Package 'adductomicsR'

September 23, 2025

Type Package

Title Processing of adductomic mass spectral datasets

Version 1.25.0

Date 2019-04-12

Author Josie Hayes <jlhayes1982@gmail.com>

Maintainer Josie Hayes <jlhayes1982@gmail.com>

Description
Processes MS2 data to identify potentially adducted peptides from spectra that has been corrected for mass drift and retention time drift and quantifies MS1 level mass spectral peaks.

Depends R (>= 3.6), adductData, ExperimentHub, AnnotationHub

Imports parallel (>= 3.3.2), data.table (>= 1.10.4), OrgMassSpecR (>= 0.4.6), foreach (>= 1.4.3), mzR (>= 2.14.0), ade4 (>= 1.7.6), rvest (>= 0.3.2), pastecs (>= 1.3.18), reshape2 (>= 1.4.2), pracma (>= 2.0.4), DT (>= 0.2), fpc (>= 2.1.10), doSNOW (>= 1.0.14), fastcluster (>= 1.1.22), RcppEigen (>= 0.3.3.3.0), bootstrap (>= 2017.2), smoother (>= 1.1), dplyr (>= 0.7.5), zoo (>= 1.8), stats (>= 3.5.0), utils (>= 3.5.0), graphics (>= 3.5.0), grDevices (>= 3.5.0), methods (>= 3.5.0), datasets (>= 3.5.0)

License Artistic-2.0

biocViews MassSpectrometry,Metabolomics,Software,ThirdPartyClient,DataImport, GUI

RoxygenNote 6.1.0

Suggests knitr (>= 1.15.1), rmarkdown (>= 1.5), Rdisop (>= 1.34.0), testthat

VignetteBuilder knitr

Collate '00AdductSpec-class.R' '00AdductQuantif-class.R' 'IsotopicDistributionMod.R' 'adductQuant.R' 'adductSpecGen.R' 'digestMod.R' 'dotProdMatrix.R' 'dotProdSpectra.R' 'dynamicNoiseFilter.R' 'filterAdductTable.R' 'findPeaks.R' 'generateTargTable.R' 'loessWrapperMod.R' 'ms2Group.R' 'nAdjPeaks.R' 'outputPeakTable.R' 'peakIdQuant_newMethod.R' 'peakIntegrate.R' 'peakListId.R' 'peakRangeSum.R' 'probPeaks.R' 'retentionCorr.R' 'rtDevModelSave.R' 'rtDevModelling.R' 'signalGrouping.R' 'specSimPepId.R' 'spectraCreate.R' 'truePeakTrough.R'

BugReports https://github.com/JosieLHayes/adductomicsR/issues

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

git_branch devel

git_last_commit 91b17a9

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-09-23

Contents

adductQuant . . . 3
AdductQuantif-class . . . 4
AdductSpec-class . . . 5
adductSpecGen . . . 6
digestMod . . . 7
dotProdMatrix . . . 8
dotProdSpectra . . . 9
dynamicNoiseFilter . . . 9
filterAdductTable . . . 10
findPeaks . . . 11
generateTargTable . . . 12
IsotopicDistributionMod . . . 12
loessWrapperMod . . . 13
ms2Group . . . 13
nAdjPeaks . . . 14
outputPeakTable . . . 15
peakIdQuant_newMethod . . . 15
peakIntegrate . . . 16
peakListId . . . 17
peakRangeSum . . . 18
probPeaks . . . 19
retentionCorr . . . 20
rtDevModelling . . . 20
rtDevModelSave . . . 22
signalGrouping . . . 23
specSimPepId . . . 23
spectraCreate . . . 25
truePeakTrough . . . 26

Index **27**

adductQuant *Adduct quantification for adductomicsR*

Description

reads mzXML files from a directory, corrects RT according to RT correction model and quantifies peaks.

Usage

```
adductQuant(nCores = NULL, targTable = NULL,
intStdRtDrift = NULL, rtDevModels = NULL,
filePaths = NULL, quantObject = NULL, indivAdduct = NULL, maxPpm = 4,
minSimScore = 0.8, spikeScans = 2, minPeakHeight = 100, maxRtDrift = 20,
maxRtWindow = 120, isoWindow = 80,
hkPeptide = "LVNEVTEFAK", gaussAlpha = 16)
```

Arguments

nCores	number of cores to use for analysis. If NULL then 1 core will be used.
targTable	is the fullpath to the target table. See inst/extdata/examplePeptideTargetTable.csv for an example.
intStdRtDrift	the maximum drift for the internal standard in seconds. Default = NULL and therefore no RT correction is applied to the internal standard.
rtDevModels	is the full path to the rtDevModels.RData file from rtDevModels(). default is NULL and therefore has no RT correction.
filePaths	required list of mzXML files for analysis. If all files are in the same directory these can be accessed using list.files('J:\parentdirectory\directoryContainingfiles', pattern='.mzXML', all.files=FALSE, full.names=TRUE).
quantObject	character string for filepath to an AdductQuantif object to be integrated.
indivAdduct	numeric vector of AdductQuantif targets to re-integrate
maxPpm	numeric for the maximum parts per million to be used.
minSimScore	a numeric between 0
spikeScans	a numeric for the number of scans that a spike must be seen in for it to be integrated. Default is 2.
minPeakHeight	numeric to determine the minimum height for a peak to be integrated. Default is set low at 100.
maxRtDrift	numeric for the maximum retention time drift to be considered. Default is 20.
maxRtWindow	numeric in seconds for the retention time window (total window will be 2 times this value)
isoWindow	numeric for the pepide isotope window in seconds, default is 80
hkPeptide	is capitalized string for the housekeeping peptide. The default is 'LVNEVTEFAK' from human serum albumin.
gaussAlpha	numeric for the gaussian smoothing parameter to smooth the peaks. Default is 16. Output is an adductQuantf object saved to the working directory

