Package 'ncGTW'

November 1, 2025

Type Package

Title Alignment of LC-MS Profiles by Neighbor-wise Compound-specific Graphical Time Warping with Misalignment Detection

Version 1.25.0

Author Chiung-Ting Wu <`ctwu@vt.edu`>

Maintainer Chiung-Ting Wu <`ctwu@vt.edu`>

biocViews Software, MassSpectrometry, Metabolomics, Alignment

Description The purpose of ncGTW is to help XCMS for LC-MS data alignment. Currently, ncGTW can detect the misaligned feature groups by XCMS, and the user can choose to realign these feature groups by ncGTW or not.

License GPL-2

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

LinkingTo Rcpp

Suggests BiocStyle, knitr, testthat, rmarkdown

VignetteBuilder knitr

Depends methods, BiocParallel, xcms

Imports Rcpp, grDevices, graphics, stats

BugReports `https://github.com/ChiungTingWu/ncGTW/issues`

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

git_branch devel

git_last_commit b8d200e

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2025-10-31

Contents

`ncGTW-package` *ncGTW: A package for detecting and aligning the misaligned features in LC-MS data*

Description

The purpose of ncGTW is to help XCMS for LC-MS data alignment. Currently, ncGTW can detect the misaligned feature groups by XCMS, and the user can choose to realign these feature groups by ncGTW or not.

References

Wu, Chiung-Ting, et al. Alignment of LC-MS Profiles by Neighbor-wise Compound-specific Graphical Time Warping with Misalignment Detection. bioRxiv, 2019, 715334.

`adjustRT` *Adjust retention time*

Description

This function produces the new warping functions (RT lists) with the realignment result.

Usage

```
adjustRT(xcmsLargeWin, ncGTWinput, ncGTWoutput, ppm)
```

Arguments

`xcmsLargeWin`	A `xcmsSet-class` object.
`ncGTWinput`	A `ncGTWinput` object.
`ncGTWoutput`	A `ncGTWoutput` object.
`ppm`	Should be set as same as the one when performing the peak detection function in `xcms`.

Details

This function produces the new warping functions (RT lists) with the realignment result.

Value

A `ncGTWwarp` object.

Examples

```
# obtain data
data('xcmsExamples')
xcmsLargeWin <- xcmsExamples$xcmsLargeWin
xcmsSmallWin <- xcmsExamples$xcmsSmallWin
ppm <- xcmsExamples$ppm

# detect misaligned features
excluGroups <- misalignDetect(xcmsLargeWin, xcmsSmallWin, ppm)

# obtain the paths of the sample files
filepath <- system.file("extdata", package = "ncGTW")
file <- list.files(filepath, pattern="mzxml", full.names=TRUE)

tempInd <- matrix(0, length(file), 1)
for (n in seq_along(file)){
    tempCha <- file[n]
    tempLen <- nchar(tempCha)
    tempInd[n] <- as.numeric(substr(tempCha, regexpr("example", tempCha) + 7,
        tempLen - 6))
```

```
}
# sort the paths by data acquisition order
file <- file[sort.int(tempInd, index.return = TRUE)$ix]
## Not run:
# load the sample profiles
ncGTWinputs <- loadProfile(file, excluGroups)

# initialize the parameters of ncGTW alignment with default
ncGTWparam <- new("ncGTWparam")

# run ncGTW alignment
ncGTWoutputs <- vector('list', length(ncGTWinputs))
for (n in seq_along(ncGTWinputs))
    ncGTWoutputs[[n]] <- ncGTWalign(ncGTWinputs[[n]], xcmsLargeWin, 5,
        ncGTWparam = ncGTWparam)

# adjust RT with the realignment results from ncGTW
ncGTWres <- xcmsLargeWin
ncGTWRt <- vector('list', length(ncGTWinputs))
for (n in seq_along(ncGTWinputs)){
    adjustRes <- adjustRT(ncGTWres, ncGTWinputs[[n]], ncGTWoutputs[[n]], ppm)
    xcms::peaks(ncGTWres) <- ncGTWpeaks(adjustRes)
    ncGTWRt[[n]] <- rtncGTW(adjustRes)
}

# apply the adjusted RT to a xcmsSet object
xcms::groups(ncGTWres) <- excluGroups[ , 2:9]
xcms::groupidx(ncGTWres) <- xcms::groupidx(xcmsLargeWin)[excluGroups[ , 1]]
rtCor <- vector('list', length(xcms::filepaths(ncGTWres)))
for (n in seq_along(file)){
    rtCor[[n]] <- vector('list', dim(excluGroups)[1])
    for (m in seq_len(dim(excluGroups)[1]))
        rtCor[[n]][[m]] <- ncGTWRt[[m]][[n]]
}
slot(ncGTWres, 'rt')$corrected <- rtCor

## End(Not run)
```

