10], linkDfInd = linkDfInds,
    linkDf = linkDf_cut, similarityMatrix = similarityMat)
```

`recordPlotFill_degreeFeatures` *Record a plot of filled features and the degree of features*

Description

'recordPlotFill_degreeFeatures' records a plot of filled features and returns the degree of features.

Usage

```
recordPlotFill_degreeFeatures(type_match, ...)
```

Arguments

`type_match`	'character', ordered vector according to type
`...`	further arguments passed to 'plotCircos'

Details

Helper function for 'shinyCircos'.

Value

'list' of length 2, entry 'plotFill' is of 'recordedplot' and entry 'degreeFeatures' that is a 'list' of vectors of 'numeric(1)'

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
type_match <- c("a_1", "a_2", "a_3", "b_1", "b_2", "b_3", "c_1", "c_2")
MetCirc:::recordPlotFill_degreeFeatures(type_match)
```

`recordPlotHighlight` *Return a 'recordedplot' of 'plotCircos' plot with 'highlight = TRUE'*

Description

'recordPlotHighlight' returns a 'recordedplot' object of 'plotCircos' with 'highlight = TRUE'

Usage

```
recordPlotHighlight(type_match, ...)
```

Arguments

`type_match`	'character', ordered vector according to type
`...`	further arguments passed to 'plotCircos'

Details

Helper function for 'shinyCircos'.

Value

'recordedplot'

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
type_match <- c("a_1", "a_2", "a_3", "b_1", "b_2", "b_3", "c_1", "c_2")
MetCirc:::recordPlotHighlight(type_match)
```

`replayPlotAdd` *Plot plotCircos or highlight*

Description

'replayPlotAdd' plots additional plots on a plot, either plots 'plotCircos' or 'highlight'.

Usage

```
replayPlotAdd(
  orderMatch = "mz",
  onCircle = FALSE,
  linkDf,
  mz_match,
  rt_match,
  clust_match,
  ind,
  indMz,
  indRT,
  indCluster
)
```

Arguments

`orderMatch`	orderMatch 'character', either '"mz"', '"retentionTime"' or '"clustering"'
`onCircle`	'logical', are coordinates on circle. If FALSE and no features are selected ('length(ind) == 0'), then filled plots are replayed, otherwise highlighted plots are replayed.
`linkDf`	'data.frame' that contains information of linked features for given thresholds
`mz_match`	'character', ordered vector according to m/z
`rt_match`	'character', ordered vector according to retention time
`clust_match`	'character', ordered vector according to clustering
`ind`	'numeric', indices of clicked features
`indMz`	'numeric', indices of clicked features for '"mz"' ordering
`indRT`	'numeric', indices of clicked features for '"retentionTime"' ordering
`indCluster`	'numeric', indices of clicked features for '"clustering"' ordering

Details

Helper function for 'shinyCircos'.

Value

Depending on 'onCircle' and 'indMz' either returns 'plotCircos' or 'highlight'

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("spectra", package = "MetCirc")
similarityMat <- compare_Spectra(spectra_tissue[1:10],
    fun = normalizeddotproduct, binSize = 0.01)
## order according to retention time
mz_match <- MetCirc:::typeMatch_link0(similarityMatrix = similarityMat,
    spectra = spectra_tissue, type = "mz",
    condition = c("SPL", "LIM", "ANT", "STY"))
linkDf <- mz_match[["link0df"]]
mz_match <- mz_match[["type_match"]]
rt_match <- MetCirc:::typeMatch_link0(similarityMatrix = similarityMat,
    spectra = spectra_tissue, type = "retentionTime",
    condition = c("SPL", "LIM", "ANT", "STY"))
rt_match <- rt_match[["type_match"]]
clust_match <- MetCirc:::typeMatch_link0(similarityMatrix = similarityMat,
    spectra = spectra_tissue, type = "clustering",
    condition = c("SPL", "LIM", "ANT", "STY"))
clust_match <- clust_match[["type_match"]]
circos.initialize(mz_match,##, levels  =  mz_match),
    xlim = matrix(rep(c(0,1), length(mz_match)), ncol = 2, byrow = TRUE))
#circos.trackPlotRegion(factor(mz_match, levels = mz_match), ylim = c(0,1))
MetCirc:::replayPlotAdd(orderMatch = "mz", onCircle = FALSE, linkDf = linkDf,
    mz_match = mz_match, rt_match = rt_match, clust_match = clust_match,
    ind = 1, indMz = NULL, indRT = NULL, indCluster = NULL)
```

---

replayPlotOrder *Wrapper for 'replayPlot'*

Description

'replayPlotOrder' will call 'replayPlot' from 'grDevices' with a 'recordedplot' object based on 'orderMatch'.

