Package ‘MassSpecWavelet’

October 18, 2022

Type Package

Title Mass spectrum processing by wavelet-based algorithms

Version 1.62.0

Encoding UTF-8

Date 2022-04-05

Description Processing Mass Spectrometry spectrum by using wavelet based algorithm.

URL https://github.com/zeehio/MassSpecWavelet

BugReports http://github.com/zeehio/MassSpecWavelet/issues

License LGPL (>= 2)

Suggests signal, waveslim, BiocStyle, knitr, rmarkdown

biocViews ImmunoOncology, MassSpectrometry, Proteomics

VignetteBuilder knitr

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

git_branch RELEASE_3_15

git_last_commit 12e47e6

git_last_commit_date 2022-04-26

Date/Publication 2022-10-18

Author Pan Du [aut],
Warren Kibbe [aut],
Simon Lin [aut],
Sergio Oller Moreno [cre] (<https://orcid.org/0000-0002-8994-1549>)

Maintainer Sergio Oller Moreno <sergioller@gmail.com>

R topics documented:

`MassSpecWavelet-package`

Peak detection of mass spectrum by Wavelet transform based methods

---

Description

Process Mass Spectrum (MS) by Wavelet Transforms-based algorithms

Details

Package:	MassSpecWavelet
Type:	Package
Version:	1.0.4
Date:	2007-04-05
License:	GPL 2 or newer

MassSpecWavelet R package is aimed to process Mass Spectrometry (MS) data mainly based on Wavelet Transforms. The current version only supports the peak detection based on Continuous Wavelet Transform (CWT). Future versions will include more functions covering entire MS data processes.

Author(s)

Pan Du, Simon Lin

Maintainer: Pan Du <dupan@northwestern.edu>

References

Du, P., Kibbe, W.A. and Lin, S.M. (2006) Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching, Bioinformatics, 22, 2059-2065.

Examples

```
data(exampleMS)
SNR.Th <- 3
peakInfo <- peakDetectionCWT(exampleMS, SNR.Th=SNR.Th)
majorPeakInfo = peakInfo$majorPeakInfo
peakIndex <- majorPeakInfo$peakIndex
plotPeak(exampleMS, peakIndex, main=paste('Identified peaks with SNR >', SNR.Th))
```

`cwt` *Continuous Wavelet Transform (CWT)*

Description

CWT(Continuous Wavelet Transform) with Mexican Hat wavelet (by default) to match the peaks in Mass Spectrometry spectrum

Usage

```
cwt(ms, scales = 1, wavelet = "mexh")
```

Arguments

`ms`	Mass Spectrometry spectrum (a vector of MS intensities)
`scales`	a vector represents the scales at which to perform CWT.
`wavelet`	The wavelet base, Mexican Hat by default. User can provide wavelet Psi(x) as a form of two row matrix. The first row is the x value, and the second row is Psi(x) corresponding to x.

Value

The return is the 2-D CWT coefficient matrix, with column names as the scale. Each column is the CWT coefficients at that scale.

Author(s)

Pan Du, Simon Lin

Examples

```
data(exampleMS)
scales <- seq(1, 64, 3)
wCoefs <- cwt(exampleMS[5000:11000], scales=scales, wavelet='mexh')

## Plot the 2-D CWT coefficients as image (It may take a while!)
xTickInterval <- 1000
image(5000:11000, scales, wCoefs, col=terrain.colors(256), axes=FALSE,
          xlab='m/z index', ylab='CWT coefficient scale', main='CWT coefficients')
axis(1, at=seq(5000, 11000, by=xTickInterval))
axis(2, at=c(1, seq(10, 64, by=10)))
box()
```

exampleMS *An example mass spectrum*

Description

An example mass spectrum from CAMDA 2006. All-in-1 Protein Standard II (Ciphergen Cat. \# C100-0007) were measured on Ciphergen NP20 chips. There are 7 polypeptides in the sample with m/z values of 7034, 12230, 16951, 29023, 46671, 66433, 147300.

Usage

```
data(exampleMS)
```

Format

A numeric vector represents the mass spectrum with equal sample intervals.

