Package: enviGCMS (via r-universe)

October 9, 2026

Type Package

Title GC/LC-MS Data Analysis for Environmental Science

Version 0.9.1

Description Gas/Liquid Chromatography-Mass Spectrometer(GC/LC-MS) Data Analysis for Environmental Science. This package covered topics such molecular isotope ratio, matrix effects and Short-Chain Chlorinated Paraffins analysis etc. in environmental analysis.

URL https://yufree.github.io/enviGCMS/

BugReports https://github.com/yufree/enviGCMS/issues

License GPL-2

Encoding UTF-8

LazyData true

Suggests knitr, testthat, plotly, shiny, rmarkdown, DT, crosstalk, animation, mixtools, RaMS

VignetteBuilder knitr

Depends R (>= 3.5)

Imports Rcpp, BiocParallel, grDevices, graphics, stats, utils, RColorBrewer, data.table, igraph

LinkingTo Rcpp

Config/roxygen2/version 8.1.1

Config/pak/sysreqs libglpk-dev libxml2-dev

Repository https://ymgroup.r-universe.dev

Date/Publication 2026-10-09 14:24:20 UTC

RemoteUrl https://github.com/yufree/enviGCMS

RemoteRef HEAD

RemoteSha 0b46f157b3638ae5c247f888c2e5f6aeb109c643

Contents

c.mzrt . . . 3
checkGoldenRules . . . 4
cleanMGF . . . 5
findline . . . 6
findlipid . . . 7
findmet . . . 7
findohc . . . 8
findpfc . . . 9
getalign . . . 10
getalign2 . . . 10
getcompare . . . 11
getcsv . . . 12
getdoe . . . 13
getdwtus . . . 14
getfilter . . . 14
getformula . . . 15
getHRMF . . . 16
getimputation . . . 17
getintegration . . . 18
getisotopologues . . . 19
getmass . . . 20
getmassdefect . . . 20
getmdh . . . 21
getmdr . . . 21
getms1anno . . . 22
getMSP . . . 23
getmzrtcsv . . . 23
getoverlappeak . . . 24
getpn . . . 24
getpower . . . 25
getpqsi . . . 26
getrangecsv . . . 26
getretcor . . . 27
getrmd . . . 27
gifmr . . . 28
HRMF . . . 29
integration . . . 31
list . . . 31
ma . . . 32
matrix . . . 32
plotcc . . . 33
plotden . . . 33
plotdwtus . . . 34
plotEIC . . . 34
plotgroup . . . 35
plothist . . . 36

plothm . . . 36
plotint . . . 37
plotintslope . . . 37
plotkms . . . 38
plotmr . . . 38
plotmrc . . . 39
plotms . . . 40
plotmz . . . 40
plotpca . . . 41
plotpeak . . . 42
plotridge . . . 43
plotridges . . . 43
plotrla . . . 44
plotrsd . . . 45
plotrug . . . 45
plotsms . . . 46
plotsub . . . 46
plott . . . 47
plottic . . . 47
plotTopMS1Peaks . . . 48
plotTopMS2Peaks . . . 49
runMDPlot . . . 50
runsccp . . . 50
sccp . . . 51
scoreIsotopes . . . 51
TBBPA . . . 52
writeMSP . . . 53

Index **54**

`c.mzrt` *c method for mzrt object*

Description

c method for mzrt object

Usage

```
## S3 method for class 'mzrt'
c(x, ...)
```

Arguments

`x`	an mzrt object
`...`	further mzrt objects to concatenate

Value

a combined mzrt object

`checkGoldenRules` *Check Chemical Formula against Fiehn Seven Golden Rules*

Description

Evaluates whether a candidate chemical formula satisfies heuristic chemical rules based on Senior's valence rules (Senior 1951), Hydrogen-to-Carbon (H/C) ratios, heteroatom ratios (N/C, O/C, P/C, S/C, Halogen/C), and Double Bond Equivalents (DBE) as described by Kind & Fiehn (2007).

Usage

```
checkGoldenRules(formula, z = 0, min_hc = 0.1, max_hc = 6, max_dbe = 40)
```

Arguments

`formula`	Character string representing molecular formula (e.g. `"C6H12O6"`).
`z`	Integer charge state (default: 0).
`min_hc`	Numeric minimum H/C ratio (default: 0.1).
`max_hc`	Numeric maximum H/C ratio (default: 6.0).
`max_dbe`	Numeric maximum DBE (default: 40.0).

Value

A list containing booleans indicating whether the formula passed Senior rules, H/C ratio, heteroatom ratios, DBE, nitrogen rule, and overall validity.

References

Kind, T., Fiehn, O. Seven Golden Rules for heuristic filtering of molecular formulas obtained by accurate mass spectrometry. BMC Bioinformatics 8, 105 (2007). doi:10.1186/147121058105

Senior, J. K. Partitions and their connection with the problems of chemical structure. J. Chem. Phys. 19, 865-873 (1951).

Examples

```
checkGoldenRules("C6H12O6")
checkGoldenRules("C3H32N12O3") # chemically impossible formula
```

cleanMGF *Clean an MGF file by keeping only formula-explainable peaks*

Description

For each MS2 spectrum, the precursor neutral mass is derived from PEPMASS, CHARGE and the supplied adduct, candidate parent formulae are enumerated with the native HORIZON engine, and fragment peaks that can be explained as sub-formulae of the best candidate (the one explaining the most peaks) are kept. Spectra without PEPMASS or without any candidate formula are written back unchanged.

Usage

```
cleanMGF(
  file,
  out_file = NULL,
  adduct = "[M+H]+",
  charge = NULL,
  mass_accuracy = 5,
  elements = "CHNOPS",
  max_candidates = 3,
  intensity_cutoff = 0,
  min_peaks = 2
)
```

Arguments

file	character. Path to an MGF file.
out_file	character. Output MGF path. Default NULL writes <file stem>_clean.mgf next to the input file.
adduct	character. Single adduct name (see HRMF), e.g. "[M+H]+". Fragments are assumed singly-charged with this adduct.
charge	integer. Precursor charge state used when the CHARGE header is missing. Default NULL uses 1 (from the adduct sign).
mass_accuracy	numeric. Mass accuracy in ppm for both precursor decomposition and fragment matching. Default 5.
elements	character. Elements allowed in precursor formula enumeration, e.g. "CHNOPS". Default "CHNOPS".
max_candidates	integer. Maximum number of precursor formula candidates to evaluate per spectrum. Default 3.
intensity_cutoff	numeric. Discard peaks below this absolute intensity before annotation. Default 0 (keep all).
min_peaks	integer. Drop spectra left with fewer than this many peaks after cleaning. Default 2.

Details

The cleaned file preserves the original header lines and peak line formatting; the chosen parent formula is appended as a FORMULA= header when a candidate is found.