Value

adductQuant object

Examples

```
## Not run:
eh = ExperimentHub();
temp = query(eh, 'adductData');
adductQuant(nCores=2, targTable=paste0(system.file("extdata",
package = "adductomicsR"),'/exampletargTable2.csv'), intStdRtDrift=30,
rtDevModels=paste0(hubCache(temp),"/rtDevModels.RData"),
filePaths=list.files(hubCache(temp),pattern=".mzXML", all.files=FALSE,
full.names=TRUE)[1],quantObject=NULL,
indivAdduct=NULL,maxPpm=5,minSimScore=0.8,spikeScans=1,
minPeakHeight=100,maxRtDrift=20,maxRtWindow=240,isoWindow=80,
hkPeptide='LVNEVTEFAK', gaussAlpha=16)

## End(Not run)
```

`AdductQuantif-class` *AdductQuantif class The AdductQuantif class contains a peak integral matrix, peak ranges and region of integration, the isotopic distribution identified for each integrated peak and the target table of peaks integrated.*

Description

AdductQuantif class The AdductQuantif class contains a peak integral matrix, peak ranges and region of integration, the isotopic distribution identified for each integrated peak and the target table of peaks integrated.

Usage

```
x
```

Format

An object of class `NULL` of length 0.

Value

peak integral matrix, peak ranges and region of integration, the isotopic distribution identified for each integrated peak and the target table of peaks integrated and their corresponding MS1 scan isotopic patterns

Slots

`peakQuantTable` a matrix containing the peak integration results and consisting of a row for each peak identified in each sample (e.g 200 samples and 50 targets 200 * 50 = 10,000 rows)

`peakIdData` list of peak IDs

`predIsoDist` list of predicted Iso distances

`targTable` dataframe target table

`file.paths` character path to file

`Parameters` dataframe of specified parameters

Methods

c `signature(object = "AdductQuantif")`: Concatenates the spectra information.

file.paths `signature(object = "AdductQuantif")`: Accesses the file paths.

peakQuantTable `signature(object = "AdductQuantif")`: Accesses the peak quantification data as a table.

peakIdData `signature(object = "AdductQuantif")`: Accesses the ID data for the peaks.

predIsoDist `signature(object = "AdductQuantif")`: Accesses the predicted isotopic distribution.

targTable `signature(object = "AdductQuantif")`: Accesses the user provided target table.

Author(s)

JL Hayes <jlhayes1982@gmail.com>

AdductSpec-class *AdductSpec class*

Description

The AdductSpec class contains dynamic noise filtered composite MS/MS spectra and their corresponding MS1 scan isotopic patterns. Produced by adductSpecGen() from mzXML files.

Usage

```
x
```

Format

An object of class `NULL` of length 0.

Value

dynamic noise filtered composite MS/MS spectra and their corresponding MS1 scan isotopic patterns

Slots

`adductMS2spec` list of adduct MS2 spectras

`groupMS2spec` list of group MS2 spectras

`metaData` dataframe of metadata from mzXML

`aaResSeqs` matrix of amino acid sequences

`specPepMatches` list of spectra peptide matches

`specPepCompSpec` list of comp spectra peptide matches

`sumAdductType` dataframe of adduct types

`Peptides` dataframe of peptides under study

`rtDevModels` list of rtDevModels

`targetTable` dataframe target table

`file.paths` character of file path

`Parameters` dataframe of parameters

Methods

c `signature(object = "AdductSpec")`: Concatenates the spectra information.

Specfile.paths `signature(object = "AdductSpec")`: Accesses the file paths.

adductMS2spec `signature(object = "AdductSpec")`: Accesses the adduct MS2 spectral information.

metaData `signature(object = "AdductSpec")`: Accesses the scan metadata.

Parameters `signature(object = "AdductSpec")`: Accesses the user parameters.

groupMS2spec `signature(object = "AdductSpec")`: Accesses the spectral information for the grouped MS2 spectra.

rtDevModels `signature(object = "AdductSpec")`: Accesses the retention time deviation models.

sumAdductType `signature(object = "AdductSpec")`: Accesses the total adduct types.

Peptides `signature(object = "AdductSpec")`: Accesses the peptide information.

specPepMatches `signature(object = "AdductSpec")`: Accesses the peptide matches in the spectra.

Author(s)

JL Hayes <jlhayes1982@gmail.com>

`adductSpecGen` *Constructor of AdductSpec object deconvolute spectra MS2 and MS1 levels*

Description

reads mzXML files from a directory extracts metadata info, groups ion signals with `signalGrouping`, filters noise dynamically `dynamicNoiseFilter` and identifies precursor ion charge state, by isotopic pattern.

Usage

```
adductSpecGen(mzXmlDir=NULL, runOrder=NULL, nCores=NULL,
intStdMass=834.77692,intStdPeakList=c(290.21, 403.30, 516.38,
587.42,849.40, 884.92, 958.46, 993.97,1050.52, 1107.06, 1209.73,
1337.79,1465.85),TICfilter=10000, DNF=2, minInt=300,
minPeaks=5,intStd_MaxMedRtDrift=360, intStd_MaxPpmDev=200,minSpecEx=40,
outputPlotDir=NULL)
```

