Package ‘enviGCMS’

February 3, 2020

Type Package
Title GC/LC-MS Data Analysis for Environmental Science
Version 0.6.0
Date 2020-02-03
Maintainer Miao YU <yufreecas@gmail.com>
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://github.com/yufree/enviGCMS
License GPL-2
Encoding UTF-8
LazyData true
Suggests knitr, testthat, xcms, MSnbase
VignetteBuilder knitr

biocViews
Depends R (>= 2.10)
Imports Rdisop, RColorBrewer, mixtools, BiocParallel, genefilter, grDevices, graphics, stats, utils, methods, reshape2, animation (>= 2.2.3), data.table, rmarkdown, shiny, shinythemes, DT, crosstalk, dplyr, plotly, broom, igraph, ggraph, ggplot2, ggridges
RoxygenNote 7.0.2
NeedsCompilation no
Author Miao YU [aut, cre] (<https://orcid.org/0000-0002-2804-6014>), Thanh Wang [ctb] (<https://orcid.org/0000-0002-5729-1908>)
Repository CRAN
Date/Publication 2020-02-03 22:30:02 UTC
## R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch</td>
<td>4</td>
</tr>
<tr>
<td>cbmd</td>
<td>4</td>
</tr>
<tr>
<td>findline</td>
<td>5</td>
</tr>
<tr>
<td>findmet</td>
<td>6</td>
</tr>
<tr>
<td>findohc</td>
<td>6</td>
</tr>
<tr>
<td>getarea</td>
<td>7</td>
</tr>
<tr>
<td>getareastd</td>
<td>8</td>
</tr>
<tr>
<td>getbgremove</td>
<td>9</td>
</tr>
<tr>
<td>getbiotechrep</td>
<td>10</td>
</tr>
<tr>
<td>getcsv</td>
<td>10</td>
</tr>
<tr>
<td>getdata</td>
<td>11</td>
</tr>
<tr>
<td>getdata2</td>
<td>13</td>
</tr>
<tr>
<td>getdoe</td>
<td>14</td>
</tr>
<tr>
<td>getdwus</td>
<td>15</td>
</tr>
<tr>
<td>getfeaturesanova</td>
<td>15</td>
</tr>
<tr>
<td>getfeaturest</td>
<td>16</td>
</tr>
<tr>
<td>getfilter</td>
<td>17</td>
</tr>
<tr>
<td>getformula</td>
<td>18</td>
</tr>
<tr>
<td>getgrouprep</td>
<td>18</td>
</tr>
<tr>
<td>getimputation</td>
<td>19</td>
</tr>
<tr>
<td>GetIntegration</td>
<td>20</td>
</tr>
<tr>
<td>Getisotopologues</td>
<td>21</td>
</tr>
<tr>
<td>getmass</td>
<td>21</td>
</tr>
<tr>
<td>getmassdefect</td>
<td>22</td>
</tr>
<tr>
<td>getmd</td>
<td>23</td>
</tr>
<tr>
<td>getmdh</td>
<td>23</td>
</tr>
<tr>
<td>getmdr</td>
<td>24</td>
</tr>
<tr>
<td>getmr</td>
<td>25</td>
</tr>
<tr>
<td>getmzrt</td>
<td>26</td>
</tr>
<tr>
<td>getmzrt2</td>
<td>27</td>
</tr>
<tr>
<td>getmzrtcsv</td>
<td>28</td>
</tr>
<tr>
<td>getoverlapmass</td>
<td>28</td>
</tr>
<tr>
<td>getoverlappeak</td>
<td>29</td>
</tr>
<tr>
<td>getoverlaprt</td>
<td>30</td>
</tr>
<tr>
<td>getpower</td>
<td>30</td>
</tr>
<tr>
<td>getpqsi</td>
<td>31</td>
</tr>
<tr>
<td>getQCraw</td>
<td>31</td>
</tr>
<tr>
<td>getrmd</td>
<td>32</td>
</tr>
<tr>
<td>getsccp</td>
<td>33</td>
</tr>
<tr>
<td>getsim</td>
<td>34</td>
</tr>
<tr>
<td>gettechrep</td>
<td>34</td>
</tr>
<tr>
<td>gettimegroupprep</td>
<td>35</td>
</tr>
<tr>
<td>getupload</td>
<td>36</td>
</tr>
<tr>
<td>getupload2</td>
<td>37</td>
</tr>
<tr>
<td>getupload3</td>
<td>37</td>
</tr>
<tr>
<td>gifmr</td>
<td>38</td>
</tr>
</tbody>
</table>
Integration ....................................................... 39
list .............................................................. 40
ma ............................................................. 40
Mode ............................................................. 41
plotcc ........................................................... 41
plotden ......................................................... 42
plotdwtus ...................................................... 42
plote ............................................................ 43
plotgroup ........................................................ 44
plothist ........................................................ 44
plothm .......................................................... 45
plotint .......................................................... 45
plotintslope .................................................... 46
plotkms ........................................................ 46
plotmr ........................................................... 47
plotmrc ........................................................ 48
plotms ........................................................... 48
plotmsrt ......................................................... 49
plotmz ........................................................... 50
plotpca .......................................................... 51
plotridges ...................................................... 52
plotrla .......................................................... 52
plotrsd .......................................................... 53
plotrtms ........................................................ 53
plotsms ........................................................ 54
plotsub ........................................................ 55
plot ............................................................. 55
plottic ........................................................... 56
qbatch ........................................................... 56
runMDPlot ....................................................... 57
runseccp ......................................................... 57
seccp ............................................................ 58
submd ............................................................ 58
svabatch ........................................................ 59
svacor ............................................................ 60
svadata .......................................................... 61
svapca ........................................................... 62
svaplot ........................................................... 63
svaupload ....................................................... 64
TBBPA ............................................................ 64
writeMSP ........................................................ 65