`alignData,ncGTWoutput-method` *ncGTWoutput-accessors*

Description

Accessors to the alignment information and result by ncGTW.

Usage

```
## S4 method for signature 'ncGTWoutput'
alignData(object)
```

Arguments

`object`	a `ncGTWoutput` object.

Value

`alignData` returns a matrix in which each row is a sample profile after downsampling.

Examples

```
# obtain data
data('xcmsExamples')
xcmsLargeWin <- xcmsExamples$xcmsLargeWin
xcmsSmallWin <- xcmsExamples$xcmsSmallWin
ppm <- xcmsExamples$ppm

# detect misaligned features
excluGroups <- misalignDetect(xcmsLargeWin, xcmsSmallWin, ppm)

# obtain the paths of the sample files
filepath <- system.file("extdata", package = "ncGTW")
file <- list.files(filepath, pattern="mzxml", full.names=TRUE)

tempInd <- matrix(0, length(file), 1)
for (n in seq_along(file)){
    tempCha <- file[n]
    tempLen <- nchar(tempCha)
    tempInd[n] <- as.numeric(substr(tempCha, regexpr("example", tempCha) + 7,
        tempLen - 6))
}
# sort the paths by data acquisition order
file <- file[sort.int(tempInd, index.return = TRUE)$ix]
## Not run:
# load the sample profiles
ncGTWinputs <- loadProfile(file, excluGroups)

# initialize the parameters of ncGTW alignment with default
ncGTWparam <- initncGTWparam()

# run ncGTW alignment
ncGTWoutputs <- vector('list', length(ncGTWinputs))
for (n in seq_along(ncGTWinputs))
    ncGTWoutputs[[n]] <- ncGTWalign(ncGTWinputs[[n]], xcmsLargeWin, 5,
        ncGTWparam = ncGTWparam)

data <- alignData(ncGTWoutputs[[1]])
rt <- scanRange(ncGTWoutputs[[1]])
paths <- ncGTWpath(ncGTWoutputs[[1]])
downSam <- downSample(ncGTWoutputs[[1]])

## End(Not run)
```

`compCV` *Compare CV*

Description

This function calculates the coefficient of variation of each feature.

Usage

```
compCV(XCMSresFilled, na.rm = FALSE)
```

Arguments

`XCMSresFilled`	A `xcmsSet-class` object.
`na.rm`	Omit the samples in which the feature is not detected, and the default is FALSE.

Details

This function calculates the coefficient of variation of each feature across all the samples. If a sample is detected with more than one peaks in the feature, the function will pick the one with the highest intensity value.

Value

A vector of the same length as the row number of the `group` slot in `XCMSresFilled`, in which each element is the CV.

Examples

```
# obtain data
data('xcmsExamples')
xcmsLargeWin <- xcmsExamples$xcmsLargeWin

cv <- compCV(xcmsLargeWin)
```

`fillPeaksChromPar` *Edited XCMS fillPeaksChromPar for feature-wise warping functions*

Description

This function is edited from `fillPeaksChromPar` in `fillPeaks.chrom-methods` to accept feature-wise warping functions.

Usage

```
fillPeaksChromPar(arg)
```

Arguments

`arg`	A list sent from `fillPeaks.chrom-methods`.

Details

This function is for parallelly filling missing peaks with feature-wise warping functions. The original function in `fillPeaks.chrom-methods` can only handle sample-wise warping functions.