Arguments

mzXmlDir	character a full path to a directory containing either .mzXML or .mzML data
runOrder	character a full path to a csv file specifying the runorder for each of the files the first column must contain the precise file name and the second column an integer representing the precise run order.
nCores	numeric the number of cores to use for parallel computation. The default is to 1 core
intStdMass	numeric vector of the mass of the internal standard. Default is the mass of
intStdPeakList	numeric vector of masses for the internal standard peaks
TICfilter	numeric minimimum total ion current of an MS/MS scan. Any MS/MS scan below this value will be filtered out (default=0).
DNF	dynamic noise filter minimum signal to noise threshold (default = 2), calculated as the ratio between the linear model predicted intensity value and the actual intensity.
minInt	numeric minimum intensity value
minPeaks	minimum number of signal peaks following dynamic noise filtration (default = 5).
intStd_MaxMedRtDrift	numeric the maximum retention time drift window (in seconds) to identify internal standard MS/MS spectrum scans (default = 600).
intStd_MaxPpmDev	numeric the maximum mass accuracy window (in ppm). to identify internal standard MS/MS spectrum scans (default = 200 ppm).
minSpecEx	numeric the minimum percentage of the total ion current explained by the internal standard fragments (default = 40). Sometime spectra are not identified due to this cutoff being set too high. If unexpected datapoints have been interpolated then reduce this value.
outputPlotDir	character string for the output directory for plots, default is working directory.

Value

AdductSpec object

`digestMod` *modified* `Digest` *function (from OrgMassSpecR package)*

Description

allows maxCharge to be set to calculate precursor m/z

Usage

```
digestMod(sequence, enzyme = "trypsin", missed = 0,
maxCharge = 8,IAA = TRUE, N15 = FALSE, custom = list())
```

Arguments

sequence	a character string representing the amino acid sequence.
enzyme	is the enzyme to perform in silico digestion with
missed	the maximum number of missed cleavages. Must be an integer of 0 (default) or greater. An error will result if the specified number of missed cleavages is greater than the maximum possible number of missed cleavages.
maxCharge	numeric max charge charge for predicted precursor m/z
IAA	logical. TRUE specifies iodoacetylated cysteine and FALSE specifies unmodified cysteine. Used only in determining the elemental formula, not the three letter codes.
N15	logical indicating if the nitrogen-15 isotope should be used in place of the default nitrogen-14 isotope. calculation
custom	list of custom masses

Details

see Digest for details of further function arguments.

Value

dataframe

Examples

```
digestMod('MKWVTFISLLFLFSSAYSRGVFRRDAHKSEVAHRFKDLGEENFKALVLIA',
enzyme = "trypsin", missed = 0, maxCharge = 8,IAA = TRUE, N15 = FALSE,
custom = list())
```

---

dotProdMatrix *dot product matrix calculation*

Description

dot product matrix calculation

Usage

```
dotProdMatrix(allSpectra = NULL, spectraNames = NULL, binSizeMS2 =
NULL)
```

Arguments

allSpectra	a numeric matrix consisting of two columns 1. mass and 2. intensity
spectraNames	character names of individual spectra to compare must equal number of rows of allSpectra
binSizeMS2	numeric the MS2 bin size to bin MS2 data prior to dot product calculation (default = 0.1 Da).

Value

a matrix of equal dimension corresponding to the number of unique spectrum names

dotProdSpectra *dot product calculation*

Description

hierarchical clustering (complete method see hclust). Dissimilarity metric based on 1-dot product spectral similarity. Retention time and mass groups are therefore further subdivided based on spectral similarity. If outlying mass spectra have been erroneously grouped then these will be reclassified.

Usage

```
dotProdSpectra(adductSpectra = NULL, nCores = NULL,
minDotProdSpec = 0.8, maxGroups = 10)
```

Arguments

adductSpectra	AdductSpec object
nCores	numeric the number of cores to use for parallel computation. The default is to use 1 core.
minDotProdSpec	numeric minimum dot product score
maxGroups	numeric maximum number of groups to include from the dendrogram.

Value

adductSpectra AdductSpec object

dynamicNoiseFilter *Dynamic Noise filtration*

Description

Dynamic Noise filtration

Usage

```
dynamicNoiseFilter(spectrum.df = NULL, DNF = 2, minPeaks = 5,
minInt = 100)
```

Arguments

spectrum.df	a dataframe or matrix with two columns: 1. Mass/ Mass-to-charge ratio 2. Intensity
DNF	dynamic noise filter minimum signal to noise threshold (default = 2), calculated as the ratio between the linear model predicted intensity value and the actual intensity.
minPeaks	minimum number of signal peaks following dynamic noise filtration (default = 5).
minInt	integer minimum dynamic noise filter

Details

Dynamic noise filter adapted from the method described in Xu H. and Frietas M. 'A Dynamic Noise Level Algorithm for Spectral Screening of Peptide MS/MS Spectra' 2010 BMC Bioinformatics. The function iteratively calculates linear models starting from the median value of the lower half of all intensities in the spectrum.df. The linear model is used to predict the next peak intensity and ratio is calculated between the predicted and actual intensity value. Assuming that all preceeding intensities included in the linear model are noise, the signal to noise ratio between the predicted and actual values should exceed the minimum signal to noise ratio (default DNF = 2). The function continues until either the DNF value minimum has been exceeded and is also below the maxPeaks or maximum number of peaks value. As the function must necessarily calculate potentially hundreds of linear models the RcppEigen package is used to increase the speed of computation.