Value

Invisibly, a data.frame with one row per spectrum: title, pepmass, charge, chosen formula, peak counts before/after, kept fraction and status ("cleaned", "kept" = unchanged, or "dropped").

See Also

HRMF, getMSP

Examples

```
## Not run:
summary <- cleanMGF("ms2.mgf", adduct = "[M+H]+")
summary[, c("title", "formula", "n_before", "n_kept", "status")]

## End(Not run)
```

---

findline *find line of the regression model for GC-MS*

Description

find line of the regression model for GC-MS

Usage

```
findline(data, threshold = 2, temp = c(100, 320))
```

Arguments

data	imported data matrix of GC-MS
threshold	the threshold of the response (log based 10)
temp	the scale of the oven temperature (constant rate)

Value

list linear regression model for the matrix

Examples

```
## Not run:
data(matrix)
findline(matrix)

## End(Not run)
```

`findlipid` *Find lipid class of metabolites base on referenced Kendrick mass defect*

Description

Find lipid class of metabolites base on referenced Kendrick mass defect

Usage

```
findlipid(list, mode = "pos")
```

Arguments

`list`	list with data as peaks list, mz, rt and group information, retention time should be in seconds
`mode`	'pos' for positive mode, 'neg' for negative mode and 'none' for neutral mass, only support [M+H] and [M-H] for each mode

Value

list list with dataframe with the lipid referenced Kendrick mass defect(RKMD) and logical for class

References

Method for the Identification of Lipid Classes Based on Referenced Kendrick Mass Analysis. Lerno LA, German JB, Lebrilla CB. Anal Chem. 2010 May 15;82(10):4236-45.

Examples

```
data(list)
RKMD <- findlipid(list)
```

`findmet` *Screen metabolites by Mass Defect*

Description

Screen metabolites by Mass Defect

Usage

```
findmet(list, mass, mdr = 50)
```

Arguments

list	list with data as peaks list, mz, rt and group information, retention time should be in seconds
mass	mass to charge ratio of specific compounds
mdr	mass defect range, default 50mDa

Value

list with filtered metabolites mass to charge index of certain compound

---

findohc *Screen organohalogen compounds*

Description

Screen organohalogen compounds by retention time, mass defect analysis and isotope relationship modified by literature report. Also support compounds with [M] and [M+2] ratio cutoff.

Usage

```
findohc(
  list,
  sf = 78/77.91051,
  step = 0.001,
  stepsd1 = 0.003,
  stepsd2 = 0.005,
  mzc = 700,
  cutoffint = 1000,
  cutoffr = 0.4,
  clustercf = 10
)
```

Arguments

list	list with data as peaks list, mz, rt and group information, retention time should be in seconds
sf	scale factor, default 78/77.91051(Br)
step	mass defect step, default 0.001
stepsd1	mass defect uncertainty for lower mass, default 0.003
stepsd2	mass defect uncertainty for higher mass, default 0.005
mzc	threshold of lower mass and higher mass, default 700
cutoffint	the cutoff of intensity, default 1000
cutoffr	the cutoff of [M] and [M+2] ratio, default 0.4
clustercf	the cutoff of cluster analysis to separate two different ions groups for retention time, default 10

Value

list with filtered organohalogen compounds

References

Identification of Novel Brominated Compounds in Flame Retarded Plastics Containing TBBPA by Combining Isotope Pattern and Mass Defect Cluster Analysis Ana Ballesteros-Gomez, Joaquin Ballesteros, Xavier Ortiz, Willem Jonker, Rick Helmus, Karl J. Jobst, John R. Parsons, and Eric J. Reiner Environmental Science & Technology 2017 51 (3), 1518-1526 DOI: 10.1021/acs.est.6b03294

`findpfc` *Find PFCs based on mass defect analysis*

Description

Find PFCs based on mass defect analysis

Usage

```
findpfc(list)
```

Arguments

`list`	list with data as peaks list, mz, rt and group information, retention time should be in seconds

Value

list list with potential PFCs compounds index

References

Liu, Y.; D'Agostino, L. A.; Qu, G.; Jiang, G.; Martin, J. W. High-Resolution Mass Spectrometry (HRMS) Methods for Nontarget Discovery and Characterization of Poly- and per-Fluoroalkyl Substances (PFASs) in Environmental and Human Samples. TrAC Trends in Analytical Chemistry 2019, 121, 115420.

Examples

```
data(list)
pfc <- findpfc(list)
```

getalign *Align two peaks vectors by mass to charge ratio and/or retention time*

Description

Align two peaks vectors by mass to charge ratio and/or retention time

Usage

```
getalign(mz1, mz2, rt1 = NULL, rt2 = NULL, ppm = 10, deltart = 10)
```

Arguments

mz1	the mass to charge of reference peaks
mz2	the mass to charge of peaks to be aligned
rt1	retention time of reference peaks
rt2	retention time of peaks to be aligned
ppm	mass accuracy, default 10
deltart	retention time shift table, default 10 seconds

Value

data frame with aligned peaks table

Examples

```
mz1 <- c(221.1171, 227.1390, 229.1546, 233.1497, 271.0790 )
mz2 <- c(282.279, 281.113, 227.139, 227.139, 302.207)
rt1 <- c(590.8710, 251.3820, 102.9230, 85.8850, 313.8240)
rt2 <- c(787.08, 160.02, 251.76, 251.76, 220.26)
getalign(mz1,mz2,rt1,rt2)
```

getalign2 *Align mass to charge ratio and/or retention time to remove redundancy*

Description

Align mass to charge ratio and/or retention time to remove redundancy

Usage

```
getalign2(mz, rt, ppm = 5, deltart = 5)
```

Arguments

mz	the mass to charge of reference peaks
rt	retention time of reference peaks
ppm	mass accuracy, default 10
deltart	retention time shift table, default 10 seconds

Value

index for

Examples

```
mz <- c(221.1171, 221.1170, 229.1546, 233.1497, 271.0790 )
rt <- c(590.8710, 587.3820, 102.9230, 85.8850, 313.8240)
getalign2(mz,rt)
```

---

getcompare *Align multiple peaks list to one peak list*

Description

Align multiple peaks list to one peak list

Usage

```
getcompare(..., index = 1, ppm = 5, deltart = 5)
```

Arguments

...	peaks list, mzrt objects
index	numeric, the index of reference peaks.
ppm	pmd mass accuracy, default 5
deltart	retention time shift table, default 10 seconds

Value

list object with aligned mzrt objects

getcsv *Convert an list object to csv file*

Description

Convert an list object to csv file

Usage

```
getcsv(list, name, mzdigit = 4, rtdigit = 1, type = "o", target = FALSE, ...)
```

Arguments

list	list with data as peaks list, mz, rt and group information
name	result name for csv and/or eic file, default NULL
mzdigit	m/z digits of row names of data frame, default 4
rtdigit	retention time digits of row names of data frame, default 1
type	csv format for further analysis, m means Metaboanalyst, a means xMSannotator, p means Mummichog(NA values are imputed by 'getimputation', and F test is used here to generate stats and p value), o means full information csv (for 'pmd' package), default o. mapo could output all those format files.
target	logical, preserve original rowname of data or not for target data, default FALSE.
...	other parameters for 'write.table'

Value

NULL, csv file

References

Li, S.; Park, Y.; Duraisingham, S.; Strobel, F. H.; Khan, N.; Soltow, Q. A.; Jones, D. P.; Pulendran, B. PLOS Computational Biology 2013, 9 (7), e1003123. Xia, J., Sinelnikov, I.V., Han, B., Wishart, D.S., 2015. MetaboAnalyst 3.0 - making metabolomics more meaningful. Nucl. Acids Res. 43, W251-W257.

