Package 'IDSL.IPA'

September 9, 2022

Type Package

Title Intrinsic Peak Analysis (IPA) for HRMS Data

Version 2.2

Depends R (>= 4.0)

Imports IDSL.MXP (>= 1.4), xml2, RNetCDF, base64enc, grid, readxl, parallel, doParallel, foreach, ggplot2, gridExtra, png

Author Sadjad Fakouri-Baygi [cre, aut] (<https://orcid.org/0000-0002-6864-6911>), Dinesh Barupal [aut] (<https://orcid.org/0000-0002-9954-8628>)

Maintainer Sadjad Fakouri-Baygi <sadjad.fakouri-baygi@mssm.edu>

Description A sophisticated pipeline for processing LC/HRMS data to extract signals of organic compounds. The package performs isotope pairing, peak detection, alignment, RT correction, gap filling, peak annotation and visualization of extracted ion chromatograms (EIC) and total ion chromatograms (TIC).

License MIT + file LICENSE

URL https://ipa.idsl.me, https://github.com/idslme/idsl.ipa

BugReports https://github.com/idslme/idsl.ipa/issues

Encoding UTF-8

LazyData true

Archs i386, x64

NeedsCompilation no

Repository CRAN

Date/Publication 2022-09-09 17:02:58 UTC

R topics documented:

asymmetry_factor *Asymmetry factor for a chromatographic peak*

Description

This function calculates an asymmetry factor for a chromatographic peak.

Usage

```
asymmetry_factor(rt, int)
```

Arguments

rt	a vector of retention times for the chromatographic peak.
int	a vector of intensities corresponding to the vector of retention times for the chromatographic peak.

Value

asymmetry of the chromatographic peak. 1 is for very symmetric peak.

Examples

```
data(peak_spline)
rt <- peak_spline[, 1]
int <- peak_spline[, 2] - peak_spline[, 3]
asymmetry_factor(rt, int)
```

baseline_developer *Develop a baseline for the chromatogram using local minima*

Description

This function generates a vector of baselines for the chromatogram using local minima. It also is capable of excluding outlier local minima to generate a realistic baseline including true baseline regions. This baseline may represent the local noise levels for the chromatogram.

Usage

```
baseline_developer(segment, int)
```

Arguments

segment	a matrix or a vecotr of adjusted scan number of local minima w/ or w/o redundant local minima. Adjusted scan numbers are the scan numbers but adjusted to start at 1.
int	a vector of intensities of the chromatogram.

Value

A vector of baselines in the same size of the "int" vector.

Examples

```
data(segment)
data(chromatogram_builder)
int <- chromatogram_builder$SmoothedChromatogram
baseline_developer(segment, int)
```

`chromatogram_builder` *chromatogram builder for m/z = 263.1678 in 003.d from cord blood sample*

Description

illusterates a chromatogram and baseline vectors to indicate chromatogram development.

Usage

```
data("chromatogram_builder")
```

Format

A data frame with 219 observations on the following 6 variables.

`ScanNumber` a numeric vector

`RetentionTIme` a numeric vector

`SmoothedChromatogram` a numeric vector

`RawChromatogram` a numeric vector

`'12C/13C Isotopologue Pairs'` a numeric vector

`Baseline` a numeric vector

Examples

```
data(chromatogram_builder)
```

chromatography_analysis
Chromatography analysis

Description

This function detects individual chromatographic peaks and measures their peak qualification metrics.

Usage

```
chromatography_analysis(spec_scan_xic, smoothing_window,
peak_resolving_power, min_nIsoPair, min_peak_height,
min_ratio_IsoPair, max_rpw, min_snr_baseline,
max_R13C_integrated_peak, max_percentage_missing_scans,
mz_target, rt_target = 0, mass_accuracy_xic, spectraList,
RetentionTime, n_spline)
```