Source

CAMDA, CAMDA 2006 Competition Data Set. 2006, http://camda.duke.edu.

extendLength *Extend the length of a signal or matrix*

Description

Extend the length of a signal or matrix by row

Usage

```
extendLength(
 x,
 addLength = NULL,
 method = c("reflection", "open", "circular"),
 direction = c("right", "left", "both")
)
```

Arguments

x — a vector or matrix with column with each column as a signal

addLength — the length to be extended

method — three methods available, c("reflection", "open", "circular"). By default, it is "reflection".

direction — three options available: c("right", "left", "both")

Value

return the extended vector or matrix.

Author(s)

Pan Du

See Also

extendNBase

Examples

```
# a = matrix(rnorm(9), 3)
# extendLength(a, 3, direction='right')  ## not exposed function
```

extendNBase *Extend the row number of a matrix as the exponential of base N*

Description

Extend the data as the exponential of base N by increasing row number.

Usage

```
extendNBase(x, nLevel=1, base=2, ...)
```

Arguments

x — data matrix

nLevel — the level of DWT decomposition. Basically, it is equivalent to changing the 'base' as base\^nLevel

base — the base, 2 by default

... — other parameters of used by extendLength

Details

The method 'open' is padding the the matrix with the last row.

Value

Return a extended matrix

Author(s)

Pan Du

See Also

extendLength

Examples

```
# a = matrix(rnorm(9), 3)
# extendNBase(a) ## not exposed function
```

---

getLocalMaximumCWT   *Identify the local maximum of each column in 2-D CWT coefficients matrix*

Description

Identify the local maximum of each column in 2-D CWT coefficients matrix by using a slide window. The size of slide window linearly changes from the coarse scale (bigger window size) to detail scale. The scale of CWT increases with the column index.

Usage

```
getLocalMaximumCWT(wCoefs, minWinSize= 5, amp.Th = 0)
```

Arguments

`wCoefs` 2-D CWT coefficients, each column corresponding to CWT coefficient at one scale. The column name is the scale.

`minWinSize` The minimum slide window size used.

`amp.Th` The minimum peak amplitude.

Value

return a matrix with same dimension as CWT coefficient matrix, wCoefs. The local maxima are marked as 1, others are 0.

Author(s)

Pan Du

See Also

localMaximum

Examples

```
data(exampleMS)
scales <- seq(1, 64, 3)
wCoefs <- cwt(exampleMS[5000:11000], scales=scales, wavelet='mexh')

localMax <- getLocalMaximumCWT(wCoefs)
plotLocalMax(localMax)
```

getRidge *Identify ridges based on the local maximum matrix*

Description

Identify ridges by connecting the local maximum of 2-D CWT coefficients from the coarse scale to detail scale. The local maximum matrix is returned from getLocalMaximumCWT

Usage

```
getRidge(
 localMax,
 iInit = ncol(localMax),
 step = -1,
 iFinal = 1,
 minWinSize= 5,
 gapTh = 3,
 skip = NULL
)
```

Arguments

localMax	The local maximum matrix is returned from getLocalMaximumCWT with 1 represents maximum, others are 0.
iInit	The start column to search ridge. By default, it starts from the coarsest scale level.
step	Search step. -1 by default, which means searching from coarse scale to detail scale column by column.
iFinal	The final column index of search ridge.
minWinSize	The minimum slide window size used.
gapTh	The gap allowed during searching for ridge. 3 by default.
skip	The column to be skipped during search.

Value

Return a list of ridge. As some ridges may end at the scale larger than 1, in order to keep the uniqueness of the ridge names, we combined the smallest scale of the ridge and m/z index of the peak at that scale together to name the ridges. For example the ridge name "1_653" means the peak ridge ends at the CWT scale 1 with m/z index 653 at scale 1.

Author(s)

Pan Du, Simon Lin

References

Du, P., Kibbe, W.A. and Lin, S.M. (2006) Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching, Bioinformatics, 22, 2059-2065.