Value

A list of sameple index vector and filled peak matrix.

`getPeaksncGTW` *Edited XCMS getPeaks for feature-wise warping functions*

Description

This function is edited from `getPeaks-methods` to accept feature-wise warping functions.

Usage

```
getPeaksncGTW(object, peakrange, step = 0.1, naidx)
```

Arguments

`object`	An `xcmsRaw-class` object.
`peakrange`	`matrix` with 4 required columns `"mzmin"`, `"mzmax"`, `"rtmin"` and `"rtmax"`.
`step`	`numeric(1)` defining the bin size for the profile matrix generation.
`naidx`	A vector contains the sample indexes need to be filled.

Details

This function is for parallelly filling missing peaks with feature-wise warping functions. The original code function in `getPeaks-methods` can only handle sample-wise warping functions.

Value

A list of sameple index vector and filled peak matrix.

`groupInfo,ncGTWinput-method`

ncGTWinput-accessors

Description

Accessors to the feature and profiles loaded by `loadProfile`.

Usage

```
## S4 method for signature 'ncGTWinput'
groupInfo(object)

## S4 method for signature 'ncGTWinput'
profiles(object)

## S4 method for signature 'ncGTWinput'
rtRaw(object)

## S4 method for signature 'ncGTWoutput'
scanRange(object)

## S4 method for signature 'ncGTWoutput'
ncGTWpath(object)

## S4 method for signature 'ncGTWoutput'
downSample(object)
```

Arguments

`object`	a `ncGTWinput` object.

Value

`groupInfo` returns a vector of the information of the loaded feature.

`profiles` returns a raw data matrix in which each row is a sample profile.

`rtRaw` returns a raw RT matrix in which each row is the corresponding sample RT.

`scanRange` returns a downsampled RT matrix in which each row is the corresponding sample RT in `data`.

`ncGTWpath` returns a list of the same length as the sample number, in which each element is a matrix of the alignment result of the corresponding sample.

`downSample` returns the factor of downsampling when perform ncGTW alignment.

Examples

```
# obtain data
data('xcmsExamples')
xcmsLargeWin <- xcmsExamples$xcmsLargeWin
xcmsSmallWin <- xcmsExamples$xcmsSmallWin
ppm <- xcmsExamples$ppm

# detect misaligned features
excluGroups <- misalignDetect(xcmsLargeWin, xcmsSmallWin, ppm)

# obtain the paths of the sample files
filepath <- system.file("extdata", package = "ncGTW")
file <- list.files(filepath, pattern="mzxml", full.names=TRUE)

tempInd <- matrix(0, length(file), 1)
for (n in seq_along(file)){
    tempCha <- file[n]
    tempLen <- nchar(tempCha)
    tempInd[n] <- as.numeric(substr(tempCha, regexpr("example", tempCha) + 7,
        tempLen - 6))
}
# sort the paths by data acquisition order
file <- file[sort.int(tempInd, index.return = TRUE)$ix]
## Not run:
# load the sample profiles
ncGTWinputs <- loadProfile(file, excluGroups)

gInfo <- groupInfo(ncGTWinputs[[1]])
prof <- profiles(ncGTWinputs[[1]])
rtR <- rtRaw(ncGTWinputs[[1]])

## End(Not run)
```

`loadProfile` *Load sample profiles for each peak group*

Description

This function loads each raw sample profiles from the file with certain m/z and RT range.

Usage

```
loadProfile(filePaths, excluGroups, mzAdd = 0.005, rtAdd = 10,
  profstep = 0, BPPARAM = BiocParallel::SnowParam(workers = 1))
```

Arguments

`filePaths`	The character vector of the loading file paths.

`excluGroups`	The output matrix of `misalignDetect` or `xcmsSet-class`$group, in which `mzmin`, `mzmax`, `rtmin`, and `rtmax` are set as the m/z and RT range for loading.
`mzAdd`	The extra m/z range for loading (both sides), and the default is 0.005.
`rtAdd`	The extra RT range for loading (both sides), and the default is 10 (seconds).
`profstep`	The size of each m/z bin for peak integration, and the default is 0.
`BPPARAM`	A object of **BiocParallel** to control parallel processing, and can be created by `SerialParam`, `MulticoreParam`, or `SnowParam`.