Index ............................................................ 66
**batch**  
*Get the MIR and related information from the files*

**Description**
Get the MIR and related information from the files

**Usage**

```r
batch(file, mz1, mz2)
```

**Arguments**

- `file`: data file, CDF or other format supported by xcmsRaw
- `mz1`: the lowest mass
- `mz2`: the highest mass

**Value**

Molecular isotope ratio

**Examples**

```r
## Not run:
mr <- batch(data,mz1 = 79, mz2 = 81)
## End(Not run)
```

---

**cbmd**  
*Combine two data with similar retention time while different mass range*

**Description**

Combine two data with similar retention time while different mass range

**Usage**

```r
cbmd(data1, data2, mzstep = 0.1, rtstep = 0.01)
```

**Arguments**

- `data1`: data file path of lower mass range
- `data2`: data file path of higher mass range
- `mzstep`: the m/z step for generating matrix data from raw mass spectral data
- `rtstep`: the alignment accuracy of retention time, e.g. 0.01 means the retention times of combined data should be the same at the accuracy 0.01s. Higher rtstep would return less scans for combined data
findline

Value

matrix with the row as scantime in second and column as m/z

Examples

## Not run:
# mz100_200 and mz201_300 were the path to the raw data
matrix <- getmd(mz100_200,mz201_300)

## 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 <- getmd(rawdata)
findline(data)

## End(Not run)
**findmet**  
*Screen metabolites by Mass Defect*

**Description**

Screen metabolites by Mass Defect

**Usage**

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

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>list</td>
<td>list with data as peaks list, mz, rt and group information, retention time should be in seconds</td>
</tr>
<tr>
<td>mass</td>
<td>mass to charge ratio of specific compounds</td>
</tr>
<tr>
<td>mdr</td>
<td>mass defect range, default 50mDa</td>
</tr>
</tbody>
</table>

**Value**

list with filtered metabolites mass to charge index of certain compound

---

**findohc**  
*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.*

**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**

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

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


---

getarea

Get the peak information from samples for SCCPs detection

Description

Get the peak information from samples for SCCPs detection

Usage

getarea(data, ismz = 323, ppm = 5, rt = NULL, rts = NULL)

Arguments

- **data**: list from `xcmsRaw` function
- **ismz**: internal standards m/z
- **ppm**: resolution of mass spectrum
- **rt**: retention time range of SCCPs
- **rts**: retention time range of internal standards
Value

list with peak information

See Also

getareastd, getsccp

gotareastd

Get the peak information from SCCPs standards

Description

Get the peak information from SCCPs standards

Usage

getareastd(data = NULL, ismz = 323, ppm = 5, con = 2000, rt = NULL, rts = NULL)

Arguments

data list from 'xcmsRaw' function
ismz internal standards m/z
ppm resolution of mass spectrum
con concentration of standards
rt retention time range of sccps
rts retention time range of internal standards