See Also

`getLocalMaximumCWT`, `identifyMajorPeaks`

Examples

```
data(exampleMS)
scales <- seq(1, 64, 3)
wCoefs <- cwt(exampleMS[5000:11000], scales=scales, wavelet='mexh')

localMax <- getLocalMaximumCWT(wCoefs)
ridgeList <- getRidge(localMax)
plotRidgeList(ridgeList)
```

`getRidgeLength` *Estimate the length of the ridge*

Description

Estimate the length of the ridge line, which is composed of local maxima at adjacent CWT scales. The ridge line is cut off at the end point, whose amplitude divided by the maximum ridge amplitude is larger than the cutoff amplitude ratio threshold (0.5 by default).

Usage

```
getRidgeLength(ridgeList, Th = 0.5)
```

Arguments

`ridgeList`	a list of identified ridges
`Th`	the cutoff amplitude ratio (the amplitude divided by the maximum amplitude of the ridge) threshold of the ridge line end.

Value

a vector of estimated ridge length

Author(s)

Pan Du

getRidgeValue *Get the CWT coefficient values corresponding to the peak ridge*

Description

Get the CWT coefficient values corresponding to the peak ridge

Usage

```
getRidgeValue(ridgeList, wCoefs, skip = 0)
```

Arguments

`ridgeList`	a list of ridge lines
`wCoefs`	2-D CWT coefficients
`skip`	the CWT scale level to be skipped, by default the 0 scale level (raw spectrum) is skipped.

Value

A list of ridge values corresponding to the input ridgeList.

Author(s)

Pan Du

identifyMajorPeaks *Identify peaks based on the ridges in 2-D CWT coefficient matrix*

Description

Indentify the peaks based on the ridge list (returned by `getRidge`) in 2-D CWT coefficient matrix and estimated Signal to Noise Ratio (SNR)

Usage

```
identifyMajorPeaks(
  ms,
  ridgeList,
  wCoefs,
  scales = as.numeric(colnames(wCoefs)),
  SNR.Th = 3,
  peakScaleRange = 5,
  ridgeLength = 32,
  nearbyPeak = FALSE,
  nearbyWinSize = 100,
  winSize.noise = 500,
  SNR.method = "quantile",
  minNoiseLevel = 0.001
)
```

Arguments

`ms`	the mass spectrometry spectrum
`ridgeList`	returned by `getRidge`
`wCoefs`	2-D CWT coefficients
`scales`	scales of CWT, by default it is the colnames of wCoefs
`SNR.Th`	threshold of SNR
`peakScaleRange`	the CWT scale range of the peak.
`ridgeLength`	the maximum ridge scale of the major peaks.
`nearbyPeak`	determine whether to include the small peaks close to large major peaks
`nearbyWinSize`	the window size to determine the nearby peaks. Only effective when nearbyPeak is true.
`winSize.noise`	the local window size to estimate the noise level.
`SNR.method`	method to estimate noise level. Currently, only 95 percentage quantile is supported.
`minNoiseLevel`	the minimum noise level used in calculating SNR, i.e., if the estimated noise level is less than "minNoiseLevel", it will use "minNoiseLevel" instead. If the noise level is less than 0.5, it will be treated as the ratio to the maximum amplitude of the spectrum.

Details

The determination of the peaks is based on three rules: Rule 1: The maximum ridge scale of the peak should larger than a certain threshold Rule 2: Based on the scale of the peak (corresponding to the maximum value of the peak ridge) should be within certain range Rule 3: Based on the peak SNR

Value

Return a list with following elements:

`peakIndex` the m/z indexes of the identified peaks

`peakCenterIndex`
the m/z indexes of peak centers, which correspond to the maximum on the ridge. peakCenterIndex includes all the peaks, not just the identified major peaks.

`peakCenterValue`
the CWT coefficients (the maximum on the ridge) corresponding to peakCenterIndex

`peakSNR` the SNR of the peak, which is the ratio of peakCenterValue and noise level

`peakScale` the estimated scale of the peak, which corresponds to the peakCenerIndex

`potentialPeakIndex`
the m/z indexes of all potential peaks, which satisfy all requirements of a peak without considering its SNR. Useful, if you want to change to a lower SNR threshold later.