Details

This function obtains the extracted ion chromatogram for each sample at the givin m/z and RT range with a certain m/z bin size for integration. Considering there may be missing peak by peak detection, `mzAdd` and `rtAdd` are to increase the integration range.

Value

A list of the same length as the row number of `excluGroups`, in which each element is a `ncGTWinput` object.

Examples

```
# obtain data
data('xcmsExamples')
xcmsLargeWin <- xcmsExamples$xcmsLargeWin
xcmsSmallWin <- xcmsExamples$xcmsSmallWin
ppm <- xcmsExamples$ppm

# detect misaligned features
excluGroups <- misalignDetect(xcmsLargeWin, xcmsSmallWin, ppm)

# obtain the paths of the sample files
filepath <- system.file("extdata", package = "ncGTW")
file <- list.files(filepath, pattern="mzxml", full.names=TRUE)

tempInd <- matrix(0, length(file), 1)
for (n in seq_along(file)){
    tempCha <- file[n]
    tempLen <- nchar(tempCha)
    tempInd[n] <- as.numeric(substr(tempCha, regexpr("example", tempCha) + 7,
        tempLen - 6))
}
# sort the paths by data acquisition order
file <- file[sort.int(tempInd, index.return = TRUE)$ix]

# load the sample profiles
ncGTWinputs <- loadProfile(file, excluGroups)
```

---

meanCorOl *Compute average pairwise correlation and overlapping area*

Description

This function computes average pairwise correlation and overlapping area of each sample pair.

Usage

```
meanCorOl(ncGTWinput, sampleRt)
```

Arguments

`ncGTWinput`	A list in which each element is a `ncGTWinput` object.
`sampleRt`	A list of the same length as the sample number in which each element is a vector corresponding to the sample raw/adjusted RT.

Details

This function computes the pairwise correlation and overlapping area of each sample pair from the input feature, and then takes average.

Value

A list in which the first element is average pairwise correlation, and the second one is average overlapping area.

Examples

```
# obtain data
data('xcmsExamples')
xcmsLargeWin <- xcmsExamples$xcmsLargeWin
xcmsSmallWin <- xcmsExamples$xcmsSmallWin
ppm <- xcmsExamples$ppm

# detect misaligned features
excluGroups <- misalignDetect(xcmsLargeWin, xcmsSmallWin, ppm)

# obtain the paths of the sample files
filepath <- system.file("extdata", package = "ncGTW")
file <- list.files(filepath, pattern="mzxml", full.names=TRUE)

tempInd <- matrix(0, length(file), 1)
for (n in seq_along(file)){
    tempCha <- file[n]
    tempLen <- nchar(tempCha)
    tempInd[n] <- as.numeric(substr(tempCha, regexpr("example", tempCha) + 7,
        tempLen - 6))
}
```

```
# sort the paths by data acquisition order
file <- file[sort.int(tempInd, index.return = TRUE)$ix]

# load the sample profiles
ncGTWinputs <- loadProfile(file, excluGroups)

XCMSCor <- matrix(0, length(ncGTWinputs), 1)
XCMSOl <- matrix(0, length(ncGTWinputs), 1)
for (n in seq_along(ncGTWinputs)){
    XCMSmean <- meanCorOl(ncGTWinputs[[n]],
        slot(xcmsLargeWin, 'rt')$corrected)
    XCMSCor[n] <- XCMSmean$cor
    XCMSOl[n] <- XCMSmean$ol
}
```

`misalignDetect` *Detect misaligned peak groups in xcmsSet object of XCMS*

Description

This function detects the misaligned peak groups with two `xcmsSet-class` object with two different values of bw parameter in `group`.