Value

list with peak information

See Also

getarea, getsccp
Get the peak list with blank samples' peaks removed

**Description**

Get the peak list with blank samples' peaks removed

**Usage**

```r
getbgremove(
  xset,
  method = "medret",
  intensity = "into",
  file = NULL,
  rsdcf = 30,
  inscf = 1000
)
```

**Arguments**

- `xset`: the xcmsset object with blank and certain group samples' data
- `method`: parameter for groupval function
- `intensity`: parameter for groupval function
- `file`: file name for further annotation, default NULL
- `rsdcf`: rsd cutoff for peaks, default 30
- `inscf`: intensity cutoff for peaks, default 1000

**Value**

diff report

**Examples**

```r
## Not run:
library(faahKO)
cdfpath <- system.file("cdf", package = "faahKO")
xset <- getdata(cdfpath, pmethod = '
')
getbgremove(xset)

## End(Not run)
```
getbiotechrep  

Get the report for biological replicates.

Description
Get the report for biological replicates.

Usage
getbiotechrep(
    xset,
    method = "medret",
    intensity = "into",
    file = NULL,
    rsdcf = 30,
    inscf = 1000
)

Arguments
- **xset**: the xcmsset object which for all of your technique replicates for bio replicated sample in single group
- **method**: parameter for groupval function
- **intensity**: parameter for groupval function
- **file**: file name for further annotation, default NULL
- **rsdcf**: rsd cutoff for peaks, default 30
- **inscf**: intensity cutoff for peaks, default 0

Value
dataframe with mean, standard deviation and RSD for those technique replicates & biological replicates combined with raw data

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", ...)
getdata

Get xcmsset object in one step with optimized methods.

Description

Get xcmsset object in one step with optimized methods.

Usage

getdata(
  path,
  index = F,
  BPPARAM = BiocParallel::SnowParam(),
  pmethod = "hplcorbitrap",
  minfrac = 0.67,
  ...
)
getdata

Arguments

- **path** the path to your data
- **index** the index of the files
- **BPPARAM** used for BiocParallel package
- **pmethod** parameters used for different instrumentals such as 'hplcorbitrap', 'uplcorbitrap', 'hplcqtof', 'hplchqtof', 'uplcqtof', 'uplchqtof'. The parameters were from the reference
- **minfrac** minimum fraction of samples necessary in at least one of the sample groups for it to be a valid group, default 0.67
- ... arguments for xcmsSet function

Details

the parameters are extracted from the papers. If you use name other than the name above, you will use the default setting of XCMS. Also I suggest IPO packages or apLCMS packages to get reasonable data for your own instrumental. If you want to summit the results to a paper, remember to include those parameters.

Value

a xcmsset object for that path or selected samples

References


See Also

gedata2, getmzrt

Examples

```r
# Not run:
library(faahKO)
cdfpath <- system.file('cdf', package = 'faahKO')
xset <- getdata(cdfpath, pmethod = '')

# End(Not run)
```
getdata2

Get XCMSnExp object in one step from structured folder path for xcms 3.

Description

Get XCMSnExp object in one step from structured folder path for xcms 3.

Usage

```
getdata2(
  path,
  index = F,
  snames = NULL,
  sclass = NULL,
  phenoData = NULL,
  BPPARAM = BiocParallel::SnowParam(),
  mode = "onDisk",
  ppp = xcms::CentWaveParam(ppm = 5, peakwidth = c(5, 25), prefilter = c(3, 5000)),
  rtp = xcms::ObiwarpParam(binSize = 1),
  gpp = xcms::PeakDensityParam(sampleGroups = 1, minFraction = 0.67, bw = 2, binSize = 0.025),
  fpp = xcms::FillChromPeaksParam()
)
```

Arguments

- `path` the path to your data
- `index` the index of the files
- `snames` sample names. By default the file name without extension is used
- `sclass` sample classes.
- `phenoData` data.frame or NAnnotatedDataFrame defining the sample names and classes and other sample related properties. If not provided, the argument `sclass` or the subdirectories in which the samples are stored will be used to specify sample grouping.
- `BPPARAM` used for BiocParallel package
- `mode` 'inMemory' or 'onDisk' see `?MSnbase::readMSData` for details, default 'onDisk'
- `ppp` parameters for peaks picking, e.g. xcms::CentWaveParam()
- `rtp` parameters for retention time correction, e.g. xcms::ObiwarpParam()
- `gpp` parameters for peaks grouping, e.g. xcms::PeakDensityParam()
- `fpp` parameters for peaks filling, e.g. xcms::FillChromPeaksParam(), PeakGroupsParam()