Usage

```
replayPlotOrder(orderMatch = "mz", onCircle = FALSE, plot_l, ind)
```

Arguments

orderMatch	'character', either '"mz"', '"retentionTime"' or '"clustering"'
onCircle	'logical', are coordinates on circle. If FALSE and no features are selected ('length(ind) == 0'), then filled plots are replayed, otherwise highlighted plots are replayed.
plot_l	'list' with plots
ind	'numeric', indices of clicked features

Details

Helper function for ‘shinyCircos‘.

Value

‘replayedplot‘

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
type_match <- c("a_1", "a_2", "a_3", "b_1", "b_2", "b_3", "c_1", "c_2")
plotCircos(type_match, NULL, initialize = TRUE, featureNames = TRUE,
    groupSector = TRUE, groupName = FALSE, links = FALSE,
    highlight = TRUE)
p <- recordPlot()
plot.new()
plot_l <- list(highlightMz = p)
MetCirc:::replayPlotOrder(orderMatch = "mz", onCircle = TRUE,
    plot_l = plot_l, ind = NULL)
```

---

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.

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.

Format

'data.frame'

Value

'data.frame'

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Source

Li et al. (2015)

select *Select variable based on condition*

Description

'select' returns 'mz', 'rt' or 'clust' depending on 'condition'.

Usage

```
select(condition, mz, rt, clust)
```

Arguments

condition	'character', either '"mz"', '"retentionTime"' or '"clustering"'
mz	object to return if 'condition == "mz"'
rt	object to return if 'condition == "retentionTime"'
clust	object to return if 'condition == "clustering"'

Details

Helper function for 'shinyCircos', 'replayPlotOrder' and 'replayPlotAdd'.

Value

'mz', 'rt' or 'clust' depending on condition

Author(s)

Thomas Naake <thomasnaake@googlemail.com>

Examples

```
mz <- 1
rt <- 2
clust <- 3
MetCirc:::select(condition = "mz", mz = mz, rt = rt, clust = clust)
```

---

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, spectra, condition, ...)
```

Arguments

similarityMatrix	'matrix', 'similarityMatrix' contains pair-wise similarity coefficients which give information about the similarity between MS/MS features
spectra	an S4 object of class 'MSpectra', the 'MSpectra' object will be used to display information about the selected feature and will store information of annotation
condition	'character' vector, specifies which condtions/samples are displayed
...	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. The 'MSpectra' object stores annotation information about the MS/MS features. Names of features within the 'similarityMatrix' have to be found as entries in 'MSpectra'. 'names(MSpectra)' are used as identifiers and 'colnames'/'rownames' from 'similarityMatrix' are cleaved by the group identifier (separated by "_").

Annotation information is taken from ‘spectra‘ from the columns "names", "information", "classes" and "adduct" in the slot ‘elementMetadata‘ of ‘spectra‘. After exiting the application, the annotation will be written to the respective columns in the slot ‘elementMetadata‘. If one or several of these columns is already present in ‘elementMetadata‘, the column(s) will be used as the source of annotation information.

Value

‘character‘, ‘shinyCircos‘ returns a ‘character‘ vector with the permanently selected precursors and an object with the ‘MSpectra‘ object containing the annotation.

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("spectra", package = "MetCirc")
similarityMat <- compare_Spectra(spectra_tissue[1:10],
    fun = normalizeddotproduct, binSize = 0.01)
## Not run:
shinyCircos(similarityMatrix = similarityMat,
    spectra = spectra_tissue, condition = c("SPL", "LIM", "ANT", "STY"))

## End(Not run)
```

---

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‘.

Format

‘matrix‘

Value

‘matrix‘

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Source

data("spectra", package = "MetCirc") similarityMat <- compare_Spectra(spectra_tissue, fun = normalizeddotproduct, binSize = 0.01) save(similarityMat, file = "similarityMat.RData", compress = "xz")

`spectraCond` *Get MS/MS spectra that are present in condition*

Description

'spectraCond' returns the names of 'spectra' that are present in condition, corresponding to the slot 'elementMetadata@listData'.

Usage