Arguments

spec_scan_xic	a matrix consists of 5 columns. The column contents are the m/z of 12C isotopologues, intensity of 12C isotopologues, scan number (t), m/z of 13C isotopologues, and intensity of 13C isotopologues, respectively.
smoothing_window	number of scans for peak smoothing
peak_resolving_power	a value to represent peak resolving power
min_nIsoPair	minimum number of nIsoPair for an individual peak
min_peak_height	minimum peak height for an individual peak
min_ratio_IsoPair	minimum ratio of nIsoPair per number of available scans within an individual peak
max_rpw	maximum allowed value of ratio of peak width at half-height to baseline (RPW) for an individual peak
min_snr_baseline	minimum S/N baseline for an individual peak
max_R13C_integrated_peak	maximum allowed value of average R13C for an individual peak
max_percentage_missing_scans	maximum allowed value of percentage missing scans on the raw chromatogram for an individual peak.
mz_target	m/z value to perform chromatography analysis

rt_target	retention time value for a targeted peak to calculate the ancillary chromatography parameters. When this parameter set at 0, the ancillary chromatography parameters are calculated for the entire detected peaks.
mass_accuracy_xic	mass error to perform chromatography analysis
spectraList	a list of mass spectra in each chromatogram scan
RetentionTime	a vector of retention times vs. corresponding scan numbers
n_spline	number of points for further smoothing using a cubic spline smoothing method to calculate ancillary chromatographic parameters

Value

a data frame consisting of 24 columns representing chromatography and mass spectrometry parameters. Each row represents an individual separated chromatographic peak.

---

derivative_skewness *Derivative skewness*

Description

This function calculates skewness of a chromatographic peak using first order degree of numerical differentiation.

Usage

```
derivative_skewness(rt, int)
```

Arguments

rt	a vector representing retention times of the chromatographic peak.
int	a vector representing intensities of the chromatographic peak.

Value

Skewness of a chromatographic peak. 1 is for very symmetric peak. Minimum is 0 from this function.

Examples

```
data(peak_spline)
rt <- peak_spline[, 1]
int <- peak_spline[, 2]
derivative_skewness(rt, int)
```

der_5points_stencil *Numerical differentiation by five-point stencil method*

Description

This module performs numerical differentiation using the five-point stencil method.

Usage

```
der_5points_stencil(x, y, n)
```

Arguments

x	a vector of values for x.
y	a vector of values for y.
n	order of numerical differentiation (n=1-4).

Value

A matrix of 2 columns. The first column represents x and the second column represents numerical differentiation values. This matrix has four rows (two rows from the beginning and 2 rows from the end) less than length of x or y.

Examples

```
data(peak_spline)
rt <- peak_spline[, 1]
int <- peak_spline[, 2]
n <- 2 # second order derivative
der_5points_stencil(rt, int, n)
```

EIC_plotter *EIC plotter*

Description

This function plots the EIC figure and annexes the chromatographic properties to the EIC figures.

Usage

```
EIC_plotter(spec_scan_xic, peak_property_xic, smoothing_window,
peak_resolving_power, mass_accuracy_xic, spectraList, RetentionTime,
mz_target, rt_target, file_name, legend_EIC)
```

Arguments

spec_scan_xic	a matrix consists of 5 columns. The column contents are the m/z of 12C isotopologues, intensity of 12C isotopologues, scan number (t), m/z of 13C isotopologues, and intensity of 13C isotopologues.
peak_property_xic	a data frame representing chromatographic peak properties.
smoothing_window	number of scans for peak smoothing.
peak_resolving_power	a value to represent peak resolving power.
mass_accuracy_xic	a mass accuracy value to perform chromatography analysis.
spectraList	a list of mass spectra in each chromatogram scan.
RetentionTime	a vector of retention times vs. corresponding scan numbers.
mz_target	an m/z value to perform chromatography analysis.
rt_target	the retention time value of the candidate peak.
file_name	name of HRMS file used for peak construction.
legend_EIC	A file to attach the legends on the EIC figures.

Value

A figure to show the EIC and its property table.

---

fronting_tailing_resolver

Fronting and tailing peaks resolver

Description

This function attempts to resolve peak tailings or frontings into the main peak in case they were detected as seperate peaks.

Usage

```
fronting_tailing_resolver(segment, int, max_space, peak_resolving_power)
```

Arguments

segment	a matrix or a vector of peak boundaries.
int	a vector of intensities of the entire chromatogram.
max_space	maximum scan number difference between peak tailing or fronting and the main peak.
peak_resolving_power	power of peak resolving tool.

Value

A matrix of 2 columns. Each row indicates peak boundary indices on the 'int' vector after resolving fronting and tailing peaks.

Examples

```
data(segment)
data(chromatogram_builder)
int <- chromatogram_builder$SmoothedChromatogram
max_space <- 7
peak_resolving_power <- 0.2
fronting_tailing_resolver(segment, int, max_space, peak_resolving_power)
```

---

gaussianity_measurement

gaussianity measurement

Description

This module measures gaussianity of chromatographic peak using Pearson correlation coefficients (ρ) at top 80 percent of peak.