Usage

```
misalignDetect(xcmsLargeWin, xcmsSmallWin, ppm, qThre = 0.05,
  overlapRate = 0, maxRtWin = 50)
```

Arguments

`xcmsLargeWin`	A `xcmsSet-class` object with a larger bw, usually the maximum expected retension time drift.
`xcmsSmallWin`	A `xcmsSet-class` object with a smaller bw, usually the resolution of the retension time.
`ppm`	Should be set as same as the one when performing the peak detection function in `xcms`.
`qThre`	The threshould of the p-value after multiple test correction. The default is 0.05.
`overlapRate`	The threshould of the overlapping rate of sample index. The default is 0.
`maxRtWin`	The threshould of the maximum retension time range. This is for filtering out some bad groups. The default is 50 (seconds).

Details

This function includes two major steps to determine a peak group is misaligned or not. The first step calculates the p-value of each peak group in xcmsSmallWin, and find the corresponding peak group in xcmsLargeWin. The second step is to find the exclusive peak groups (the groups with no overlapping samples) with adjsted p-values smaller than `qThre`.

Value

A matrix with all detected misaligned peak groups. The column names are the same as `group` slot in `xcmsSet-class`, but the first column is the group index.

Examples

```
# obtain data
data('xcmsExamples')
xcmsLargeWin <- xcmsExamples$xcmsLargeWin
xcmsSmallWin <- xcmsExamples$xcmsSmallWin
ppm <- xcmsExamples$ppm

# detect misaligned features
excluGroups <- misalignDetect(xcmsLargeWin, xcmsSmallWin, ppm)
```

`ncGTWalign` *Run ncGTW alignment*

Description

This function applies ncGTW alignment to the input feature.

Usage

```
ncGTWalign(ncGTWinput, xcmsLargeWin, parSamp = 10, k1Num = 3,
  k2Num = 1, bpParam = BiocParallel::SnowParam(workers = 1),
  ncGTWparam = NULL)
```

Arguments

`ncGTWinput`	A `ncGTWinput` object.
`xcmsLargeWin`	A `xcmsSet-class` object.
`parSamp`	Decide how many samples are in each group when considering parallel computing, and the default is 10.
`k1Num`	Decide how many different k1 will be tested in stage 1. The default is 3.
`k2Num`	Decide how many different k2 will be tested in stage 2. The default is 1.
`bpParam`	A object of **BiocParallel** to control parallel processing, and can be created by `SerialParam`, `MulticoreParam`, or `SnowParam`.
`ncGTWparam`	A `ncGTWparam` object.

Details

This function realign the input feature with ncGTW alignment function with given m/z and RT range.

Value

A `ncGTWoutput` object.

Examples

```
# obtain data
data('xcmsExamples')
xcmsLargeWin <- xcmsExamples$xcmsLargeWin
xcmsSmallWin <- xcmsExamples$xcmsSmallWin
ppm <- xcmsExamples$ppm

# detect misaligned features
excluGroups <- misalignDetect(xcmsLargeWin, xcmsSmallWin, ppm)

# obtain the paths of the sample files
filepath <- system.file("extdata", package = "ncGTW")
file <- list.files(filepath, pattern="mzxml", full.names=TRUE)

tempInd <- matrix(0, length(file), 1)
for (n in seq_along(file)){
    tempCha <- file[n]
    tempLen <- nchar(tempCha)
    tempInd[n] <- as.numeric(substr(tempCha, regexpr("example", tempCha) + 7,
        tempLen - 6))
}
# sort the paths by data acquisition order
file <- file[sort.int(tempInd, index.return = TRUE)$ix]
## Not run:
# load the sample profiles
ncGTWinputs <- loadProfile(file, excluGroups)

# initialize the parameters of ncGTW alignment with default
ncGTWparam <- new("ncGTWparam")

# run ncGTW alignment
ncGTWoutputs <- vector('list', length(ncGTWinputs))
for (n in seq_along(ncGTWinputs))
    ncGTWoutputs[[n]] <- ncGTWalign(ncGTWinputs[[n]], xcmsLargeWin, 5,
        ncGTWparam = ncGTWparam)

## End(Not run)
```

`ncGTWinput-class` *Class "ncGTWinput"*

Description

An S4 class for storing the inputs of ncGTW alignment.