Examples

```
## Not run:
data(list)
getcsv(list,name='demo')

## End(Not run)
```

getdoe *Generate the group level rsd and average intensity based on DoE,*

Description

Generate the group level rsd and average intensity based on DoE,

Usage

```
getdoe(
  list,
  inscf = 5,
  rsdcf = 100,
  rsdcft = 30,
  imputation = "l",
  tr = FALSE,
  BPPARAM = BiocParallel::bpparam()
)
```

Arguments

list	list with data as peaks list, mz, rt and group information
inscf	Log intensity cutoff for peaks across samples. If any peaks show a intensity higher than the cutoff in any samples, this peaks would not be filtered. default 5
rsdcf	the rsd cutoff of all peaks in all group
rsdcft	the rsd cutoff of all peaks in technical replicates
imputation	parameters for 'getimputation' function method
tr	logical. TRUE means dataset with technical replicates at the base level folder
BPPARAM	An optional BiocParallelParam instance determining the parallel back-end to be used during evaluation.

Value

list with group mean, standard deviation, and relative standard deviation for all peaks, and filtered peaks index

See Also

getimputation, getpower

Examples

```
data(list)
getdoe(list)
```

getdwtus *Density weighted intensity for one sample*

Description

Density weighted intensity for one sample

Usage

```
getdwtus(peak, n = 512, log = FALSE)
```

Arguments

peak	peaks intensity one sample
n	the number of equally spaced points at which the density is to be estimated, default 512
log	log transformation

Value

Density weighted intensity for one sample

Examples

```
data(list)
getdwtus(list$data[,1])
```

getfilter *Filter the data based on row and column index*

Description

Filter the data based on row and column index

Usage

```
getfilter(list, rowindex = TRUE, colindex = TRUE, name = NULL, type = "o", ...)
```

Arguments

list	list with data as peaks list, mz, rt and group information
rowindex	logical, row index to keep
colindex	logical, column index to keep
name	file name for csv and/or eic file, default NULL
type	csv format for further analysis, m means Metaboanalyst, a means xMSannotator, p means Mummichog(NA values are imputed by 'getimputation', and F test is used here to generate stats and p value), o means full information csv (for 'pmd' package), default o. mapo could output all those format files.
...	other parameters for 'getcsv'

Value

list with remain peaks, and filtered peaks index

See Also

getimputation, getcsv

Examples

```
data(list)
li <- getdoe(list)
lif <- getfilter(li,rowindex = li$rsdindex)
```

---

getformula *Get chemical formula for mass to charge ratio using the HORIZON algorithm*

Description

Rapidly decomposes accurate mass-to-charge ratios into candidate chemical formulas using the native HORIZON (Heavy-first Ordered Recursive Inference with Zero-loop Optimal Navigation) engine. Supports chemical plausibility filtering via Double Bond Equivalents (DBE), Nitrogen rule, and Kind & Fiehn Seven Golden Rules.

Usage

```
getformula(
  mz,
  charge = 0,
  window = 0.001,
 elements = list(C = c(1, 50), H = c(1, 50), N = c(0, 50), O = c(0, 50), P = c(0, 1), S
    = c(0, 1)),
  golden_rules = TRUE,
  detailed = FALSE,
```

```
  resolution = 0,
  nthreads = 1
)
```

Arguments

mz	a vector with mass to charge ratio
charge	The charge value of the formula, default 0 for autodetect
window	The window accuracy in the same units as mass
elements	Elements list to take into account.
golden_rules	logical, whether to filter by Fiehn Seven Golden Rules (default: TRUE)
detailed	logical, if TRUE return a list of data.frames with formula, exact mass, error, DBE, H/C ratio, Senior rules, and validation details; if FALSE (default), return a list of character vectors of valid formulas
resolution	numeric, mass spectrometer resolving power for isotope peak merging (default: 0)
nthreads	integer, number of OpenMP parallel threads for batch processing (default: 1)

Value

list of chemical formulas (or data.frames if detailed = TRUE)

---

getHRMF *Batch High Resolution Mass Filtering for all compounds in an MSP file*

Description

Applies HRMF to every compound in an MSP file that has a chemical formula annotation.

Usage

```
getHRMF(
  file,
  charge = 1,
  adduct = NULL,
  mass_accuracy = 5,
  intensity_cutoff = 1,
  IR_RelAb_cutoff = 1
)
```

Arguments

file	character. Path to an MSP file.
charge	integer. Charge state (see HRMF). Default 1. Ignored when adduct is supplied.
adduct	character vector of adduct names or named numeric vector of custom mass deltas (see HRMF). Default NULL.
mass_accuracy	numeric. Mass accuracy in ppm. Default 5.
intensity_cutoff	numeric. Minimum absolute intensity. Default 1.
IR_RelAb_cutoff	numeric. Relative abundance cutoff (%) for theoretical isotopologues. Default 1.

Value

A named list where each element contains the HRMF results for one compound (with a valid formula) from the MSP file.

See Also

HRMF, getMSP

Examples

```
## Not run:
results <- getHRMF("library.msp")

# LC-MS/MS library with protonated ions
results <- getHRMF("library.msp", adduct = "[M+H]+")

## End(Not run)
```

---

getimputation *Impute the peaks list data*

Description

Impute the peaks list data

Usage

```
getimputation(list, method = "l")
```

Arguments

list	list with data as peaks list, mz, rt and group information
method	'r' means remove, 'l' means use half the minimum of the values across the peaks list, 'mean' means mean of the values across the samples, 'median' means median of the values across the samples, '0' means 0, '1' means 1. Default 'l'.