Details

This is a wrap function for metabolomics data process for xcms 3.
getdoe

**Value**

a XCMSnExp object with processed data

**See Also**

gedata, getmzrt

---

getdoe  
*Filter the data based on DoE, rsd, intensity*

**Description**

Filter the data based on DoE, rsd, intensity

**Usage**

getdoe(
  list,
  inscf = 5,
  rsdcf = 100,
  rsdcft = 30,
  imputation = "l",
  tr = F,
  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**

gedata2, getdata, getmzrt, getimputation, getmr, getpower
getdwtus

Examples

data(list)
getdoe(list)

density weighted intensity for one sample

Description

Density weighted intensity for one sample

Usage

getdwtus(peak, n = 512, log = F)

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])

getfeaturesanova  
Get the features from anova, with p value, q value, rsd and power restriction

Description

Get the features from anova, with p value, q value, rsd and power restriction
getfeaturesanova(list, power = 0.8, pt = 0.05, qt = 0.05, n = 3, ng = 3, rsdcf = 100, inscf = 5, imputation = "l", index = NULL)

Arguments

list
list with data as peaks list, mz, rt and group information (more than two groups)

power
defined power

pt
p value threshold

qt
q value threshold, BH adjust

n
c sample numbers in one group

ng
group numbers

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

index
the index of peaks considered, default NULL

Value
dataframe with peaks fit the setting above

getfeaturesrest
Get the features from t test, with p value, q value, rsd and power restriction

Description
Get the features from t test, with p value, q value, rsd and power restriction

Usage
getfeaturesrest(list, power = 0.8, pt = 0.05, qt = 0.05, n = 3, imputation = "l")
getfilter

Arguments

- **list**: list with data as peaks list, mz, rt and group information (two groups)
- **power**: defined power
- **pt**: p value threshold
- **qt**: q value threshold, BH adjust
- **n**: sample numbers in one group
- **imputation**: parameters for ‘getimputation’ function method

Value

dataframe with peaks fit the setting above

---

getfilter

*Filter the data based on row and column index*

Description

Filter the data based on row and column index

Usage

getfilter(list, rowindex = T, colindex = T, 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

getdata2, getdata, getmzrt, getimputation, getmr, getcsv
getgrouprep

Get the report for samples with biological and technique replicates in different groups

Description
Get the report for samples with biological and technique replicates in different groups

Examples

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

getformula

Get chemical formula for mass to charge ratio.

Description
Get chemical formula for mass to charge ratio.

Usage

```r
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)))
```

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.

Value

list with chemical formula
Usage

```r
getgrouprep(
  xset,
  file = NULL,
  method = "medret",
  intensity = "into",
  rsdcf = 30,
  inscf = 1000
)
```

Arguments

- `xset`: the xcmsset object all of samples with technique replicates
- `file`: file name for the peaklist to MetaboAnalyst
- `method`: parameter for groupval function
- `intensity`: parameter for groupval function
- `rsdcf`: rsd cutoff for peaks, default 30
- `inscf`: intensity cutoff for peaks, default 1000

Value

dataframe with mean, standard deviation and RSD for those technique replicates & biological replicates combined with raw data in different groups if file are defaults NULL.

---

**getimputation**

*Impute the peaks list data*

Description

Impute the peaks list data

Usage

```r
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

getdata2, getdata, getmzrt, getdoe, getmr

Examples

data(list)
getimputation(list)

GetIntegration

GetIntegration was mainly used for get the intergration of certain ion's chromatogram data and plot the data

Description

GetIntegration was mainly used for get the intergration 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 = T, 
  smoothit = T, 
  half = F
)

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 caculate the area
Getisotopologues

Value

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

Examples

```r
## 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 = "C12OH6Br4", charge = 1, width = 0.3)

Arguments

- `formula`: the molecular formula. C12OH6Br4 means BDE-47 as default
- `charge`: the charge of that molecule. 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.