```
spectraCond(spectra, condition)
```

Arguments

`spectra`	'MSpectra' object of 'MSnbase' package
`condition`	'character', vector with conditions found as columns in the elementMetadata slot

Details

Helper function in 'createLink0df' and 'shinyCircos'.

Value

'list', named 'list' with 'character' vector as entries that contains the names of the MS/MS entries in 'spectra' that are present in the 'conditon' (tissues, stress conditions, time points, etc.)

Author(s)

Thomas Naake <thomasnaake@googlemail.com>

Examples

```
data("spectra", package = "MetCirc")
MetCirc:::spectraCond(spectra_tissue,
    condition = c("SPL", "LIM", "ANT", "STY"))
```

spectra_tissue *Example data for 'MetCirc': 'spectra_tissue'*

Description

'spectra_tissue' is a 'MSpectra' object containing 'Spectrum2' objects derived from the 'idMSMStissueproject' data set. See the vignette for a workflow to reproduce the object 'spectra'.

Format

'matrix'

Value

'matrix'

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Source

data("idMSMStissueproject", package = "MetCirc") ## get all MS/MS spectra tissue <- tissue[tissue[, "id"] id_uniq <- unique(tissue[, "id"])

obtain precursor m/z from id_uniq prec_mz_l <- lapply(strsplit(as.character(id_uniq), split = "_"), "[", 1) prec_mz_l <- lapply(prec_mz_l, as.numeric)

obtain m/z from fragments per precursor m/z mz_l <- lapply(id_uniq, function(x) tissue[tissue[, "id"] == x, "mz"]) ## obtain corresponding intensity values int_l <- lapply(id_uniq, function(x) tissue[tissue[, "id"] == x, "intensity"]) ## obtain retention time by averaging all retention time values rt_l <- lapply(id_uniq, function(x) tissue[tissue[, "id"] == x, "rt"]) rt_l <- lapply(rt_l, mean)

create list of spectrum2 objects spectrum2_tissue <- lapply(1:length(mz_l), function(x) new("Spectrum2", rt = rt_l[[x]], precursorMz = prec_mz_l[[x]], mz = mz_l[[x]], intensity = int_l[[x]]))

combine list of spectrum2 objects to MSpectra object, ## use SPL, LIM, ANT, STY for further analysis spectra_tissue <- MSpectra(spectrum2_tissue, elementMetadata = DataFrame(compartmentTissue[, c("SPL", "LIM", "ANT", "STY")]))

save(spectra_tissue, file = "spectra.RData", compress = "xz")

`thresholdLinkDf` *Threshold a data frame containing information on links*

Description

Threshold a link data frame based on lower and upper similarity values. The function will return that lie within the defined bounds.

Usage

```
thresholdLinkDf(link0df, lower = 0.75, upper = 1)
```

Arguments

`link0df`	'data.frame', a link data frame that gives per each row information on linked features
`lower`	'numeric', threshold value for similarity values, below this value linked features will not be returned
`upper`	'numeric', threshold value for similarity values, above this value linked features will not be returned

Details

'lower' and 'upper' are numerical values and truncates mass spectra based on their similarity values.

Value

'thresholdLinkDf' returns a data frame that gives per each row information on linked features which are linked within certain thresholds.

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("spectra", package = "MetCirc")
data("similarityMat", package = "MetCirc")
link0df <- createLink0df(similarityMatrix = similarityMat,
    spectra_tissue, condition = c("SPL", "LIM", "ANT", "STY"))
thresholdLinkDf(link0df = link0df, lower = 0.5, upper = 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'.

Format

'data.frame'

Value

'data.frame'

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Source

internal

typeMatch_link0 *Get typeMatch and link0 data frame*

Description

'typeMatch_link0' returns a list with accessors '"link0df"' and '"type_match"'

Usage

```
typeMatch_link0(similarityMatrix, spectra, type, condition)
```

Arguments

similarityMatrix	'matrix' with pair-wise similarity values
spectra	'MSpectra' object
type	'character', either '"mz"', '"retentionTime"', '"clustering"'
condition	'character', tissue

Details

Helper function for 'shinyCircos'.

Value

'list' of length 2, entry 'link0df' is of 'data.frame' and entry 'type_match' that is a 'character' vector

Author(s)

Thomas Naake, <thomasnaake@googlemail.com>

Examples

```
data("spectra", package = "MetCirc")
similarityMat <- compare_Spectra(spectra_tissue[1:10],
    fun = normalizeddotproduct, binSize = 0.01)
## order according to retention time
MetCirc:::typeMatch_link0(similarityMatrix = similarityMat,
    spectra = spectra_tissue, type = "mz",
    condition = c("SPL", "LIM", "ANT", "STY"))
```

Index