Value

list with imputed peaks

See Also

getdoe

Examples

```
data(list)
getimputation(list)
```

`getintegration` *getintegration was mainly used for get the integration of certain ion's chromatogram data and plot the data*

Description

getintegration was mainly used for get the integration of certain ion's chromatogram data and plot the data

Usage

```
getintegration(
  data,
  rt = c(8.3, 9),
  n = 5,
  m = 5,
  slope = c(2, 2),
  baseline = 10,
  noslope = TRUE,
  smoothit = TRUE,
  half = FALSE
)
```

Arguments

`data`	file should be a dataframe with the first column RT and second column intensity of the SIM ions.
`rt`	a rough RT range contained only one peak to get the area
`n`	points in the moving average smooth box, default value is 5
`m`	numbers of points for regression to get the slope
`slope`	the threshold value for start/stop peak as percentage of max slope
`baseline`	numbers of the points for the baseline of the signal
`noslope`	logical, if using a horizon line to get area or not
`smoothit`	logical, if using an average smooth box or not. If using, n will be used
`half`	logical, if using the left half peak to calculate the area

Value

integration data such as peak area, peak height, signal and the slope data.

Examples

```
## Not run:
list <- getintegration(data)

## End(Not run)
```

`getisotopologues` *Get the selected isotopologues at certain MS data*

Description

Get the selected isotopologues at certain MS data

Usage

```
getisotopologues(formula = "C6H11O6", charge = 1, width = 0.3, cutoff = 0.05)
```

Arguments

`formula`	the molecular formula. 'C6H11O6' as default
`charge`	the charge of that molecular. 1 in EI mode as default
`width`	the width of the peak width on mass spectrum. 0.3 as default for low resolution mass spectrum.
`cutoff`	numeric, minimum relative abundance (fraction of base peak) to consider, default 0.05 (5%).

Examples

```
## Not run:
# show isotopologues
getisotopologues(formula = 'C6H11O6', charge = 1, width = 0.3)

## End(Not run)
```

`getmass` *Get the exact mass of the isotopologues from a chemical formula or reaction's isotope patterns with the highest abundances*

Description

Get the exact mass of the isotopologues from a chemical formula or reaction's isotope patterns with the highest abundances

Usage

```
getmass(data)
```

Arguments

`data`	a chemical formula or reaction e.g. 'Cl-H', 'C2H4'

Value

numerical vector

`getmassdefect` *Get mass defect with certain scaled factor*

Description

Get mass defect with certain scaled factor

Usage

```
getmassdefect(mass, sf)
```

Arguments

`mass`	vector of mass
`sf`	scaled factors

Value

dataframe with mass, scaled mass and scaled mass defect

See Also

`plotkms`

Examples

```
mass <- c(100.1022,245.2122,267.3144,400.1222,707.2294)
sf <- 0.9988
mf <- getmassdefect(mass,sf)
```

`getmdh` *Get the high order unit based Mass Defect*

Description

Get the high order unit based Mass Defect

Usage

```
getmdh(mz, cus = c("CH2,H2"), method = "round")
```

Arguments

`mz`	numeric vector for exact mass
`cus`	chemical formula or reaction
`method`	you could use ‘round‘, ‘floor‘ or ‘ceiling‘

Value

high order Mass Defect with details

Examples

```
## Not run:
getmdh(getmass('C2H4'))

## End(Not run)
```

`getmdr` *Get the raw Mass Defect*

Description

Get the raw Mass Defect

Usage

```
getmdr(mz)
```

Arguments

mz	numeric vector for exact mass

Value

raw Mass Defect

Examples

```
getmdr(28.0313)
```

---

getms1anno *Annotation of MS1 data by compounds database by predefined paired mass distance*

Description

Annotation of MS1 data by compounds database by predefined paired mass distance

Usage

```
getms1anno(pmd, mz, ppm = 10, db = NULL)
```

Arguments

pmd	adducts formula or paired mass distance for ions
mz	unknown mass to charge ratios vector
ppm	mass accuracy
db	compounds database as dataframe. Two required columns are name and monoisotopic molecular weight with column names of name and mass

Value

list or data frame

getMSP *read in MSP file as list for ms/ms or ms(EI) annotation*

Description

read in MSP file as list for ms/ms or ms(EI) annotation

Usage

```
getMSP(file)
```

Arguments

file	the path to your MSP file

Value

list a list with MSP information for annotation

getmzrtcsv *Covert the peaks list csv file into list*

Description

Covert the peaks list csv file into list

Usage

```
getmzrtcsv(path)
```

Arguments

path	the path to your csv file

Value

list with rtmz profile and group information as the first row

See Also

getMSP

getoverlappeak *Get the overlap peaks by mass and retention time range*

Description

Get the overlap peaks by mass and retention time range

Usage

```
getoverlappeak(list1, list2)
```

Arguments

list1	list with data as peaks list, mz, rt, mzrange, rtrange and group information to be overlapped
list2	list with data as peaks list, mz, rt, mzrange, rtrange and group information to overlap

Value

logical index for list 1's peaks

See Also

getimputation,getdoe

getpn *Merge positive and negative mode data*

Description

Merge positive and negative mode data

Usage

```
getpn(pos, neg, ppm = 5, pmd = 2.02, digits = 2, cutoff = 0.9)
```

Arguments

pos	a list with mzrt profile collected from positive mode. The sample order should match the negative mode.
neg	a list with mzrt profile collected from negative mode.The sample order should match the positive mode.
ppm	pmd mass accuracy, default 5
pmd	numeric or numeric vector
digits	mass or mass to charge ratio accuracy for pmd, default 2
cutoff	correlation coefficients, default 0.9

Value

mzrt object with group information from pos mode

`getpower` *Get the index with power restriction for certain study with BH adjusted p-value and certain power*

Description

Get the index with power restriction for certain study with BH adjusted p-value and certain power

Usage

```
getpower(list, pt = 0.05, qt = 0.05, powert = 0.8, imputation = "l")
```

Arguments

`list`	list with data as peaks list, mz, rt and group information
`pt`	p value threshold, default 0.05
`qt`	q value threshold, BH adjust, default 0.05
`powert`	power cutoff, default 0.8
`imputation`	parameters for ‘getimputation‘ function method

Value

list with current power and sample numbers for each peaks

See Also

`getimputation`, `getdoe`

Examples

```
data(list)
getpower(list)
```

getpqsi *Compute pooled QC linear index according to run order*

Description

Compute pooled QC linear index according to run order

Usage

```
getpqsi(data, order, n = 5)
```

Arguments

data	peaks intensity list with row as peaks and column as samples
order	run order of pooled QC samples
n	samples numbers used for linear regression

Value

vector for the peaks proportion with significant changes in linear regression after FDR control.

getrangecsv *Get a mzrt list and/or save mz and rt range as csv file*

Description

Get a mzrt list and/or save mz and rt range as csv file

Usage

```
getrangecsv(list, name, ...)
```

Arguments

list	list with data as peaks list, mz, rt and group information
name	result name for csv and/or eic file, default NULL
...	other parameters for ‘write.table‘