`allPeakIndex` the m/z indexes of all the peaks, whose order is the same as peakCenterIndex, peakCenterValue, peakSNR and peakScale.

All of these return elements have peak names, which are the same as the corresponding peak ridges. see `getRidge` for details.

Author(s)

Pan Du, Simon Lin

References

Du, P., Kibbe, W.A. and Lin, S.M. (2006) Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching, Bioinformatics, 22, 2059-2065.

See Also

`peakDetectionCWT`, `tuneInPeakInfo`

Examples

```
data(exampleMS)
scales <- seq(1, 64, 3)
wCoefs <- cwt(exampleMS, scales=scales, wavelet='mexh')

localMax <- getLocalMaximumCWT(wCoefs)
ridgeList <- getRidge(localMax)

SNR.Th <- 3
majorPeakInfo <- identifyMajorPeaks(exampleMS, ridgeList, wCoefs, SNR.Th=SNR.Th)
## Plot the identified peaks
peakIndex <- majorPeakInfo$peakIndex
plotPeak(exampleMS, peakIndex, main=paste('Identified peaks with SNR >', SNR.Th))
```

localMaximum *Identify local maximum within a slide window.*

Description

Find local maximum by transform the vector as matrix, then get the the maximum of each column. This operation is performed twice with vecctor shifted half of the winSize.

Usage

```
localMaximum(x, winSize = 5)
```

Arguments

x	a vector represents a signal profile
winSize	the slide window size, 5 by default.

Details

Instead of find the local maximum by a slide window, which slide all possible positions, we find local maximum by transform the vector as matrix, then get the the maximum of each column. This operation is performed twice with vecctor shifted half of the winSize. The main purpose of this is to increase the efficiency of the algorithm.

Value

Return a vector with the same length of the input x. The position of local maximum is set as 1, 0 else where.

Author(s)

Pan Du

See Also

getLocalMaximumCWT

Examples

```
x <- rnorm(200)
lmax <- localMaximum(x, 5)
maxInd <- which(lmax > 0)
plot(x, type='l')
points(maxInd, x[maxInd], col='red')
```

`mzInd2vRange` *Match m/z index to m/z value with a certain error range*

Description

Match m/z index to m/z value with a certain error range

Usage

```
mzInd2vRange(mzInd, error = 0.003)
```

Arguments

`mzInd`	a vector of m/z index
`error`	error range

Value

return a vector of sorted m/z values

Author(s)

Pan Du

See Also

`mzV2indRange`

`mzV2indRange` *Match m/z value to m/z index with a certain error range*

Description

Match m/z value to m/z index with a certain error range

Usage

```
mzV2indRange(mzV, error = 0.003)
```

Arguments

`mzV`	a vector of m/z value
`error`	error range

Value

return a vector of sorted m/z indexes

Author(s)

Pan Du

See Also

`mzInd2vRange`

`peakDetectionCWT` *The main function of peak detection by CWT based pattern matching*

Description

This function is a wrapper of `cwt`, `getLocalMaximumCWT`, `getRidge`, `identifyMajorPeaks`

Usage

```
peakDetectionCWT(
 ms,
 scales = c(1, seq(2, 30, 2), seq(32, 64, 4)),
 SNR.Th = 3,
 nearbyPeak = TRUE,
 peakScaleRange = 5,
 amp.Th = 0.01,
 minNoiseLevel = amp.Th/SNR.Th,
 ridgeLength = 24,
 peakThr=NULL,
 tuneIn = FALSE,
 ...
)
```

Arguments

`ms` the mass spectrometry spectrum

`scales` scales of CWT

`SNR.Th` SNR (Signal to Noise Ratio) threshold

`nearbyPeak` Determine whether to include the nearby small peaks of major peaks. `TRUE` by default

`peakScaleRange` the scale range of the peak. larger than 5 by default.