Usage

```
gaussianity_measurement(RT, Int, BL, gauge = 0.8)
```

Arguments

RT	a vector of retention times of the chromatographic peak.
Int	a vector of intensities of the chromatographic peak.
BL	a vector of baseline of the chromatographic peak.
gauge	represents the gauge height of peak for Gaussianity measurement.

Value

Gaussianity of the chromatographic peak.

Examples

```
data("peak_spline")
RT <- peak_spline[, 1]
Int <- peak_spline[, 2]
BL <- peak_spline[, 3]
gaussianity_measurement(RT, Int, BL, gauge = 0.8)
```

IPA_CompoundsAnnotation *Compound-centric peak annotation*

Description

This function performs compound-centric peak annotation.

Usage

```
IPA_CompoundsAnnotation(PARAM)
```

Arguments

PARAM	a data frame from IPA_xlsxAnalyzer function containing the IPA parameters.

Value

This function saves individual .CSV files for each compound in the "compound_centeric_annotation" folder.

IPA_GapFiller *IPA GapFiller*

Description

This function fills the gaps on the peak table.

Usage

```
IPA_GapFiller(PARAM)
```

Arguments

PARAM	a data frame from the 'IPA_xlsxAnalyzer' function containing the IPA parameters.

Value

This function saves individual .CSV and .Rdata files for the gap-filled peak tables for peak height, area, and R13C properties in the "peak_alignment" folder.

IPA_IsotopePairing *IPA Isotope Pairing*

Description

This function pairs isotopologue in high-resolution mass spectral datasets

Usage

```
IPA_IsotopePairing(spectraList, int_threshold, mass_accuracy_isotope_pair,
massDifferenceIsotopes)
```

Arguments

spectraList	list of mass spectra in each chromatogram scan
int_threshold	intensity threshold at each chromatogram scan
mass_accuracy_isotope_pair	mass error to detect pair isotopologues
massDifferenceIsotopes	mass difference to pair isotopologues. (Default = ΔC = ¹³C - ¹²C = 1.003354835336), ΔS = ³⁴S - ³²S = 1.9957958356, or any numerical value.

Value

A matrix consists of 5 columns. The column contents are the m/z of ¹²C isotopologues, intensity of ¹²C isotopologues, scan number (t), m/z of ¹³C isotopologues, and intensity of ¹³C isotopologues, respectively.

IPA_MSdeconvoluter *MS deconvoluter*

Description

This function deconvolutes mass spectrometry files into a list of mass spectrals and a vector of retention times.

Usage

```
IPA_MSdeconvoluter(HRMS_path, MSfile, MS_level = 1)
```

Arguments

HRMS_path	address of the mass spectrometry file
MSfile	mass spectrometry file.
MS_level	MS level to extract information.

Value

spectraList	a list of mass spectra.
RetentionTime	a vector of retention times for scan numbers.
MS_polarity	mass spectromtery ionization mode (+/-)

IPA_PeakAlignment *IPA peak alignment*

Description

This function produce an aligned peak table from individual peaklists.

Usage

```
IPA_PeakAlignment(PARAM)
```

Arguments

PARAM	is a data frame from IPA_xlsxAnalyzer function.

Value

This function saves individual .CSV and .Rdata files for the aligned peak tables for peak height, area, and R13C properties in the "peak_alignment" folder.

IPA_PeakAnalyzer *IPA Peak Analyzer*

Description

This function performs the IPA peak detection module.

Usage

```
IPA_PeakAnalyzer(PARAM)
```

Arguments

PARAM	is a data frame from IPA_xlsxAnalyzer function.

Value

This function saves individual peaklist files in .CSV and .Rdata formats for HRMS files in the "peaklists" folder.

IPA_PeaklistAnnotation *IPA Peaklist Annotation*

Description

This function performs sample-centric peak annotation.

Usage

```
IPA_PeaklistAnnotation(PARAM)
```

Arguments

PARAM	a data frame from IPA_xlsxAnalyzer function.

Value

This function saves individual .CSV files for peak height, area, and R13C properties in the "sample_centeric_annotation" folder.

IPA_TargetedAnalysis *IPA Targeted Analysis*

Description

This function plots extracted ion chromatogram (EIC) figures in the targeted mode.