Value

NULL, csv file

getretcor *Perform peaks list alignment and return features table*

Description

Perform peaks list alignment and return features table

Usage

```
getretcor(list, ts = 1, ppm = 10, deltart = 5, FUN)
```

Arguments

list	each element should be a data.frame with mz, rt and intensity as m/z, retention time in seconds and intensity of certain peaks.
ts	template sample index in the list, default 1
ppm	mass accuracy, default 10
deltart	retention time shift table, default 5 seconds
FUN	function to deal with multiple aligned peaks from one sample

Value

mzrt object without group information

getrmd *Get the Relative Mass Defect*

Description

Get the Relative Mass Defect

Usage

```
getrmd(mz)
```

Arguments

mz	numeric vector for exact mass

Value

Relative Mass Defect

Examples

```
getrmd(28.0313)
```

gifmr *plot scatter plot for rt-mz profile and output gif file for multiple groups*

Description

plot scatter plot for rt-mz profile and output gif file for multiple groups

Usage

```
gifmr(
  list,
  ms = c(100, 500),
  rsdcf = 30,
  inscf = 5,
  imputation = "l",
  name = "test",
  ...
)
```

Arguments

list	list with data as peaks list, mz, rt and group information
ms	the mass range to plot the data
rsdcf	the rsd cutoff of all peaks in all group
inscf	Log intensity cutoff for peaks across samples. If any peaks show a intensity higher than the cutoff in any samples, this peaks would not be filtered. default 5
imputation	parameters for ‘getimputation‘ function method
name	file name for gif file, default test
...	parameters for ‘plot‘ function

Value

gif file

Examples

```
## Not run:
data(list)
gifmr(list)

## End(Not run)
```

HRMF *High Resolution Mass Filtering (HRMF) for GC/LC HRMS data*

Description

Performs high-resolution mass filtering by matching experimental mass spectral peaks against theoretical isotope patterns derived from one or more candidate chemical formulae. Calculates forward HRMF score, reverse (MSP) score, and Figure of Merit (FoM) for each candidate.

Usage

```
HRMF(
  msp,
  formula,
  charge = 1,
  adduct = NULL,
  mass_accuracy = 5,
  intensity_cutoff = 1,
  IR_RelAb_cutoff = 1,
  detailed = FALSE
)
```

Arguments

msp	list. A single compound entry as returned by getMSP, containing at least a spectra element with columns mz and intensity.
formula	character vector. One or more candidate chemical formulae to evaluate.
charge	integer. Charge state: 1 for positive, -1 for negative, 0 for neutral. Default 1 (radical cation in EI). Ignored when adduct is supplied.
adduct	character vector of adduct names for singly-charged ESI ions (e.g. "[M+H]+", c("[M+H]+", "[M+Na]+")), or a named numeric vector of custom mass deltas in Da with names ending in "+" or "-" (e.g. c("[M+MeOH]+" = 33.033491)). Default NULL keeps the legacy radical-ion/neutral behaviour governed by charge. Built-in adducts: "[M+H]+", "[M+Na]+", "[M+NH4]+", "[M+K]+", "[M+H-H2O]+", "[M-H]-", "[M+Cl]-", "[M+HCOO]-", "[M+CH3COO]-".
mass_accuracy	numeric. Mass accuracy in ppm. Default 5.
intensity_cutoff	numeric. Minimum absolute intensity to retain a peak. Default 1.
IR_RelAb_cutoff	numeric. Relative abundance cutoff (%) for theoretical isotopologues. Default 1.
detailed	logical. If TRUE, return detailed list with all_ions, compound, and HRMF_scores for each formula. If FALSE (default), return a summary data.frame.

Details

The method is based on Kwiecien et al. (2015) doi:10.1021/acs.analchem.5b01503.

Unlike the original MSxplorer implementation, this version uses the native HORIZON (Heavy-first Ordered Recursive Inference with Zero-loop Optimal Navigation) engine via Rcpp for both high-performance formula decomposition and isotope pattern calculation, requiring no external dependencies.

The input `msp` should be a single entry from the list returned by `getMSP`. For batch processing of entire MSP files, see `getHRMF`.

For EI (GC-MS) radical ions use the default `charge` argument. For ESI (LC-MS/MS) data with even-electron adduct ions such as [M+H]+ or [M-H]-, supply `adduct` so that fragment m/z values are converted internally; the adduct should describe how fragment ions are formed (usually the same as the precursor adduct for singly-charged fragmentation).

Value

If `detailed = FALSE`, a data.frame with one row per candidate formula (and per adduct, if supplied) and columns: Candidate, Adduct, peak_count_forw, df_theortomsp, HRMF_theor_score, peak_count_rev, df_msptotheor, HRMF_msp_score, FoM. If `detailed = TRUE`, a named list of detailed results per formula/adduct.

See Also

`getHRMF` for batch processing, `getMSP` for reading MSP files.

Examples

```
## Not run:
# Read MSP file and run HRMF on the first compound (EI radical cation)
msp_data <- getMSP("spectrum.msp")
result <- HRMF(msp_data[[1]], formula = "C8H11NO")

# Compare multiple candidates
result <- HRMF(msp_data[[1]], formula = c("C8H11NO", "C7H9NO2"))

# LC-MS/MS with protonated fragments
result <- HRMF(msp_data[[1]], formula = "C8H10N4O2", adduct = "[M+H]+")

# Multiple candidate adducts are scored separately
result <- HRMF(msp_data[[1]], formula = "C8H10N4O2",
               adduct = c("[M+H]+", "[M+Na]+"))

## End(Not run)
```

integration *Just integrate data according to fixed rt and fixed noise area*

Description

Just integrate data according to fixed rt and fixed noise area

Usage

```
integration(data, rt = c(8.3, 9), brt = c(8.3, 8.4), smoothit = TRUE)
```

Arguments

data	file should be a dataframe with the first column RT and second column intensity of the SIM ions.
rt	a rough RT range contained only one peak to get the area
brt	a rough RT range contained only one peak and enough noises to get the area
smoothit	logical, if using an average smooth box or not. If using, n will be used

Value

area integration data

Examples

```
## Not run:
area <- integration(data)

## End(Not run)
```

list *Demo data*

Description

Demo data

Usage

```
data(list)
```

Format

A list object with data, mass to charge ratio, retention time and group information. The list is generated from faahKO package.

ma *filter data by average moving box*

Description

filter data by average moving box

Usage

```
ma(x, n)
```

Arguments

x	a vector
n	A number to identify the size of the moving box.