`amp.Th` the minimum required relative amplitude of the peak (ratio to the maximum of CWT coefficients)

`minNoiseLevel` the minimum noise level used in computing the SNR `ridgeLength` the minimum highest scale of the peak in 2-D CWT coefficient matrix

`peakThr` Minimal absolute intensity (above the baseline) of peaks to be picked. If this value is provided, then the smoothing function `sgolayfilt` will be called to estimate the local intensity.(added based on the suggestion and code of Steffen Neumann)

`tuneIn` determine whether to tune in the parameter estimation of the detected peaks

`...` other parameters used by `identifyMajorPeaks`. Additionally, `fl` (filter length, with a default value of 1001) and `forder` (filter order, with a default value of 2) are set and passed to `sgolayfilt` when peakThr is given.

Value

`majorPeakInfo` return of `identifyMajorPeaks`

`ridgeList` return of `getRidge`

`localMax` return of `getLocalMaximumCWT`

`wCoefs` 2-D CWT coefficient matrix, see `cwt` for details.

Author(s)

Pan Du, Simon Lin

References

Du, P., Kibbe, W.A. and Lin, S.M. (2006) Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching, Bioinformatics, 22, 2059-2065.

See Also

`cwt`, `getLocalMaximumCWT`, `getRidge`, `identifyMajorPeaks`

Examples

```
data(exampleMS)
SNR.Th <- 3
peakInfo <- peakDetectionCWT(exampleMS, SNR.Th=SNR.Th)
majorPeakInfo = peakInfo$majorPeakInfo
peakIndex <- majorPeakInfo$peakIndex
plotPeak(exampleMS, peakIndex, main=paste('Identified peaks with SNR >', SNR.Th))

## In some cases, users may want to add peak filtering based on the absolute peak amplitude
peakInfo <- peakDetectionCWT(exampleMS, SNR.Th=SNR.Th, peakThr=500)
majorPeakInfo = peakInfo$majorPeakInfo
peakIndex <- majorPeakInfo$peakIndex
plotPeak(exampleMS, peakIndex, main=paste('Identified peaks with SNR >', SNR.Th))
```

plotLocalMax   *Plot the local maximum matrix*

Description

Plot the local maximum matrix of 2-D CWT coefficients returned by getLocalMaximumCWT

Usage

```
plotLocalMax(
 localMax,
 wCoefs = NULL,
 range = c(1, nrow(localMax)),
 colorMap = "RYB",
 main = NULL, cex = 3, pch = ".", ...
)
```

Arguments

localMax	local maximum matrix of 2-D CWT coefficients returned by getLocalMaximumCWT
wCoefs	2-D CWT coefficients
range	plot range of m/z index
colorMap	the colormap used in plotting the points
main	parameter of plot
cex	parameter of plot
pch	parameter of plot
...	other parameters of points

Value

No value is returned; this function is called for its side effects (plot).

Author(s)

Pan Du

See Also

getLocalMaximumCWT

Examples

```
data(exampleMS)
scales <- seq(1, 64, 3)
wCoefs <- cwt(exampleMS[5000:11000], scales=scales, wavelet='mexh')

localMax <- getLocalMaximumCWT(wCoefs)
plotLocalMax(localMax)
```

plotPeak *Plot the identified peaks over the spectrum*

Description

Plot the identified peaks over the spectrum. The identified peaks are returned by `peakDetectionCWT` or `identifyMajorPeaks`

Usage

```
plotPeak(ms, peakIndex = NULL, mz = 1:length(ms),
       range = c(min(mz), max(mz)), method = c("p", "l"), main = NULL, log = "", ...
)
```

Arguments

`ms`	the MS spectrum
`peakIndex`	m/z indexes of the identified peaks
`mz`	m/z value correspond to m/z index
`range`	the plot range of m/z value
`method`	plot method of the identified peaks. method 'p' plot circles on the peaks; method 'l' add vertical lines over the peaks.
`main`	parameter of `plot`
`log`	parameter of `plot`
`...`	other parameters of `points`

Value

No value is returned; this function is called for its side effects (plot).