Slots

`groupInfo` A vector of the information of the feature.

`profiles` A raw data matrix in which each row is a sample profile.

`rtRaw` A raw RT matrix in which each row is the corresponding sample RT in `profiles`.

`ncGTWoutput-class` *Class "ncGTWoutput"*

Description

An S4 class for storing the outputs of ncGTW alignment.

Slots

`alignData` A matrix in which each row is a sample profile after downsampling.

`scanRange` A downsampled RT matrix in which each row is the corresponding sample RT in `alignData`.

`path` A list of the same length as the sample number, in which each element is a matrix of the alignment result of the corresponding sample.

`downSample` The factor of downsampling when perform ncGTW alignment.

`ncGTWparam-class` *Class "ncGTWparam"*

Description

An S4 class for storing the needed paramters of ncGTW alignment.

Details

This function initializes the needed paramters of ncGTW alignment with defaults, so this function could be called without any input. The alignment should be fine with all default parameters. If the computing time is an issue, the user could consider increase `downSample` and/or decrease `stpRat` for a faster speed. If the alignment result is not good enough, one can consider increase `strNum` and/or `diaNum` to integrate more neighboring information to increase the quality of alignment, but the speed may drop.

Slots

- `downSample` A factor of downsampling. The larger, the faster speed of alignment, but the accuracy may decrease. The default is 2.
- `stpRat` A factor to control the maximum RT shift of a point in alignment, and the maximum shift is determined by `stpRat` * "The RT range of the feature". If `maxStp` is set, then `stpRat` would be neglected. The default is 0.6.
- `maxStp` A value determines the maximum RT shift of a point in ncGTW alignment. If the user wants to decide the maximum shif by the RT range of the feature, this argument should be NaN. The default is NaN.
- `strNum` A value controls how many neighboring warping functions are connected to each warping function in ncGTW graph. There are two samples corresponding to a warping function, and at least one sample should be the same in another warping function to be considered as a neighbor controlled by `strNum`.
- `diaNum` A value controls how many neighboring warping functions are connected to each warping function in ncGTW graph. There are two samples corresponding to a warping function, and the two samples could also be different to another warping function to be considered as a neighbor controlled by `diaNum`.
- `nor` A value controls p-norm to compute the distance between the points on the profiles, and the default is 1 (Manhattan norm).

`ncGTWwarp-class` *Class "ncGTWwarp"*

Description

An S4 class for storing the realigned RT and the peaks with adjusted RT of ncGTW alignment.

Slots

- `rtncGTW` A list of the same length as the sample number, in which each element is a vector of the realigned RT of the corresponding sample.
- `ncGTWpeaks` A matrix containing peak data with adjusted RT.

`plotGroup` *Plot profiles for each peak group*

Description

This function plots sample profiles with loaded information.

Usage

```
plotGroup(ncGTWinput, sampleRt,
  sampleInd = seq_len(dim(ncGTWinput@rtRaw)[1]), ind = NULL,
  savePath = NULL, show = TRUE, sub = TRUE, filter = FALSE)
```

Arguments

`ncGTWinput`	An object return by `loadProfile` of sample profiles for plotting.
`sampleRt`	A list of the same length as the sample number in which each element is a vector corresponding to the sample raw/adjusted RT for plotting.
`sampleInd`	Indicate which samples should be plotted, and the default is `1:dim(ncGTWinput$rtRaw)[1]`.
`ind`	A user defined index, and the default is `NULL`.
`savePath`	The path to save the plots, and the default is `NULL` (do not save anything).
`show`	Show the plot in R or not, and the default is `TRUE`.
`sub`	Show more information on the plot or not, and the default is `TRUE`.
`filter`	Apply a Gaussian filter for demonstration or not, and the default is `FALSE`.

Details

This function plots the extracted ion chromatogram obtained by `loadProfile`. The user can decide to save the figure, show the figure, or apply a Gaussian filter on the data by parameter setting.

Value

A plot to the current device.