Value

a list containing 3 objects: 1. Above.noise The dynamic noise filtered matrix/ dataframe 2. metaData a dataframe with the following column names: 1. Noise.level the noise level determined by the dynamic noise filter function. 2. IntCompSpec Total intensity composite spectrum. 3. TotalIntSNR Sparse ion signal to noise ratio (mean intensity/ stdev intensity) 4. nPeaks number of peaks in composite spectrum 3. aboveMinPeaks Logical are the number of signals above the minimum level

---

filterAdductTable *filter samples with low QC and features with large missing values Removes adducts that have not been integrated with many missing values and provides QC on samples*

Description

filter samples with low QC and features with large missing values Removes adducts that have not been integrated with many missing values and provides QC on samples

Usage

```
filterAdductTable(adductTable = NULL, percMissing = 51, HKPmass =
"575.3", quantPeptideMass = "811.7", remHKPzero = FALSE, remQuantPepzero
=FALSE, remHKPlow = FALSE, outputDir = NULL)
```

Arguments

adductTable	character a full path to the peaktable with number of rows equal to the number of adducts from outputPeakTable() which starts with adductQuantif_peakList_
percMissing	numeric percentage threshold to remove adducts with missing values. Default is 51. It is recommended to use just over the number of samples in the smallest group of your study. 51 is used as default for a 50:50 case control study
HKPmass	numeric mass for the housekeeping peptide. Must be the same asthat in the adduct table. max 2 decimal places. default= 575.3 for the LVNEVTEFAK peptide
quantPeptideMass	numeric mass for the peptide for which adducts are being quantified, Default is 811.7 for the ALVLIAFAQYLQQCPFEDHVK peptide

remHKPzero	logical if TRUE removes all samples where the housekeeping peptide is 0. default= FALSE
remQuantPepzero	logical if TRUE removes all samples where the peptide under quantification is 0. default= FALSE
remHKPlow	logical if TRUE removes all samples where the housekeeping peptide has an area less than 100000. default= TRUE.This is recommended because this peak should be large. If the HKP has been mis-identified quantification of all adducts will be affected.
outputDir	character path to results directory output is a csv file with only adducts and samples that passed filter. Remaining adducts can be quantified manually however it is recommended to rescale the quantification results and include the quantification method as a covariate in downstream analysis.

Value

csv file

Examples

```
filterAdductTable(adductTable=paste0(system.file("extdata",
package="adductomicsR"),'/example_adductQuantif_peakList.csv'), percMissing
=51,HKPmass = "575.3", quantPeptideMass = "811.7",
remHKPzero=FALSE,remQuantPepzero = FALSE, remHKPlow = FALSE, outputDir =
NULL)
```

---

findPeaks *identify peaks*

Description

identifies peaks in a vector of intensities.

Usage

```
findPeaks(x, m = 3)
```

Arguments

x	numeric vector of intensities.
m	number of peaks to identify

Value

string of peaks

Examples

```
findPeaks(c(200, 300,200, 200, 200 , 300, 200), m = 3)
```

`generateTargTable` *Make a target table for adductomicsR quantificaton using specSimPep results*

Description

Make a target table for adductomicsR quantificaton using specSimPep results

Usage

```
generateTargTable(allresultsFile = NULL, csvDir = NULL)
```

Arguments

`allresultsFile`	character a full path to the allResults file generated by specSimPepId
`csvDir`	character a full path to a directory to save the csv file to output is a csv file called targTable.csv which can be used in the adductQuant function

Value

cvs file

Examples

```
generateTargTable(paste0(system.file("extdata",package="adductomicsR"),
'/allResults_ALVLIAFAQYLQQCPFEDHVK_example.csv'),csvDir=getwd())
```

`IsotopicDistributionMod` *modified function from package OrgMassSpecR*

Description

modified function from package OrgMassSpecR

Usage

```
IsotopicDistributionMod(formula = list(), charge = 1)
```

Arguments

`formula`	list of character strings representing elemental formula
`charge`	numeric for charge of the element

Value

dataframe of a spectrum

Examples

```
IsotopicDistributionMod(formula=list("CH3CH2OH","H2O"),charge = 1)
```

loessWrapperMod *wrapper script for loess modeling*

Description

adapted from bisoreg package

Usage

```
loessWrapperMod(x, y, span.vals = seq(0.25, 1, by = 0.05),
folds = 5)
```

Arguments

x	predictor values
y	response values
span.vals	values of the tuning parameter to evaluate using cross validation
folds	number of 'folds' for the cross-validation procedure

Value

LOESS model

Examples

```
loessWrapperMod (rnorm(200), rnorm(200), span.vals =
seq(0.25, 1, by = 0.05),folds = 5)
```

ms2Group *group MS/MS precursor masses*

Description

hierarchically cluster ms/ms precursor scans within and across samples, according to a m/z and retention time error.

Usage

```
ms2Group(adductSpectra = NULL, nCores = NULL,
maxRtDrift = NULL,
ms1mzError = 0.1, ms2mzError = 1, dotProdClust = TRUE, minDotProd = 0.8,
fclustMethod = "median", disMetric = "euclidean", compSpecGen = TRUE,
adjPrecursorMZ = TRUE)
```

Arguments

adductSpectra	AdductSpec object
nCores	numeric the number of cores to use for parallel computation. The default is to use 1 core.
maxRtDrift	numeric for the maximum rentention time drift to be considered. Default is 20.
ms1mzError	numeric maximum MS1 mass:charge error
ms2mzError	numeric maximum MS2 mass:charge error
dotProdClust	logical remove previous dot prod clustering results
minDotProd	numeric. Minimum mean dot product spectral similarity score to keep a spectrum within an MS/MS group (default = 0.8).
fclustMethod	method to use for the fclust function
disMetric	metric to use for distance in clustering
compSpecGen	logical for whether composite spectra generation is necessary
adjPrecursorMZ	logical for precursor mass:charge adjustment

Value

a list identical to adductSpectra containing an additional list element:

---

nAdjPeaks *remove lower intensity adjacent peaks*

Description

remove lower intensity adjacent peaks

Usage

```
nAdjPeaks(peaksTmp = NULL, troughsTmp = NULL, peakRangeTmp = NULL)
```