Value

The filtered data

Examples

```
ma(rnorm(1000),5)
```

matrix *Demo raw data matrix*

Description

Demo raw data matrix

Usage

```
data(matrix)
```

Format

A matrix object from raw mass spectrometry data. The list is generated from faahKO package.

plotcc *Plot the calibration curve with error bar, r squared and equation*

Description

Plot the calibration curve with error bar, r squared and equation

Usage

```
plotcc(x, y, upper, lower = upper, ...)
```

Arguments

x	concentration
y	response
upper	upper error bar
lower	lower error bar
...	parameters for 'plot' function

Examples

```
## Not run:
plotcc(x,y,upper)

## End(Not run)
```

plotden *plot the density for multiple samples*

Description

plot the density for multiple samples

Usage

```
plotden(data, lv, index = NULL, name = NULL, lwd = 1, ...)
```

Arguments

data	data row as peaks and column as samples
lv	group information
index	index for selected peaks
name	name on the figure for samples
lwd	the line width for density plot, default 1
...	parameters for 'plot' function

Examples

```
data(list)
plotden(list$data, lv = as.character(list$group$sample_group),ylim = c(0,1))
```

`plotdwtus` *plot density weighted intensity for multiple samples*

Description

plot density weighted intensity for multiple samples

Usage

```
plotdwtus(list, n = 512, ...)
```

Arguments

`list`	list with data as peaks list, mz, rt and group information
`n`	the number of equally spaced points at which the density is to be estimated, default 512
`...`	parameters for 'plot' function

Value

Density weighted intensity for multiple samples

Examples

```
data(list)
plotdwtus(list)
```

`plotEIC` *Plot extracted ion chromatograms from raw mzML/mzXML data*

Description

Reads mzML/mzXML raw data and plots extracted ion chromatograms (EIC) for a list of target m/z values at MS1 or MS2 level. Uses **RaMS** for lightweight data access and **plotly** for interactive visualization.

Usage

```
plotEIC(filepath, featlist, diff = 0.005)
```

Arguments

filepath	character. Path to an mzML or mzXML file.
featlist	data.frame with columns: name (feature label), mz (target m/z), ms_level (either "ms1" or "ms2").
diff	numeric. Half-width of the m/z extraction window in Da. Default 0.005.

Value

A **plotly** object with overlaid extracted ion chromatograms.

Examples

```
## Not run:
featlist <- data.frame(
  name = c("target1", "target2"),
  mz = c(300.1234, 350.5678),
  ms_level = c("ms1", "ms2")
)
plotEIC("sample.mzML", featlist)

## End(Not run)
```

---

plotgroup *Plot the response group of GC-MS*

Description

Plot the response group of GC-MS

Usage

```
plotgroup(data, threshold = 2)
```

Arguments

data	imported data matrix of GC-MS
threshold	the threshold of the response (log based 10) to separate the group

Value

list linear regression model for the data matrix

Examples

```
## Not run:
data(matrix)
plotgroup(matrix)

## End(Not run)
```

plothist *Plot the density of the GC-MS data with EM algorithm to separate the data into two log normal distribution*

Description

Plot the density of the GC-MS data with EM algorithm to separate the data into two log normal distribution

Usage

```
plothist(data)
```

Arguments

data	imported data matrix of GC-MS

Examples

```
## Not run:
# generate a matrix from raw data with row as m/z and column as retention time
plothist(matrix)

## End(Not run)
```

plothm *Plot the heatmap of mzrt profiles*

Description

Plot the heatmap of mzrt profiles

Usage

```
plothm(data, lv, index = NULL)
```

Arguments

data	data row as peaks and column as samples
lv	group information
index	index for selected peaks

Examples

```
data(list)
plothm(list$data, lv = as.factor(list$group$sample_group))
```

plotint *plot the information of integration*

Description

plot the information of integration

Usage

```
plotint(list, name = NULL)
```

Arguments

list	list from getinteagtion
name	the title of the plot

Examples

```
## Not run:
list <- getinteagtion(rawdata)
plotint(list)

## End(Not run)
```

plotintslope *plot the slope information of integration*

Description

plot the slope information of integration

Usage

```
plotintslope(list, name = NULL)
```

Arguments

list	list from getintegration
name	the title of the plot

Examples

```
## Not run:
list <- getinteragtion(rawdata)
plotintslope(list)

## End(Not run)
```

plotkms *plot the kendrick mass defect diagram*

Description

plot the kendrick mass defect diagram

Usage

```
plotkms(data, cutoff = 1000)
```

Arguments

data	vector with the name m/z
cutoff	remove the low intensity

See Also

getmassdefect

Examples

```
## Not run:
mz <- c(10000,5000,20000,100,40000)
names(mz) <- c(100.1022,245.2122,267.3144,400.1222,707.2294)
plotkms(mz)

## End(Not run)
```

plotmr *plot the scatter plot for peaks list with threshold*

Description

plot the scatter plot for peaks list with threshold

Usage

```
plotmr(
  list,
  rt = NULL,
  ms = NULL,
  inscf = 5,
  rsdcf = 30,
  imputation = "l",
  ...
)
```

Arguments

list	list with data as peaks list, mz, rt and group information
rt	vector range of the retention time
ms	vector vector range of the m/z
inscf	Log intensity cutoff for peaks across samples. If any peaks show a intensity higher than the cutoff in any samples, this peaks would not be filtered. default 5
rsdcf	the rsd cutoff of all peaks in all group, default 30
imputation	parameters for 'getimputation' function method
...	parameters for 'plot' function

Value

data fit the cutoff

Examples

```
data(list)
plotmr(list)
```

---

plotmrc *plot the diff scatter plot for peaks list with threshold between two groups*

Description

plot the diff scatter plot for peaks list with threshold between two groups

Usage

```
plotmrc(list, ms = c(100, 800), inscf = 5, rsdcf = 30, imputation = "l", ...)
```

Arguments

list	list with data as peaks list, mz, rt and group information
ms	the mass range to plot the data
inscf	Log intensity cutoff for peaks across samples. If any peaks show a intensity higher than the cutoff in any samples, this peaks would not be filtered. default 5
rsdcf	the rsd cutoff of all peaks in all group
imputation	parameters for 'getimputation' function method
...	parameters for 'plot' function

Examples

```
data(list)
plotmrc(list)
```

plotms *plot GC/LC-MS data as a heatmap with TIC*

Description

plot GC/LC-MS data as a heatmap with TIC

Usage

```
plotms(data, log = FALSE)
```

Arguments

data	imported data matrix of GC-MS
log	transform the intensity into log based 10

Value

heatmap

Examples

```
## Not run:
png('test.png')
plotms(matrix)
dev.off()

## End(Not run)
```

plotmz *plot GC/LC-MS data as scatter plot*

Description

plot GC/LC-MS data as scatter plot

Usage

```
plotmz(data, inscf = 3.5, ...)
```

Arguments

data	imported data matrix of GC-MS
inscf	Log intensity cutoff for peaks, default 3.5
...	parameters for ‘plot‘ function

Value

scatter plot

Examples

```
## Not run:
data(matrix)
png('test.png')
plotmz(matrix)
dev.off()

## End(Not run)
```