Usage

```
IPA_TargetedAnalysis(spreadsheet, mzCandidate, rtCandidate, exportEIC = TRUE,
exportTable = FALSE)
```

Arguments

spreadsheet	a spreadsheet containing the parameters.
mzCandidate	a vector of candidate m/z values.
rtCandidate	a vector of candidate RT values.
exportEIC	TRUE by default. To plot and save EICs.
exportTable	FALSE by default. To return the whole peaklists for the m/z and RT vectors, select TRUE.

Value

This function saves extracted ion chromatograms in .png format in the "EICs" folder when "exportEIC = TRUE", and saves a table of peak properties when "exportTable = TRUE".

Examples

```
s_path <- system.file("extdata", package = "IDSL.IPA")
SSh1 <- paste0(s_path,"/IPA_parameters.xlsx")
spreadsheet <- readxl::read_xlsx(SSh1, sheet = 'IPA_targeted')
temp_wd <- tempdir()
temp_wd_zip <- paste0(temp_wd,"/idsl_ipa_test_files.zip")
download.file(paste0("https://github.com/idslme/IDSL.IPA/blob/main/",
"IPA_educational_files/idsl_ipa_test_files.zip?raw=true"),
destfile = temp_wd_zip)
unzip(temp_wd_zip, exdir = temp_wd)
spreadsheet[2, 4] <- temp_wd
spreadsheet[5, 4] <- temp_wd
mzCandidate <- c(53.01853, 61.00759)
rtCandidate <- c(0.951, 0.961)
IDSL.IPA::IPA_TargetedAnalysis(spreadsheet, mzCandidate, rtCandidate)
```

IPA_Workflow *IPA Workflow*

Description

This function executes the IPA workflow in order.

Usage

```
IPA_Workflow(spreadsheet)
```

Arguments

spreadsheet	IPA spreadsheet

Value

This function organizes the IPA file processing for a better performance using the template spreadsheet.

See Also

https://ipa.idsl.me/home

Examples

```
library(IDSL.IPA)
s_path <- system.file("extdata", package = "IDSL.IPA")
SSh1 <- paste0(s_path,"/IPA_parameters.xlsx")
temp_wd <- tempdir()
```

```
temp_wd_zip <- paste0(temp_wd,"/idsl_ipa_test_files.zip")
spreadsheet <- readxl::read_xlsx(SSh1)
download.file(paste0("https://github.com/idslme/IDSL.IPA/blob/main/",
"IPA_educational_files/idsl_ipa_test_files.zip?raw=true"),
destfile = temp_wd_zip)
unzip(temp_wd_zip, exdir = temp_wd)
spreadsheet[7, 4] <- temp_wd
spreadsheet[40, 4] <- s_path
spreadsheet[10, 4] <- temp_wd
IPA_Workflow(spreadsheet)
```

`IPA_xlsxAnalyzer` *IPA xlsx Analyzer*

Description

This function processes the spreadsheet of the IPA parameters to ensure the parameter inputs are in agreement with the IPA requirements.

Usage

```
IPA_xlsxAnalyzer(spreadsheet)
```

Arguments

`spreadsheet`	IPA spreadsheet

Value

This function returns the IPA parameters to feed the IPA_Workflow, IPA_CompoundsAnnotation, IPA_GapFiller, IPA_PeakAlignment, IPA_PeakAnalyzer, and IPA_PeaklistAnnotation functions.

Examples

```
s_path <- system.file("extdata", package = "IDSL.IPA")
SSh1 <- paste0(s_path, "/IPA_parameters.xlsx")
temp_wd <- tempdir()
temp_wd_zip <- paste0(temp_wd,"/idsl_ipa_test_files.zip")
spreadsheet <- readxl::read_xlsx(SSh1)
download.file(paste0("https://github.com/idslme/IDSL.IPA/blob/main/",
"IPA_educational_files/idsl_ipa_test_files.zip?raw=true"),
destfile = temp_wd_zip)
unzip(temp_wd_zip, exdir = temp_wd)
spreadsheet[7, 4] <- temp_wd
spreadsheet[40, 4] <- s_path # reference file location
spreadsheet[10, 4] <- temp_wd # output data location
PARAM <- IDSL.IPA::IPA_xlsxAnalyzer(spreadsheet)
```

islocalminimum *islocalminimum*

Description

This function returns indices of local minimum points on a curve.

Usage

```
islocalminimum(y)
```

Arguments

y	is a vector of y values.

Value

A vector in the same size of the vector 'y'. Local minimum arrays represented by -1.

Examples

```
data(chromatogram_builder)
int <- chromatogram_builder$SmoothedChromatogram
islocalminimum(int)
```

islocaloptimum *islocaloptimum*

Description

This function returns indices of local minimum and maximum points on a curve.