Arguments

peaksTmp	character vector with indices of detected peaks from findPeaks
troughsTmp	character vector with indices of detected troughs from findPeaks
peakRangeTmp	matrix of the peak range data with at least 3 columns (1. mass-to-charge, 2. intensity, 3. retention time)

Value

peaksTmp but with lower intensity adjacent peaks between the same troughs removed

outputPeakTable *output peak table from AdductQuantif object*

Description

output peak table from AdductQuantif object

Usage

```
outputPeakTable(object = NULL, outputDir = NULL)
```

Arguments

object	a 'AdductQuantif' class object
outputDir	character full path to a directory to output the peak to default is the current working directory

Value

a peaktable with number of rows equal to the number of adducts quantified and 14 peak group information columns plus a number of columns equal to the number of samples quantified. The peak table is saved as a csv file in the output directory named: adductQuantif_peakList_'todays date'.csv. The peak table is also returned to the R session and can be assigned to an object.

Examples

```
eh = ExperimentHub();
Temp = query(eh, c("adductData", "adductQuant", "Rda"))[[1]];
outputPeakTable(object=Temp)
```

peakIdQuant_newMethod *Adduct Peak quant*

Description

peak must be at least 50 percent resolved from overlapping peaks. i.e. the peaks trough must be at least 50 percent of the peak apex intensity for the peak to be considered sufficiently resolved.

Usage

```
peakIdQuant_newMethod(mzTmp = NULL, rtTmp = NULL,
peakRangeRtSub = NULL, rtDevModel = NULL, isoPat = NULL,
isoPatPred = NULL, minSimScore = 0.96, maxPpm = 4,
gaussAlpha = 16, spikeScans = 2, minPeakHeight = 5000,
maxRtDrift = 20, showPlots = FALSE,
isoWindow = 10, maxGapMs1Scan = 5, intMaxPeak = FALSE)
```

Arguments

mzTmp	expected mass to charge of target
rtTmp	expected retention time (in minutes) of target
peakRangeRtSub	matrix MS1 scans covering entire chromatographic range within which to identify peaks of interest. Contains the following three columns column 1 = mass, column 2 = intensity, column 3 = retention time, column 4 = scan number.
rtDevModel	loess retention time deviation model for the file.
isoPat	named numeric containing the expected mass differences between isotopes for the peptide of interest.
isoPatPred	matrix output from the IsotopicDistribution function with additional 'id' column.
minSimScore	numeric minimum dot product score for consideration (must be between 0-1, default = 0.96).
maxPpm	numeric ppm value for EIC extraction and integration.
gaussAlpha	numeric alpha value for smth.gaussian of smoother package. If supplied gaussian smoothing will be performed (suggested value = 16).
spikeScans	numeric number of scans that constitute a spike.
minPeakHeight	numeric minimum peak height, default 5000
maxRtDrift	numeric maximum retention time drift, default 20 secs
showPlots	boolean for whether plots should be produced
isoWindow	numeric isowindow size, default 10
maxGapMs1Scan	maximum MS1 scan gap, default 5
intMaxPeak	boolean integrate maximum peak

Value

list

---

peakIntegrate *integrate a peak from a peak table with peak start and peak end retention times*

Description

integrate a peak from a peak table with peak start and peak end retention times

Usage

```
peakIntegrate(peakTable = NULL, peakStart = NULL,
peakEnd = NULL, expMass = NULL,
expRt = NULL)
```

Arguments

peakTable	a table of at least 5 columns: 1. mass-to-charge. 2. intensity 3. adjusted retention time 4. raw retention time 5. scan numbers
peakStart	retention time for peak start (in seconds).
peakEnd	retention time for peak end (in seconds).
expMass	expected mass-to-charge of target.
expRt	expected retention time of target (in seconds).

Value

list with peak and peak table

---

peakListId *peak list Identification*

Description

peak list Identification

Usage

```
peakListId(adductSpectra = NULL, peakList = c(290.21, 403.3,
516.38, 587.42, 849.4, 884.92, 958.46, 993.97, 1050.52, 1107.06,
1209.73, 1337.79,
1465.85), exPeakMass = 834.7769, frag.delta = 1, minPeaksId = 7,
minSpecEx = 50, maxRtDrift = 360, maxPpmDev = 200, allScans = TRUE,
closestMassByFile = TRUE, outputPlotDir = NULL)
```

Arguments

adductSpectra	AdductSpec object param peakList numeric vector of peak masses param exPeakMass numeric internal standard peak mass
peakList	numeric vector of peak masses
exPeakMass	numeric mass of explained peak
frag.delta	integer delta mass accuracy difference.
minPeaksId	numeric minimum number of peaks IDed
minSpecEx	numeric the minimum percentage of the total ion current explained by the internal standard fragments (default = 40). Sometime spectra are not identified due to this cutoff being set too high. If unexpected datapoints have been interpolated then reduce this value.

maxRtDrift	numeric the maximum retention time drift (in seconds) to identify MS/MS spectrum scans (default = 360). param outputPlotDir character string of output directory (e.g. internal standard IAA-T3 peak list = peakList= c(290.21, 403.30, 516.38, 587.42, 849.40, 884.92, 958.46, 993.97, 1050.52, 1107.06, 1209.73, 1337.79, 1465.85))
maxPpmDev	numeric ppm deviation
allScans	boolean include all scans
closestMassByFile	boolean closest mass in files
outputPlotDir	character string for output plot directory

Value

dataframe peak list

---

peakRangeSum *raw eic signal intensity and mass summation and spike removal.*

Description

raw eic signal intensity and mass summation and spike removal.