`plotpca` *plot the PCA for multiple samples*

Description

plot the PCA for multiple samples

Usage

```
plotpca(
  data,
  lv = NULL,
  index = NULL,
  center = TRUE,
  scale = TRUE,
  xrange = NULL,
  yrange = NULL,
  pch = NULL,
  ...
)
```

Arguments

`data`	data row as peaks and column as samples
`lv`	group information
`index`	index for selected peaks
`center`	parameters for PCA
`scale`	parameters for scale
`xrange`	x axis range for return samples, default NULL
`yrange`	y axis range for return samples, default NULL
`pch`	default pch would be the first character of group information or samples name
`...`	other parameters for ‘plot‘ function

Value

if xrange and yrange are not NULL, return file name of all selected samples on 2D score plot

Examples

```
data(list)
plotpca(list$data, lv = as.character(list$group$sample_group))
```

`plotpeak` *plot intensity of peaks across samples or samples across peaks*

Description

plot intensity of peaks across samples or samples across peaks

Usage

```
plotpeak(data, lv = NULL, indexx = NULL, indexy = NULL, ...)
```

Arguments

`data`	matrix
`lv`	factor vector for the column
`indexx`	index for matrix row
`indexy`	index for matrix column
`...`	parameters for ‘title‘ function

Value

parallel coordinates plot

Examples

```
data(list)
# selected peaks across samples
plotpeak(t(list$data), lv = as.factor(c(rep(1,5),rep(2,nrow(list$data)-5))),1:10,1:10)
# selected samples across peaks
plotpeak(list$data, lv = as.factor(list$group$sample_group),1:10,1:10)
```

plotridge *plot ridgeline density plot*

Description

plot ridgeline density plot

Usage

```
plotridge(data, lv = NULL, indexx = NULL, indexy = NULL, ...)
```

Arguments

data	matrix
lv	factor vector for the column
indexx	index for matrix row
indexy	index for matrix column
...	parameters for ‘title‘ function

Value

ridgeline density plot

Examples

```
data(list)
plotridge(t(list$data),indexy=c(1:10),xlab = 'Intensity',ylab = 'peaks')
plotridge(log(list$data),as.factor(list$group$sample_group),xlab = 'Intensity',ylab = 'peaks')
```

plotridges *Relative Log Abundance Ridge (RLAR) plots for samples or peaks*

Description

Relative Log Abundance Ridge (RLAR) plots for samples or peaks

Usage

```
plotridges(data, lv, type = "g")
```

Arguments

data	data row as peaks and column as samples
lv	factor vector for the group information of samples
type	’g’ means group median based, other means all samples median based.

Value

Relative Log Abundance Ridge(RLA) plots

Examples

```
data(list)
plotridges(list$data, as.factor(list$group$sample_group))
```

`plotrla` *Relative Log Abundance (RLA) plots*

Description

Relative Log Abundance (RLA) plots

Usage

```
plotrla(data, lv, type = "g", ...)
```

Arguments

`data`	data row as peaks and column as samples
`lv`	factor vector for the group information
`type`	'g' means group median based, other means all samples median based.
`...`	parameters for boxplot

Value

Relative Log Abundance (RLA) plots

Examples

```
data(list)
plotrla(list$data, as.factor(list$group$sample_group))
```

plotrsd *plot the rsd influences of data in different groups*

Description

plot the rsd influences of data in different groups

Usage

```
plotrsd(list, ms = c(100, 800), inscf = 5, rsdcf = 100, imputation = "l", ...)
```

Arguments

list	list with data as peaks list, mz, rt and group information
ms	the mass range to plot the data
inscf	Log intensity cutoff for peaks across samples. If any peaks show a intensity higher than the cutoff in any samples, this peaks would not be filtered. default 5
rsdcf	the rsd cutoff of all peaks in all group
imputation	parameters for 'getimputation' function method
...	other parameters for 'plot' function

Examples

```
data(list)
plotrsd(list)
```

plotrug *plot 1-d density for multiple samples*

Description

plot 1-d density for multiple samples

Usage

```
plotrug(data, lv = NULL, indexx = NULL, indexy = NULL, ...)
```

Arguments

data	matrix
lv	factor vector for the column
indexx	index for matrix row
indexy	index for matrix column
...	parameters for 'title' function

Examples

```
data(list)
plotrug(list$data)
plotrug(log(list$data), lv = as.factor(list$group$sample_group))
```

`plotsms` *Plot the intensity distribution of GC-MS*

Description

Plot the intensity distribution of GC-MS

Usage

```
plotsms(meanmatrix, rsdmatrix)
```

Arguments

`meanmatrix`	mean data matrix of GC-MS(n=5)
`rsdmatrix`	standard deviation matrix of GC-MS(n=5)

Examples

```
## Not run:
plotsms(meanmatrix,rsdmatrix)

## End(Not run)
```

`plotsub` *Plot the background of data*

Description

Plot the background of data

Usage

```
plotsub(data)
```

Arguments

`data`	imported data matrix of GC-MS

Examples

```
## Not run:
plotsub(matrix)

## End(Not run)
```

plott *plot GC-MS data as a heatmap for constant speed of temperature rising*

Description

plot GC-MS data as a heatmap for constant speed of temperature rising

Usage

```
plott(data, log = FALSE, temp = c(100, 320))
```

Arguments

data	imported data matrix of GC-MS
log	transform the intensity into log based 10
temp	temperature range for constant speed

Value

heatmap

Examples

```
## Not run:
plott(matrix)

## End(Not run)
```

plottic *Plot Total Ion Chromatogram (TIC)*

Description

Plot Total Ion Chromatogram (TIC)

Usage

```
plottic(data, n = FALSE)
```

Arguments

data	imported data matrix of GC-MS
n	logical smooth or not

Value

plot

Examples

```
## Not run:
plottic(matrix)

## End(Not run)
```

`plotTopMS1Peaks` *Extract top MS1 ions from MS2 EIC interactively*

Description

A Shiny app that displays extracted ion chromatograms from MS2 data. Click on a peak to select a retention time window, then extract the most intense MS1 ions from that window and overlay their chromatograms.

Usage

```
plotTopMS1Peaks(
  filepath,
  featlist,
  numTopIons = 10,
  diff = 0.01,
  rtWindow = 0.3
)
```

Arguments

`filepath`	character. Path to an mzML or mzXML file.
`featlist`	data.frame with columns: `name`, `mz`, `ms_level`.
`numTopIons`	integer. Number of most intense MS1 ions to extract. Default 10.
`diff`	numeric. Half-width of m/z extraction window (Da). Default 0.01.
`rtWindow`	numeric. Half-width of RT window (seconds) around clicked peak. Default 0.3.

Value

Opens an interactive Shiny app in the browser.