Usage

```
islocaloptimum(y)
```

Arguments

y	is a vector of y values.

Value

A vector in the same size of the vector 'y'. Local minimum and maximum arrays represented by -1 and +1, respectively.

Examples

```
data(chromatogram_builder)
int <- chromatogram_builder$SmoothedChromatogram
islocaloptimum(int)
```

`loadRdata` *loadRdata*

Description

This function loads .Rdata files into a variable.

Usage

```
loadRdata(fileName)
```

Arguments

`fileName`	is an .Rdata file.

Value

The called variable into the new assigned variable name.

`mzRTindexer` *m/z - RT Indexer*

Description

This function locate the closest pair of a reference (m/z - RT) pair in a 2-D grid of 'm/z' and 'RT' vectors.

Usage

```
mzRTindexer(MZvec, RTvec, MZref, RTref, MZtolerance, RTtolerance)
```

Arguments

`MZvec`	m/z vector
`RTvec`	RT vector
`MZref`	a reference m/z
`RTref`	a reference RT
`MZtolerance`	m/z tolerance
`RTtolerance`	RT tolerance

Value

index of closest pair to the reference (m/z - RT) pair

Note

This function returns NULL in case no match is detected.

`mz_clustering_xic` *mz clustering XIC*

Description

This function clusters related 12C m/z values.

Usage

```
mz_clustering_xic(spec_scan, mass_accuracy_xic, min_peak_height, min_nIsoPair)
```

Arguments

`spec_scan`	a matrix consists of 3 columns. The column contents are the m/z of 12C isotopologues, intensity of 12C isotopologues, and scan number (t).
`mass_accuracy_xic`	mass accuracy to detect related 12C m/z values.
`min_nIsoPair`	minimum number of nIsoPair for an individual peak.
`min_peak_height`	minimum peak height for an individual peak.

Value

This function returns an list on index numbers of EICs for the "spec_scan" variable.

`opendir` *opendir*

Description

This function opens the directory.

Usage

```
opendir(dir)
```

Arguments

`dir`	full address of the directory.

Value

This function opens its input directory for the user.

`peak_alignment` *Peak alignment*

Description

This function aligns peaks from multiple peaklists and produce a peak table to find common peaks among multiple samples.

Usage

```
peak_alignment(input_path_pl, file_names_pl, RT_pl, mz_error, rt_tol,
n_quantile, number_processing_threads = 1)
```

Arguments

`input_path_pl`	path to directory of peaklists.
`file_names_pl`	name of peaklists for peak table production.
`RT_pl`	a list of corrected or uncorrected retention times for each peaklist.
`mz_error`	mass error to detect common peaks.
`rt_tol`	retention time tolerance to detect common peaks.
`n_quantile`	number of total m/z quantiles to split the whole table for faster processing.
`number_processing_threads`	number of processing threads

Value

This function returns an aligned peak table with index numbers from individual peaklists for each peak.

`peak_area` *peak area*

Description

This function calculates area under the curve using a trapezoid method.

Usage

```
peak_area(x, y)
```

Arguments

x	is a vector of x values.
y	is a vector of y values.

Value

A number for the integrated peak area.

Examples

```
data("peak_spline")
rt <- peak_spline[, 1]
int <- peak_spline[, 2]
peak_area(rt, int)
```

---

peak_detection *peak detection*

Description

This function detects separated chromatographic peaks on the chromatogram.

Usage

```
peak_detection(int)
```

Arguments

int	a vector of intensities of the chromatogram.

Value

A matrix of 2 columns. Each row indicates peak boundary indices on the 'int' vector.

Examples

```
data(chromatogram_builder)
int <- chromatogram_builder$SmoothedChromatogram
peak_detection(int)
```

peak_property_table_correlation *Peak Property Table Correlation*

Description

Peak Property Table Correlation

Usage

```
peak_property_table_correlation(peakPropertyTable, RTtolerance = 0.05,
minFreqDetection = 1, method = "pearson", minThresholdCorrelation = 0,
number_processing_threads = 1)
```

Arguments

peakPropertyTable	peak property table such as 'peak_height', 'peak_area' and 'peak_R13C'
RTtolerance	retention time tolerance (min)
minFreqDetection	minimum frequency of detection for a (m/z-RT) peak across the peak property table
method	a character string indicating which correlation coefficient (or covariance) is to be computed. One of "pearson" (default), "kendall", or "spearman": can be abbreviated. (from 'cor' function of the 'stats' package)
minThresholdCorrelation	minimum threshold for the correlation method
number_processing_threads	number of processing threads

Value

A list of related peak IDs for each individual (m/z-RT) pair on the peak property table

peak_sharpness *Peak sharpness*

Description

This function measures sharpness of a chromatographic peak

Usage

```
peak_sharpness(int)
```

Arguments

int	a vector of intensities of the chromatographic peak.