Usage

```
peakRangeSum(peakRange = NULL, spikeScans = 2, rtDevModel = NULL,
gaussAlpha = NULL,
maxEmptyRt = 7)
```

Arguments

peakRange	matrix consisting of 5 columns: 1. mass-to-charge values 2. intensity 3. retention time (in seconds) 4. scan number
spikeScans	numeric number of scans <= a spike. Any peaks <= this value will be removed (default = 2).= FALSE
rtDevModel	loess model to correct retention times.
gaussAlpha	numeric alpha value for smth.gaussian of smoother package. If supplied gaussian smoothing will be performed (suggested value = 16).
maxEmptyRt	numeric maximum size of empty retention time beyond which missing values will be zero-filled

Value

matrix with masses and intensities summed by retention time and retention time correction based on the loess model supplied, the matrix has spikes removed (consecutive non-zero intensity values <= spikeScans in length), empty time segments are zero filled (> 3 seconds), optionally gaussian smoothed using the linksmth.gaussian function of the smoother package and is also subset based on the minimum and maximum retention time windows supplied (rtWin). The returned matrix consists of 5 columns:

1. average mass-to-charge values by unique retention time in supplied peakRange table
2. maximum intensity values by unique retention time in supplied peakRange table
3. loess model corrected retention times
4. original retention time values
5. scan number by unique retention time in supplied peakRange table

---

`probPeaks` *potentially problematic peak identification*

Description

potentially problematic peak identification

Usage

```
probPeaks(object = NULL, nTimesMad = 3,
metrics = c("nMadDotProdDistN",
"nMadSkewness", "nMadKurtosis", "nMadRtGroupDev",
"nMadPeakArea", "duplicates"))
```

Arguments

`object`	an 'AdductQuantif' class object
`nTimesMad`	numeric number of median absolute deviations to identify potential problem peaks.
`metrics`	character string column names of metrics with which to identify potential problem peaks or a list with individual nTimesMad arguments and with list element names corresponding to column names of metrics.
`...`	further arguments to `mad`

Value

'AdductQuantif' class object

retentionCorr *loess-based retention time deviation correction*

Description

loess-based retention time deviation correction

Usage

```
retentionCorr(adductSpectra = NULL,
smoothingSpan = NULL, nMissing = 1,
nExtra = 1, folds = 7, outputFileDir = NULL)
```

Arguments

adductSpectra	AdductSpec object
smoothingSpan	numeric. fixed smoothing span, argument to loess. If argument is not supplied then optimal smoothing span is calculated for each file seperately.
nMissing	numeric. maximum number of missing files for a MS/MS scan group to be utilized in the loess retention time deviation model. Roughly 15 percent missing values is a good starting point (e.g. nMissing=10 for 68 samples).
nExtra	numeric maximum number of extra scans above the total number of files for a MS/MS scan group to be utilized in the loess retention time deviation model. If a MS/MS scan group consists of many scans far in excess of the number of files then potentially MS/MS scans from large tailing peaks or isobars may be erroneously grouped together and used to adjust retention time incorrectly.
folds	numeric. number of cross validation steps to perform in identifying optimal smoothing span parameter (see: bisoreg package for more details)
outputFileDir	character full path to a directory to save the output images

Value

LOESS RT models as adductSpectra AdductSpec object

rtDevModelling *MS/MS spectrum grouping and retention time deviation modelling for adductomicsR*

Description

MS/MS spectrum grouping and retention time deviation modelling for adductomicsR

Usage

```
rtDevModelling(MS2Dir = NULL, runOrder = NULL,
nCores = NULL, TICfilter = 0,
intStdPeakList=c(290.21, 403.30, 516.38, 587.42,849.40, 884.92, 958.46,
993.97,1050.52, 1107.06, 1209.73, 1337.79,1465.85),
intStdMass = 834.77692, intStd_MaxMedRtDrift = 600, intStd_MaxPpmDev = 200,
minSpecEx = 40,
minDotProd = 0.8, percMissing = 15, percExtra = 100, smoothingSpan = 0.8,
saveRtDev = 1, outputPlotDir = NULL)
```

Arguments

MS2Dir	character a full path to a directory containing either .mzXML or .mzML data
runOrder	character a full path to a csv file specifying the runorder for each of the files the first column must contain the precise file name and the second column an integer representing the precise run order.
nCores	numeric the number of cores to use for parallel computation. The default is to 1 core.
TICfilter	numeric minimimum total ion current of an MS/MS scan. Any MS/MS scan below this value will be filtered out (default=0).
intStdPeakList	character a comma seperated list of expected fragment ions for the internal standard spectrum (no white space).
intStdMass	numeric expected mass-to-charge ratio of internal standard precursor (default = 834.77692).
intStd_MaxMedRtDrift	numeric the maximum retention time drift window (in seconds) to identify internal standard MS/MS spectrum scans (default = 600).
intStd_MaxPpmDev	numeric the maximum mass accuracy window (in ppm) to identify internal standard MS/MS spectrum scans (default = 200 ppm).
minSpecEx	numeric the minimum percentage of the total ion current explained by the internal standard fragments (default = 40). Sometimes spectra are not identified due to this cutoff being set too high. If unexpected datapoints have been interpolated then reduce this value.
minDotProd	numeric. Minimum mean dot product spectral similarity score to keep a spectrum within an MS/MS group (default = 0.8).
percMissing	numeric. percentage of missing files for a MS/MS scan group to be utilized in the loess retention time deviation model. Roughly 15 percent missing values (default = 15%) is a good starting point (e.g. nMissing=10 for 68 samples).
percExtra	numeric percentage of extra scans above the total number of files for a MS/MS scan group to be utilized in the loess retention time deviation model. If a MS/MS scan group consists of many scans far in excess of the number of files then potentially MS/MS scans from large tailing peaks or isobars may be erroneously grouped together and used to adjust retention time incorrectly (default = 100% i.e. the peak group can only have one scan per file, this value can be increased if two or more consecutive scans for example can be considered).
smoothingSpan	numeric. fixed smoothing span, argument to loess. If argument is not supplied then optimal smoothing span is calculated for each file seperately using 7-fold CV.

saveRtDev	integer (default = 1) should just the retention time deviation model be saved (TRUE = 1) or the AdductSpec class object (FALSE = 0) as .RData workspace files.
outputPlotDir	character (default = NULL) output directory for plots.