Examples

```
# show isotopologues for BDE-47
Getisotopologues(formula = 'C12OH6Br4')
```

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)
getmassdefect

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

Value
 numerical vector

Examples
 getmass('CH2')

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)
**getmd**

Import data and return the annotated matrix for GC/LC-MS by m/z range and retention time

**Description**

Import data and return the annotated matrix for GC/LC-MS by m/z range and retention time

**Usage**

```r
getmd(data, mzstep = 0.1, mzrange = F, rrange = F)
```

**Arguments**

- `data` file type which xcmsRaw could handle
- `mzstep` the m/z step for generating matrix data from raw mass spectral data
- `mzrange` vector range of the m/z, default all
- `rrange` vector range of the retention time, default all

**Value**

matrix with the row as increasing m/z second and column as increasing scantime

**Examples**

```r
## Not run:
library(faahKO)
cdfpath <- system.file('cdf', package = 'faahKO')
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
matrix <- getmd(cdffiles[1])
## End(Not run)
```

---

**getmdh**

Get the high order unit based Mass Defect

**Description**

Get the high order unit based Mass Defect

**Usage**

```r
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

```r
getmdh(getmass('C2H4'))
```

---

**getmdr**

*Get the raw Mass Defect*

Description

Get the raw Mass Defect

Usage

```r
getmdr(mz)
```

Arguments

- **mz**: numeric vector for exact mass

Value

raw Mass Defect

Examples

```r
getmdr(getmass('C2H4'))
```
getmr

Get the mzrt profile and group information for batch correction and plot as a list directly from path with default setting

Description

Get the mzrt profile and group information for batch correction and plot as a list directly from path with default setting

Usage

getmr(
  path,
  index = F,
  BPPARAM = BiocParallel::SnowParam(),
  pmethod = "hplcorbitrap",
  minfrac = 0.67,
  ...
)

Arguments

- `path`: the path to your data
- `index`: the index of the files
- `BPPARAM`: used for BiocParallel package
- `pmethod`: parameters used for different instrumentals such as 'hplcorbitrap', 'uplcorbitrap', 'hplcqtof', 'hplchqtof', 'uplcqtof', 'uplchqtof'. The parameters were from the references
- `minfrac`: minimum fraction of samples necessary in at least one of the sample groups for it to be a valid group, default 0.67
- `...`: arguments for xcmsSet function

Value

list with rtmz profile and group information

See Also

`getdata, getupload, getmzrt, getdoe`

Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file(’cdf’, package = ’faahKO’)
list <- getmr(cdfpath, pmethod = ’ ’)

## End(Not run)
```
getmzrt

Get the mzrt profile and group information as a mzrt list and/or save them as csv or rds for further analysis.

Description

Get the mzrt profile and group information as a mzrt list and/or save them as csv or rds for further analysis.

Usage

getmzrt(
  xset,
  name = NULL,
  mzdigit = 4,
  rtgdigit = 1,
  method = "medret",
  value = "into",
  eic = F,
  type = "o"
)

Arguments

  xset  xcmsSet/XCMSnExp objects
  name  file name for csv and/or eic file, default NULL
  mzdigit  m/z digits of row names of data frame, default 4
  rtgdigit  retention time digits of row names of data frame, default 1
  method  parameter for groupval or featureDefinitions function, default medret
  value  parameter for groupval or featureDefinitions function, default into
  eic  logical, save xcmsSet and xcmsEIC objects for further investigation with the same name of files, you will need raw files in the same directory as defined in xcmsSet to extract the EIC based on the binned data. You could use ‘plot’ to plot EIC for specific peaks. For example, ‘plot(xcmsEIC, xcmsSet, groupidx = 'M123.4567T278.9')’ could show the EIC for certain peaks with m/z 206 and retention time 2789. default F
  type  csv formate for furthor 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 vlaue), o means full infomation csv (for ‘pmd’ package), default o. mapo could output all those format files.

Value

  mzrt object, a list with mzrt profile and group infomation
References


See Also

getdata, getdata2, getdoe, getcsv, getfilter

Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file("/quotesingle.Var cdf /quotesingle.Var", package = "faahKO")
xset <- getdata(cdfpath, pmethod = ")
getmzrt(xset, name = 'demo', type = 'mapo')
## End(Not run)
```

---

**getmzrt2**

*Get the mzrt profile and group information for batch correction and plot as a list for xcms 3 object*

**Description**

Get the mzrt profile and group information for batch correction and plot as a list for xcms 3 object

**Usage**

`getmzrt2(xset, name = NULL)`

**Arguments**

- `xset`: a XCMSnExp object with processed data
- `name`: file name for csv file, default NULL

**Value**

list with rtmz profile and group information

**See Also**

getdata2, getupload2, getmzrt, getdoe, getmzrtcsv
getoverlapmass

## Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file('cdf', package = 'faahKO')
xset <- getdata2(cdfpath,
    ppp = xcms::MatchedFilterParam(),
    rtp = xcms::Obi warpParam(),
    gpp = xcms::PeakDensityParam())
getmzrt2(xset)

## End(Not run)
```