Value

A number representing peak sharpness. The higher values indicate higher sharpness.

Examples

```
data("peak_spline")
int <- peak_spline[, 2]
peak_sharpness(int)
```

---

peak_spline *peak spline*

Description

illusterates a smoothe peak using cubic spline smoothing method

Usage

```
data("peak_spline")
```

Format

A data frame with 100 observations on the following 3 variables.

`rt_spline` a numeric vector

`int_spline` a numeric vector

`bl_approx` a numeric vector

Examples

```
data(peak_spline)
```

peak_width *peak width measuement*

Description

This function measures peak width at different peak heights.

Usage

```
peak_width(rt, int, gauge)
```

Arguments

rt	a vector of retention times of the chromatographic peak.
int	a vector of intensities of the chromatographic peak.
gauge	a height gauge to measure the peak width. This parameter should be between 0-1.

Value

A peak width at the guaged height.

Examples

```
data("peak_spline")
rt <- peak_spline[, 1]
int <- peak_spline[, 2] - peak_spline[, 3]
gauge <- 0.5
peak_width(rt, int, gauge)
```

peak_Xcol2 *Peak table producer*

Description

This function fills the peak table from individual peaklists.

Usage

```
peak_Xcol2(input_path_peaklist, file_names_peaklist, peak_Xcol)
```

Arguments

input_path_peaklist	address of the peaklists.
file_names_peaklist	a vector of the peaklist file names.
peak_Xcol	a matrix of index numbers in individual peaklists for each peak (m/z-RT).

Value

peak_height	peak table for height values
peak_area	peak table for area values
peak_R13C	peak table for R13C values

---

plot_mz_eic *plot_mz_eic*

Description

plot_mz_eic

Usage

```
plot_mz_eic(filelist, filelocation, mztarget, mzdelta,
number_processing_threads = 1, rtstart = 0, rtend = 0, plotTitle = "")
```

Arguments

filelist	filelist
filelocation	filelocation
mztarget	mztarget
mzdelta	mzdelta
number_processing_threads	number of processing threads
rtstart	rtstart
rtend	rtend
plotTitle	plotTitle

Value

plot_mz_eic

plot_simple_tic *plot_simple_tic*

Description

plot_simple_tic

Usage

```
plot_simple_tic(filelist, filelocation, number_processing_threads = 1,
plotTitle = "Total Ion Chromatogram")
```

Arguments

filelist	filelist
filelocation	filelocation
number_processing_threads	number of processing threads
plotTitle	plotTitle

Value

plot_simple_tic

primary_peak_analyzer *Primary peak analyzer*

Description

This function performs the first round of the chromatography analysis.

Usage

```
primary_peak_analyzer(spec_scan, index_xic, scan_tol,
spectraList, RetentionTime, mass_accuracy_xic,
smoothing_window, peak_resolving_power, min_nIsoPair,
min_peak_height, min_ratio_IsoPair, max_rpw, min_snr_baseline,
max_R13C_integrated_peak, max_percentage_missing_scans,
n_spline)
```

Arguments

spec_scan	a matrix consists of 5 columns. The column contents are the m/z of 12C isotopologues, intensity of 12C isotopologues, scan number (t), m/z of 13C isotopologues, and intensity of 13C isotopologues.
index_xic	a list of indices of candidate 12C m/z values from spec_scan matrix.
scan_tol	scan tolerance to extend the chromatogram for better calculations.
spectraList	a list of mass spectra in each chromatogram scan.
RetentionTime	a vector of retention times vs. corresponding scan numbers.
mass_accuracy_xic	a m/z value to perform chromatography analysis.
smoothing_window	number of scans for peak smoothing.
peak_resolving_power	a value to represent peak resolving power.
min_nIsoPair	minimum number of nIsoPair for an individual peak.
min_peak_height	minimum peak height for an individual peak.
min_ratio_IsoPair	minimum ratio of nIsoPair per number of available scans within an individual peak.
max_rpw	maximum allowed value of ratio of peak width at half-height to baseline (RPW) for an individual peak.
min_snr_baseline	minimum S/N baseline for an individual peak.
max_R13C_integrated_peak	maximum allowed value of average R13C for an individual peak.
max_percentage_missing_scans	maximum allowed value of percentage missing scans on the raw chromatogram for an individual peak.
n_spline	number of points for further smoothing using a cubic spline smoothing method.