Value

LOESS RT models as adductSpectra AdductSpec object

Examples

```
eh = ExperimentHub();
temp = query(eh, 'adductData');
temp[['EH2061']]; #first mzXML file
file.rename(cache(temp["EH2061"]), file.path(hubCache(temp),
'data42_21221_2.mzXML'));
rtDevModelling(MS2Dir=hubCache(temp),nCores=2,runOrder=paste0(
system.file("extdata",package="adductomicsR"),
'/runOrder2.csv'), intStdPeakList=c(290.21, 403.30, 516.38,
587.42,849.40, 884.92, 958.46, 993.97,1050.52, 1107.06, 1209.73,
1337.79,1465.85))
```

---

rtDevModelSave *extract and save retention time deviation models from adductSpec class object*

Description

extract and save retention time deviation models from adductSpec class object

Usage

```
rtDevModelSave(object = NULL, outputDir = NULL)
```

Arguments

object	an 'adductSpec' class object or full path to a .RData file of the 'adductSpec' object
outputDir	character full path to a directory to save the .RData file (defaults to the current working directory if unsupplied).

Value

save a .RData file containing the rt deviation models and returns to the workspace.

`signalGrouping` *Signal grouping*

Description

Euclidean distances between m/z signals are hierarchically clustering using the average method and the composite spectrum groups determined by an absolute error cutoff

Usage

```
signalGrouping(spectrum.df = NULL, mzError = 0.8, minPeaks = 5)
```

Arguments

`spectrum.df`	a dataframe or matrix with two or more columns: 1. Mass/ Mass-to-charge ratio 2. Intensity
`mzError`	interpeak absolute m/z error for signal grouping (Default = 0.001)
`minPeaks`	numeric minimum number of peaks to integrate

Value

dataframe of m/z grouped signals, the m/z values of the input dataframe/ matrix peak groups are averaged and the signal intensities summed.

`specSimPepId` *spectral similarity based adducted peptide identification for adductomicsR*

Description

spectral similarity based adducted peptide identification for adductomicsR

Usage

```
specSimPepId(MS2Dir=NULL,nCores=NULL,
rtDevModels=NULL, topIons=100, topIntIt=5,minDotProd=0.8, precCh=3,
minSNR=3,minRt=20, maxRt=35, minIdScore=0.4,minFixed=3, minMz=750,
maxMz=1000,modelSpec=c('ALVLIAFAQYLQQCPFEDHVK','RHPYFYAPELLFFAK'),
groupMzabs=0.005, groupRtDev=0.5, possFormMzabs=0.01,
minMeanSpecSim=0.7,idPossForm=0, outputPlotDir= NULL)
```

Arguments

`MS2Dir`	character a full path to a directory containing either .mzXML or .mzML data
`nCores`	numeric the number of cores to use for parallel computation. The default is to use 1 core.
`rtDevModels`	a list object or a full path to an RData file containing the retention time deviation models for the dataset.

`topIons` numeric the number of most intense ions to consider for the basepeak to fragment mass difference calculation (default = 100). Larger values will slightly increase computation time, however when the modified/variable ions happen to be low abundance this value should be set high to ensure these fragment ions are considered.

`topIntIt` numeric the number of most intense peaks to calculate the peak to peak mass differences from (default = 5 i.e. the base peak and the next 4 most intense ions greater than 10 daltons in mass from one another will be considered the multiple iterations increase computation time but in the case that the peptide spectrum is contaminated/chimeric or the variable ions are of lower intensity this parameter should be increased).

`minDotProd` numeric minimum dot product similarity score (cosine) between the model spectra's variable ions and the corresponding intensities of the basepeak to fragment ion mass differences identified in the experimental spectrum scans (default = 0.8). Low values will greatly increase the potential for false positive peptide annotations.

`precCh` integer charge state of precursors (default = 3).

`minSNR` numeric the minimum signal to noise ratio for a fragment ion to be considered. The noise level for each fixed or variable ion is calculated by taking the median of the bottom half of ion intensities within the locality of the fragment ion. The locality is defined as within +/- 100 Daltons of the fragment ion.

`minRt` numeric the minimum retention time (in minutes) within which to identify peptide spectra (default=20).

`maxRt` numeric the maximum retention time (in minutes) within which to identify peptide spectra (default=45).

`minIdScore` numeric the minimum identification score this is an average score of all of the 7 scoring metrics (default=0.4).

`minFixed` numeric the minimum number of fixed fragment ions that must have been identified in a spectrum for it to be considered.

`minMz` numeric the minimum mass-to-charge ratio of a precursor ion.

`maxMz` numeric the maximum mass-to-charge ration of a precursor ion.

`modelSpec` character full path to a model spectrum file (.csv). Alternatively built in model tables (in the extdata directory) can be used by just supplying the one letter amino acid code for the peptide (currently available are: "ALVLIAFAQYLQQCPFEDHVK" and "RHPYFYAPELLFFAK"). If supplying a custom table it must consist of the following mandatory columns ("mass", "intensity", "ionType" and "fixed or variable").