getmzrtcsv

### Description

Covert the peaks list csv file into list

### Usage

```r
getmzrtcsv(path)
```

### Arguments

- **path**  
  the path to your csv file

### Value

list with rtmz profile and group information as the first row

### See Also

[getmzrt](#)

getoverlapmass

### Description

Get the overlap peaks by mass range

### Usage

```r
getoverlapmass(mzrange1, mzrange2)
```
getoverlappeak

Arguments

mzrange1  mass range 1 to be overlapped
mzrange2  mass range 2 to overlap

Value

logical index for mzrange1’s peaks

See Also

getmzrt, getimputation, getmr, getdoe, getoverlappeak, getoverlaprt

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

getmzrt, getimputation, getmr, getdoe, getoverlapmass, getoverlaprt
getoverlaprt  
*Get the overlap peaks by retention time*

**Description**
Get the overlap peaks by retention time

**Usage**
getoverlaprt(rtrange1, rtrange2)

**Arguments**
rtrange1  mass range 1 to be overlapped  
rtrange2  mass range 2 to overlap

**Value**
logical index for rtrange1’s peaks

**See Also**
getmzrt, getimputation, getmr, getdoe, getoverlapmass, getoverlappeak

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

gedata2, getdata, getmzrt, getimputation, getmr, 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.

getQCraw

get the data of QC compound for a group of data

Description

get the data of QC compound for a group of data

Usage

getQCraw(path, mzrange, rrange, index = NULL)
Arguments

- **path**: data path for your QC samples
- **mzrange**: mass of the QC compound
- **rtrange**: retention time of the QC compound
- **index**: index of the files contained QC compounds, default is all of the compounds

Value

- number vector, each number indicate the peak area of that mass and retention time range

---

**getrmd**

*Get the Relative Mass Defect*

Description

Get the Relative Mass Defect

Usage

```r
getrmd(mz)
```

Arguments

- **mz**: numeric vector for exact mass

Value

Relative Mass Defect

Examples

```r
getrmd(getmass('C2H4'))
```
Quantitative analysis for short-chain chlorinated paraffins (SCCPs)

**Description**

Quantitative analysis for short-chain chlorinated paraffins (SCCPs)

**Usage**

```r
getsccp(  
  pathstds,  
  pathsample,  
  ismz = 323,  
  ppm = 5,  
  con = 2000,  
  rt = NULL,  
  rts = NULL,  
  log = T  
)
```

**Arguments**

- `pathstds` : mzxml file path for SCCPs standards
- `pathsample` : mzxml file path for samples
- `ismz` : internal standards m/z
- `ppm` : resolution of mass spectrum
- `con` : concentration of standards
- `rt` : retention time range of sccps
- `rts` : retention time range of internal standards
- `log` : log transformation for response factor

**Value**

- list with peak information

**See Also**

- `getareastd`, `getarea`
getsim

output the similarity of two dataset

Description

output the similarity of two dataset

Usage

getsim(xset1, xset2)

Arguments

xset1 the first dataset
xset2 the second dataset

Value

similarity on retention time and rsd

gtechrep

Get the report for technique replicates.

Description

Get the report for technique replicates.

Usage

gtechrep(
  xset,
  method = "medret",
  intensity = "into",
  file = NULL,
  rsdcf = 30,
  inscf = 1000
)

Arguments

xset the xcmsset object which for all of your technique replicates for one sample
method parameter for groupval function
intensity parameter for groupval function
file file name for further annotation, default NULL
rsdcf rsd cutoff for peaks, default 30
inacf intensity cutoff for peaks, default 1000
gettimegrouprep

Value

dataframe with mean, standard deviation and RSD for those technique replicates combined with raw data

Description

Get the time series or two factor DoE report for samples with biological and technique replicates in different groups

Usage

gettimegrouprep(
  xset,
  file = NULL,
  method = "medret",
  intensity = "into",
  rsdcf = 30,
  inscf = 1000
)