Value

a data frame consisting of 24 columns representing chromatography and mass spectrometry parameters. Each row represents an individual separated chromatographic peak.

pseudomoments_symmetry

pseudomoments symmetry

Description

This function measures peak symmetry and skewness using the inflection points of the peak on both sides.

Usage

```
pseudomoments_symmetry(rt, int)
```

Arguments

rt	a vector of retention times for the chromatographic peak.
int	a vector of intensities corresponding to the vector of retention times for the chromatographic peak.

Value

PeakSymmetry	peak symmetry for the chromatographic peak.
Skewness	skewness for the chromatographic peak.

Examples

```
data("peak_spline")
rt <- peak_spline[, 1]
int <- peak_spline[, 2] - peak_spline[, 3]
pseudomoments_symmetry(rt, int)
```

recursive_mass_correction

recursive mass correction

Description

This function performs recursive mass correction.

Usage

```
recursive_mass_correction(peaklist, spec_scan, scan_tol,
spectraList, RetentionTime, mass_accuracy_xic, smoothing_window,
peak_resolving_power, min_nIsoPair, min_peak_height, min_ratio_IsoPair,
max_rpw, min_snr_baseline, max_R13C_integrated_peak,
max_percentage_missing_scans, n_spline)
```

Arguments

peaklist	an IPA peaklist from 'primary_peak_analyzer' function.
spec_scan	a matrix consists of 5 columns. The column contents are the m/z of 12C isotopologues, intensity of 12C isotopologues, scan number (t), m/z of 13C isotopologues, and intensity of 13C isotopologues.
scan_tol	a scan tolerance to extend the chromatogram for better calculations.
spectraList	a list of mass spectra in each chromatogram scan.
RetentionTime	a vector of retention times for corresponding scan numbers.
mass_accuracy_xic	an m/z value to perform chromatography analysis.
smoothing_window	a number of scans for peak smoothing.
peak_resolving_power	a value to represent peak resolving power.
min_nIsoPair	minimum number of nIsoPair for an individual peak.
min_peak_height	minimum peak height for an individual peak.
min_ratio_IsoPair	minimum ratio of nIsoPair per number of available scans within an individual peak.
max_rpw	maximum allowed value of ratio of peak width at half-height to baseline (RPW) for an individual peak.
min_snr_baseline	minimum S/N baseline for an individual peak.
max_R13C_integrated_peak	maximum allowed value of average R13C for an individual peak.
max_percentage_missing_scans	maximum allowed value of percentage missing scans on the raw chromatogram for an individual peak.
n_spline	number of points for further smoothing using a cubic spline smoothing method to calculate ancillary chromatographic parameters.

Value

a dataframe consisting of 24 columns representing chromatography and mass spectrometry parameters. Each row represents an individual separated chromatographic peak.

`reference_peaks_detector` *Reference peaks detector*

Description

This function detects recurring reference peaks (m/z-RT) for retention time correction.

Usage

```
reference_peaks_detector(input_path_peaklist, file_names_peaklist_ref,
min_frequency_ref_peaks, mz_error, rt_tol, n_quantile, number_processing_threads = 1)
```

Arguments

`input_path_peaklist`	path to directory of peaklists.
`file_names_peaklist_ref`	name of peaklists files to detect recurring reference peaks (m/z-RT).
`min_frequency_ref_peaks`	minimum frequency of the recurring reference peaks (m/z-RT) in the reference files.
`mz_error`	mass error to detect common peaks.
`rt_tol`	retention time tolerance to detect common peaks.
`n_quantile`	number of total m/z quantiles to split the whole table for faster processing.
`number_processing_threads`	number of processing threads

Value

`reference_mz_rt_peaks`	a matrix of two columns of m/z and RT of common peaks in the reference samples.
`listRefRT`	a list of corrected or uncorrected retention times for each peaklist.

`sample_rt_corrector` *sample retention time corrector*

Description

This function calculates corrected retention times for the peaklists.