1. mass - m/z of fragment ions.
2. intensity - intensity of fragment ions can be either relative or absolute intensity
3. ionType - the identity of the B and Y fragments can optionally added here (e.g. [b6]2+, [y2]1+) or if not known such as for mixed disulfates this column can also contain empty fields.
4. fixed or variable - this column contains whether a fragment ion should be considered either 'fixed', 'variable' (i.e. modified) or if it is an empty field it will not be considered.

As default the following model spectra are included in the external data directory of the adductomics package:

1. 'modelSpectrum_ALVLIAFAQYLQQCPFEDHVK.csv'
2. 'modelSpectrum_RHPYFYAPELLFFAK.csv'

groupMzabs	numeric after hierarchical clustering of the spectra the dendrogram will be cut at this height (in Da) generating the mass groups.
groupRtDev	numeric after hierarchical clustering of the spectra the dendrogram will be cut at this height (in minutes) generating the retention time groups.
possFormMzabs	numeric the maximum absolute mass difference for matching adduct mass to possible formulae.
minMeanSpecSim	numeric minimum mean dot product similarity score (cosine) between the spectra of a group identified by hierarchical clustering. This parameter is set to prevent erroneous clustering of dissimilar spectra (default = 0.7).
idPossForm	integer if = 1 then the average adduct masses of each spectrum group will be matched against an internal database of possible formula to generate hypotheses. The default 0 mean this will not take place as the computation is potentially time consuming.
outputPlotDir	character (default = NULL) output directory for plots.

Value

dataframe of putative adducts

Examples

```
## Not run:
eh = ExperimentHub();
temp = query(eh, 'adductData');
specSimPepId(MS2Dir=hubCache(temp),nCores=2,
rtDevModels=paste0(hubCache(temp),'/rtDevModels.RData'))

## End(Not run)
```

---

spectraCreate *Deconvolute both MS2 and MS1 levels scans adductomics*

Description

Deconvolute both MS2 and MS1 levels scans adductomics

Usage

```
spectraCreate(MS2file = NULL, TICfilter = 10000, DNF = 2, minInt =
100, minPeaks = 5)
```

Arguments

MS2file	character vector of mzXML file locations
TICfilter	numeric minimimum total ion current of an MS/MS scan. Any MS/MS scan below this value will be filtered out (default=0).

DNF	dynamic noise filter minimum signal to noise threshold (default = 2), calculated as the ratio between the linear model predicted intensity value and the actual intensity.
minInt	numeric minimum intensity value
minPeaks	minimum number of signal peaks following dynamic noise filtration (default = 5).

Value

list of MS2 spectra

`truePeakTrough` *true peak and trough detection*

Description

true peak and trough detection

Usage

```
truePeakTrough(peaksTmp = NULL, troughsTmp = NULL, peakRangeTmp =
NULL, minRes = 50, lrRes = FALSE)
```

Arguments

peaksTmp	character vector with indices of detected peaks from findPeaks
troughsTmp	character vector with indices of detected troughs from findPeaks
peakRangeTmp	matrix of the peak range data with at least 3 columns (1. mass-to-charge, 2. intensity, 3. retention time)
minRes	numeric minimum percentage left/right resolution
lrRes	logical if true both the left and right troughs must be above the minRes else the peak will be discounted. (default = FALSE i.e. if only the left or right trough is less than minRes then the peak will be retained)

Value

a named numeric of both the peaks and troughs fitting the criteria.

Index

∗ **datasets**
AdductQuantif-class, 4
AdductSpec-class, 5

adductQuant, 3
AdductQuantif-class, 4
AdductSpec-class, 5
adductSpecGen, 6

c,AdductQuantif-method *(*AdductQuantif-class*)*, 4
c,AdductSpec-method *(*AdductSpec-class*)*, 5
c,file.paths,peakQuantTable,peakIdData,AdductQuantif-method *(*AdductQuantif-class*)*, 4
c,Specfile.paths,adductMS2spec,metaData,Parameters,AdductSpec-method *(*AdductSpec-class*)*, 5

Digest, *7*, *8*
digestMod, 7
dotProdMatrix, 8
dotProdSpectra, 9
dynamicNoiseFilter, *6*, 9

filterAdductTable, 10
findPeaks, 11

generateTargTable, 12
groupMS2spec,rtDevModels,sumAdductType,Parameters,AdductSpec-class *(*AdductSpec-class*)*, 5
groupMS2spec,rtDevModels,sumAdductType,Peptides,AdductSpec-method *(*AdductSpec-class*)*, 5

hclust, *9*

IsotopicDistribution, *16*
IsotopicDistributionMod, 12

loess, *21*
loessWrapperMod, 13

mad, *19*
ms2Group, 13

nAdjPeaks, 14

outputPeakTable, 15

peakIdData,predIsoDist,targTable,AdductQuantif,AdductQu *(*AdductQuantif-class*)*, 4
peakIdQuant_newMethod, 15
peakIntegrate, 16
peakListId, 17
peakRangeSum, 18
Peptides,specPepMatches,AdductSpec-class *(*AdductSpec-class*)*, 5
predIsoDist,targTable,AdductQuantif-method *(*AdductQuantif-class*)*, 4
probPeaks, 19

retentionCorr, 20
rtDevModelling, 20
rtDevModelSave, 22

show,c,file.paths,peakQuantTable,AdductQuantif,AdductQu *(*AdductQuantif-class*)*, 4
show,c,Specfile.paths,adductMS2spec,metaData,AdductSpec *(*AdductSpec-class*)*, 5
signalGrouping, *6*, 23
smth.gaussian, *16*, *18*
specPepMatches,AdductSpec-method *(*AdductSpec-class*)*, 5
specSimPepId, 23
spectraCreate, 25

truePeakTrough, 26

x *(*AdductSpec-class*)*, 5