Examples

```
## Not run:
frags <- data.frame(
  name = c("Br79", "Br81"),
  mz = c(78.9183, 80.9163),
  ms_level = c("ms2", "ms2")
)
plotTopMS1Peaks("sample.mzML", frags, numTopIons = 3)

## End(Not run)
```

`plotTopMS2Peaks` *Extract top MS2 ions from MS1 EIC interactively*

Description

A Shiny app that displays extracted ion chromatograms from MS1 data. Click on a peak to select a retention time window, then extract the most intense MS2 ions from that window and overlay their chromatograms.

Usage

```
plotTopMS2Peaks(
  filepath,
  featlist,
  numTopIons = 10,
  diff = 0.01,
  rtWindow = 0.3
)
```

Arguments

`filepath`	character. Path to an mzML or mzXML file.
`featlist`	data.frame with columns: `name`, `mz`, `ms_level`.
`numTopIons`	integer. Number of most intense MS1 ions to extract. Default 10.
`diff`	numeric. Half-width of m/z extraction window (Da). Default 0.01.
`rtWindow`	numeric. Half-width of RT window (seconds) around clicked peak. Default 0.3.

Value

Opens an interactive Shiny app in the browser.

Examples

```
## Not run:
targets <- data.frame(
  name = c("precursor1"),
  mz = c(500.1234),
  ms_level = c("ms1")
)
plotTopMS2Peaks("sample.mzML", targets, numTopIons = 5)

## End(Not run)
```

`runMDPlot` *Shiny application for interactive mass defect plots analysis*

Description

Shiny application for interactive mass defect plots analysis

Usage

```
runMDPlot()
```

`runsccp` *Shiny application for Short-Chain Chlorinated Paraffins analysis*

Description

Shiny application for Short-Chain Chlorinated Paraffins analysis

Usage

```
runsccp()
```

sccp — *Short-Chain Chlorinated Paraffins(SCCPs) peaks information for quantitative analysis*

Description

A dataset containing the ions, formula, Cl

Usage

```
data(sccp)
```

Format

A data frame with 24 rows and 8 variables:

Cln Chlorine atom numbers

Cn Carbon atom numbers

formula molecular formula

Hn hydrogen atom numbers

ions [M-Cl]- ions

mz m/z for the isotopologues with highest intensity

intensity abundance of the isotopologues with highest intensity

Clp Chlorine contents

scoreIsotopes — *Calculate Isotopic Pattern Similarity Scores*

Description

Computes modern spectral similarity metrics including Cosine similarity, intensity-weighted dot product, and log-likelihood between observed mass spectral peaks and theoretical isotopic distributions.

Usage

```
scoreIsotopes(
  obs_mz,
  obs_int,
  theo_mz,
  theo_int,
  tolerance = 0.005,
  ppm = FALSE
)
```

Arguments

obs_mz	Numeric vector of observed m/z values.
obs_int	Numeric vector of observed intensities.
theo_mz	Numeric vector of theoretical isotopic m/z values.
theo_int	Numeric vector of theoretical isotopic abundances.
tolerance	Numeric mass tolerance for peak alignment (default: 0.005).
ppm	Logical indicating whether tolerance is in ppm (default: FALSE).

Value

A list containing:

cosine	Cosine similarity (unweighted dot product of normalized spectra).
weighted_cosine	Mass- and intensity-weighted dot product (MassBank/NIST style).
log_likelihood	Multinomial log-likelihood of observed peak distribution.
matched_peaks	Integer count of successfully matched peaks.

Examples

```
obs_m <- c(180.0634, 181.0668)
obs_i <- c(100, 6.5)
theo_m <- c(180.0634, 181.0668)
theo_i <- c(1.0, 0.066)
scoreIsotopes(obs_m, obs_i, theo_m, theo_i)
```

TBBPA *Demo data for TBBPA metabolism in Pumpkin*

Description

Demo data for TBBPA metabolism in Pumpkin

Usage

```
data(TBBPA)
```

Format

A list object with data, mass to charge ratio, retention time and group information. Three pumpkin seeding root samples' peaks list is extracted by xcms online.

References

Hou, X., Yu, M., Liu, A., Wang, X., Li, Y., Liu, J., Schnoor, J.L., Jiang, G., 2019. Glycosylation of Tetrabromobisphenol A in Pumpkin. Environ. Sci. Technol. https://doi.org/10.1021/acs.est.9b02122

writeMSP *Write MSP file for NIST search*

Description

Write MSP file for NIST search

Usage

```
writeMSP(list, name = "unknown", sep = FALSE)
```

Arguments

list	a list with spectra information
name	name of the compounds
sep	numeric or logical the numbers of spectra in each file and FALSE to include all of the spectra in one msp file

Value

none a MSP file will be created.

Examples

```
## Not run:
intensity <- c(10000,20000,10000,30000,5000)
mz <- c(101,143,189,221,234)
writeMSP(list(list(spectra = cbind.data.frame(mz,intensity))), name = 'test')

## End(Not run)
```

Index

∗ **datasets**
    list, 31
    matrix, 32
    sccp, 51
    TBBPA, 52

c.mzrt, 3
checkGoldenRules, 4
cleanMGF, 5

findline, 6
findlipid, 7
findmet, 7
findohc, 8
findpfc, 9

getalign, 10
getalign2, 10
getcompare, 11
getcsv, 12, *15*
getdoe, 13, *18*, *24, 25*
getdwtus, 14
getfilter, 14
getformula, 15
getHRMF, 16, *30*
getimputation, *13*, *15*, 17, *24, 25*
getintegration, 18
getisotopologues, 19
getmass, 20
getmassdefect, 20, *38*
getmdh, 21
getmdr, 21
getms1anno, 22
getMSP, *6*, *17*, 23, *23*, *29, 30*
getmzrtcsv, 23
getoverlappeak, 24
getpn, 24
getpower, *13*, 25
getpqsi, 26
getrangecsv, 26
getretcor, 27
getrmd, 27
gifmr, 28

HRMF, *5, 6*, *16, 17*, 29

integration, 31

list, 31

ma, 32
matrix, 32

plotcc, 33
plotden, 33
plotdwtus, 34
plotEIC, 34
plotgroup, 35
plothist, 36
plothm, 36
plotint, 37
plotintslope, 37
plotkms, *20*, 38
plotmr, 38
plotmrc, 39
plotms, 40
plotmz, 40
plotpca, 41
plotpeak, 42
plotridge, 43
plotridges, 43
plotrla, 44
plotrsd, 45
plotrug, 45
plotsms, 46
plotsub, 46
plott, 47
plottic, 47
plotTopMS1Peaks, 48
plotTopMS2Peaks, 49

runMDPlot, 50
runsccp, 50

sccp, 51
scoreIsotopes, 51

TBBPA, 52

writeMSP, 53