Usage

```
sample_rt_corrector(reference_mz_rt_peaks, peaklist, mz_error,
rt_correction_method, reference_peak_tol = 1, polynomial_degree = 3)
```

Arguments

reference_mz_rt_peaks	a matrix of reference peaks for retention time correction.
peaklist	an IPA peaklist
mz_error	mass error to detect common reference peaks.
rt_correction_method	c('RetentionIndex','Polynomial')
reference_peak_tol	number of reference peaks for retention time correction using the 'RetentionIndex' method.
polynomial_degree	polynomial degree for retention time correction using the 'Polynomial' method.

Value

a list of corrected retention times for each peaklist.

---

segment *segment*

Description

This data illustrates an output matrix of chromatogram peak detection module from the "chromatogram_builder.rda" object.

Usage

```
data("segment")
```

Format

The format is: num [1:16, 1:2] 7 15 23 33 38 46 67 86 102 118 ...

Examples

```
data(segment)
```

snr_rms *SNR RMS*

Description

This function calculates signal-to-noise ratio using root mean square.

Usage

```
snr_rms(int, baseline, gauge)
```

Arguments

int	is the vector of intensities corresponding to the vector of retention times for the chromatographic peak.
baseline	is a vector of baseline of the chromatographic peak.
gauge	represents the gauge height of peak for gaussianity measurement.

Value

S/N value

Examples

```
data("peak_spline")
int <- peak_spline[, 2]
baseline <- peak_spline[, 3]
gauge <- 0.8
snr_rms(int, baseline, gauge)
```

snr_signal2baseline *SNR baseline*

Description

This function calculates S/N using local noise levels from baseline,

Usage

```
snr_signal2baseline(int, baseline)
```

Arguments

int	a vector of intensities corresponding to the vector of retention times for the chromatographic peak.
baseline	a vector of baseline of the chromatographic peak.

Value

S/N value

Examples

```
data("peak_spline")
int <- peak_spline[, 2]
baseline <- peak_spline[, 3]
snr_signal2baseline(int, baseline)
```

`snr_xcms` *SNR xcms*

Description

This function calculates S/N values using a method suggested in the xcms paper (Tautenhahn, 2008).

Usage

```
snr_xcms(int)
```

Arguments

`int`	a vector of intensities corresponding to the vector of retention times for the chromatographic peak.

Value

S/N value

References

Tautenhahn, R., Böttcher, C. and Neumann, S. (2008). Highly sensitive feature detection for high resolution LC/MS. *BMC bioinformatics*, 9(1), 1-16, doi: 10.1186/147121059504.

Examples

```
data(peak_spline)
int <- peak_spline[, 2]
snr_xcms(int)
```

spectraList_filtering *spectraList filtering*

Description

This function reduces the size of the spectraList value by removing m/z values with no correspondence to 12C/13C isotopologue pairs.

Usage

```
spectraList_filtering(spec_scan.xic, spectraList, rounding_digit = 1)
```

Arguments

spec_scan.xic	a matrix of any size, but the first column containing the m/z of 12C isotopologues are used.
spectraList	a list of mass spectra in each chromatogram scan.
rounding_digit	rounding digit to choose power of size reduction.

Value

a list of mass spectrals

usp_tailing_factor *USP tailing factor*

Description

This function calculates USP tailing factor at above 10 percent of the height.

Usage

```
usp_tailing_factor(rt, int)
```

Arguments

rt	a vector of retention times for the chromatographic peak.
int	a vector of intensities corresponding to the vector of retention times for the chromatographic peak.

Value

USP tailing factor for the chromatographic peak.

Examples

```
data(peak_spline)
rt <- peak_spline[, 1]
int <- peak_spline[, 2] - peak_spline[, 3]
usp_tailing_factor(rt, int)
```

`XIC` *XIC*

Description

XIC

Usage

```
XIC(spectraList.xic, scan_number_start = 1, mz_target, mass_accuracy_xic)
```

Arguments

`spectraList.xic`	a list of mass spectra in each chromatogram scan.
`scan_number_start`	the first scan number.
`mz_target`	an m/z value to perform XIC analysis.
`mass_accuracy_xic`	a mass error to perform XIC analysis.

Value

A matrix of three columns representing scan number, m/z, and intensity.

`xlsxAnalyzer_EIC` *xlsxAnalyzer EIC*

Description

This function processes the spreadsheet of the IPA parameters to ensure the parameter inputs are in agreement with the IPA_EIC requirements.

Usage

```
xlsxAnalyzer_EIC(spreadsheet)
```

Arguments

`spreadsheet`	contains the IPA parameters.

Value

This function returns the IPA parameters to feed the IPA_TargetedAnalysis function.

Index