Author(s)

Pan Du

See Also

`peakDetectionCWT`, `identifyMajorPeaks`

Examples

```
data(exampleMS)
SNR.Th <- 3
peakInfo <- peakDetectionCWT(exampleMS, SNR.Th=SNR.Th)
majorPeakInfo = peakInfo$majorPeakInfo
peakIndex <- majorPeakInfo$peakIndex
plotPeak(exampleMS, peakIndex, main=paste('Identified peaks with SNR >', SNR.Th))
```

plotRidgeList *Plot the ridge list*

Description

Plot the ridge list returned by getRidge

Usage

```
plotRidgeList(ridgeList, wCoefs = NULL, range = NULL,
               colorMap = ”RYB”, main = NULL, pch = ”.”, cex = 3, ...)
```

Arguments

ridgeList	returned by getRidge
wCoefs	2-D CWT coefficients
range	plot range of m/z index
colorMap	colorMap to plot the points of local maximum
main	parameter of plot
pch	parameter of plot
cex	parameter of plot
...	other parameters of points

Value

No value is returned; this function is called for its side effects (plot).

Author(s)

Pan Du

See Also

getRidge

Examples

```
data(exampleMS)
scales <- seq(1, 64, 3)
wCoefs <- cwt(exampleMS[5000:11000], scales=scales, wavelet='mexh')

localMax <- getLocalMaximumCWT(wCoefs)
ridgeList <- getRidge(localMax)
plotRidgeList(ridgeList)
```

smoothDWT *smooth (denoise) the spectrum by DWT (Discrete Wavelet Transform)*

Description

Smooth (denoise) the spectrum by DWT (Discrete Wavelet Transform)

Usage

```
smoothDWT(ms, nLevel = 6, wf = "la8", localNoiseTh = seq(1, 0, by = -0.2),
       localWinSize = 500, globalNoiseTh = 0.75, smoothMethod = c("soft", "hard"),
          method = c('dwt', 'modwt'))
```

Arguments

ms	a vector representing the mass spectrum
nLevel	the level of DWT decomposition
wf	the name of wavelet for DWT
localNoiseTh	local noise level threshold
localWinSize	local window size for estimate local noise threshold
globalNoiseTh	global noise level threshold
smoothMethod	the method used for denoising. 'hard' means keeping the dwt coefficients higher than the threshold unchanged; "soft" means the dwt coefficients higher than the threshold were subtracted by the threshold.
method	'dwt' or 'modwt' used for decomposition

Value

return the smoothed mass spectrum with the 'detail' component of DWT as an attribute 'detail'.

Author(s)

Pan Du

tuneInPeakInfo *Tune in the peak information: peak position and peak scale*

Description

Based on the identified peak position, more precise estimation of the peak information, i.e., peak position and peak scale, can be got by this function. The basic idea is to cut the segment of spectrum near the identified peaks, and then do similar procedures as peakDetectionCWT, but with more detailed scales around the estimated peak scale.

Usage

```
tuneInPeakInfo(ms, majorPeakInfo = NULL, peakIndex = NULL,
               peakScale = NULL, maxScale = 128, ...)
```

Arguments

ms	the mass spectrometry spectrum
majorPeakInfo	return of identifyMajorPeaks
peakIndex	the m/z index of the identified peaks
peakScale	the scales of the identified peaks
maxScale	the maximum scale allowed for the peak
...	other parameters of used by getLocalMaximumCWT, getRidge, identifyMajorPeaks

Details

The majorPeakInfo or peakIndex and peakScale must be provided.

Value

peakCenterIndex	the updated peak center m/z index
peakScale	the updated peak scale
peakValue	the corresponding peak value

Author(s)

Pan Du

References

Du, P., Kibbe, W.A. and Lin, S.M. (2006) Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching, Bioinformatics, 22, 2059-2065.

See Also

peakDetectionCWT

Examples

```
data(exampleMS)
SNR.Th <- 3
peakInfo <- peakDetectionCWT(exampleMS, SNR.Th=SNR.Th)
majorPeakInfo <- peakInfo$majorPeakInfo
betterPeakInfo <- tuneInPeakInfo(exampleMS, majorPeakInfo)
plot(500:length(exampleMS), exampleMS[500:length(exampleMS)], type='l', log='x')
abline(v=betterPeakInfo$peakCenterIndex, col='red')
```

Index