Examples

```
# obtain data
data('xcmsExamples')
xcmsLargeWin <- xcmsExamples$xcmsLargeWin
xcmsSmallWin <- xcmsExamples$xcmsSmallWin
ppm <- xcmsExamples$ppm

# detect misaligned features
excluGroups <- misalignDetect(xcmsLargeWin, xcmsSmallWin, ppm)

# obtain the paths of the sample files
filepath <- system.file("extdata", package = "ncGTW")
file <- list.files(filepath, pattern="mzxml", full.names=TRUE)

tempInd <- matrix(0, length(file), 1)
for (n in seq_along(file)){
    tempCha <- file[n]
    tempLen <- nchar(tempCha)
    tempInd[n] <- as.numeric(substr(tempCha, regexpr("example", tempCha) + 7,
        tempLen - 6))
}
# sort the paths by data acquisition order
file <- file[sort.int(tempInd, index.return = TRUE)$ix]

# load the sample profiles
ncGTWinputs <- loadProfile(file, excluGroups)

# plot all loaded features
```

```
for (n in seq_along(ncGTWinputs))
    plotGroup(ncGTWinputs[[n]], slot(xcmsLargeWin, 'rt')$corrected)
```

---

rtncGTW,ncGTWwarp-method

ncGTWwarp-accessors

Description

Accessors to the realigned RT and the peaks with adjusted RT of ncGTW alignment.

Usage

```
## S4 method for signature 'ncGTWwarp'
rtncGTW(object)

## S4 method for signature 'ncGTWwarp'
ncGTWpeaks(object)
```

Arguments

`object`	a `ncGTWwarp` object.

Value

`rtncGTW` returns a list of the same length as the sample number, in which each element is a vector of the realigned RT of the corresponding sample.

`rtncGTW` returns a matrix containing peak data with adjusted RT.

Examples

```
# obtain data
data('xcmsExamples')
xcmsLargeWin <- xcmsExamples$xcmsLargeWin
xcmsSmallWin <- xcmsExamples$xcmsSmallWin
ppm <- xcmsExamples$ppm

# detect misaligned features
excluGroups <- misalignDetect(xcmsLargeWin, xcmsSmallWin, ppm)

# obtain the paths of the sample files
filepath <- system.file("extdata", package = "ncGTW")
file <- list.files(filepath, pattern="mzxml", full.names=TRUE)

tempInd <- matrix(0, length(file), 1)
for (n in seq_along(file)){
    tempCha <- file[n]
    tempLen <- nchar(tempCha)
    tempInd[n] <- as.numeric(substr(tempCha, regexpr("example", tempCha) + 7,
```

```
        tempLen - 6))
}
# sort the paths by data acquisition order
file <- file[sort.int(tempInd, index.return = TRUE)$ix]
## Not run:
# load the sample profiles
ncGTWinputs <- loadProfile(file, excluGroups)

# initialize the parameters of ncGTW alignment with default
ncGTWparam <- initncGTWparam()

# run ncGTW alignment
ncGTWoutputs <- vector('list', length(ncGTWinputs))
ncGTWoutputs[[1]] <- ncGTWalign(ncGTWinputs[[1]], xcmsLargeWin, 5,
        ncGTWparam = ncGTWparam)

# adjust RT with the realignment results from ncGTW
ncGTWres <- xcmsLargeWin
adjustRes <- adjustRT(ncGTWres, ncGTWinputs[[1]], ncGTWoutputs[[1]], ppm)
rt <- rtncGTW(adjustRes)
peaks <- ncGTWpeaks(adjustRes)

## End(Not run)
```

`xcmsExamples` *Examples of xcmsSet for inputs of ncGTW*

Description

These two `xcmsSet-class` objects are created from the example dataset attached in this package (in extdata folder) with XCMS package. They are examples for inputs of `misalignDetect`.

Usage

```
data(xcmsExamples)
```

Format

A list with two `xcmsSet-class` objects and one `ppm` parameter.

Details

The example dataset contains 20 samples picked from an in-house LC-MS dataset. The user can replicate the two `xcmsSet-class` objects with `xcmsSet`, `retcor-methods`, and `group-methods`. For details, please refer to the user manual.

Index