Arguments

xset the xcmsset object all of samples with technique replicates in time series or two factor DoE
file file name for the peaklist to MetaboAnalyst
method parameter for groupval function
intensity parameter for groupval function
rsdcf rsd cutoff for peaks, default 30
inscf intensity cutoff for peaks, default 1000

Value

dataframe with time series or two factor DoE mean, standard deviation and RSD for those technique replicates & biological replicates combined with raw data in different groups if file are defaults NULL.
getupload

Get the csv files from xcmsset/XCMSnExp/list object

Description
Get the csv files from xcmsset/XCMSnExp/list object

Usage

getupload(
  xset, method = "medret", value = "into", name = "Peaklist",
  type = "m", mzdigit = 4, rtdigit = 1
)

Arguments

  xset the xcmsset/XCMSnExp/list object which you want to submitted to Metaboanalyst
  method parameter for groupval function
  value parameter for groupval function
  name file name
  type m means Metaboanalyst, a means xMSannotator, o means full information csv
  mzdigit m/z digits of row names of data frame
  rtdigit retention time digits of row names of data frame

Value
dataframe with data needed for Metaboanalyst/xMSannotator/pmd if your want to perform local analysis.

See Also

getdata, getmzrt

Examples

## Not run:
library(faahKO)
cdfpath <- system.file("/quotesingle.Var/cdf/quotesingle.Var", package = "faahKO")
xset <- getdata(cdfpath, pmethod = "")
getupload(xset)

## End(Not run)
getupload2

Get the csv files to be submitted to Metaboanalyst

Description

Get the csv files to be submitted to Metaboanalyst

Usage

getupload2(xset, value = "into", name = "Peaklist")

Arguments

xset a XCMSnExp object with processed data which you want to submitted to Metaboanalyst
value value for 'xcms::featureValues'
name file name

Value

dataframe with data needed for Metaboanalyst if your want to perform local analysis.

See Also

getdata2, getupload, getmzrt2

Examples

## Not run:
library(faahKO)
cdfpath <- system.file('cdf', package = 'faahKO')
xset <- getdata2(cdfpath)
getupload2(xset)

## End(Not run)

getupload3

Get the csv files to be submitted to Metaboanalyst

Description

Get the csv files to be submitted to Metaboanalyst

Usage

getupload3(list, name = "Peaklist")
Arguments

- **list**: list with data as peaks list, mz, rt and group information
- **name**: file name

Value

dataframe with data needed for Metaboanalyst if you want to perform local analysis.

See Also

getmzrt, getmzrt2

Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file('cdf', package = 'faahKO')
xset <- getdata2(cdfpath,
    ppp = xcms::MatchedFilterParam(),
    rtp = xcms::ObiwapParam(),
    gpp = xcms::PeakDensityParam())
xset <- enviGCMS::getmzrt2(xset)
getupload3(xset)

## End(Not run)
```

---

### gifmr

**Description**

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

**Usage**

```r
gifmr(
    list,
    ms = c(100, 500),
    rsdcdf = 30,
    inscf = 5,
    imputation = "i",
    name = "test",
    ...
)
```
Integration

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)

Integration

Just intergrate data according to fixed rt and fixed noise area

Description

Just intergrate data according to fixed rt and fixed noise area

Usage

Integration(data, rt = c(8.3, 9), brt = c(8.3, 8.4), smoothit = T)

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 intergration data
Examples

```r
## Not run:
area <- Integration(data)

## End(Not run)
```

### list

#### Demo data

**Description**

Demo data

**Usage**

```r
data(list)
```

**Format**

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

#### ma

**filter data by average moving box**

**Description**

filter data by average moving box

**Usage**

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

**Arguments**

- `x` a vector
- `n` A number to indentify the size of the moving box.

**Value**

The filtered data

**Examples**

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

*define the Mode function*

**Description**

define the Mode function

**Usage**

```r
Mode(x)
```

**Arguments**

- `x` vector

**Value**

Mode of the vector

---

**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**

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

**Arguments**

- `x` concentration
- `y` response
- `upper` upper error bar
- `lower` lower error bar
- `...` parameters for ‘plot’ function

**Examples**

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

## End(Not run)
```
plotden

Description
plot the density for multiple samples

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

Arguments
- `data` mzrt profile with row peaks and column 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
```r
data(list)
plotden(list$data, lv = as.character(list$group), ylim = c(0,1))
```

plotdwtus

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
plote

Value

Density weighted intensity for multiple samples

Examples

data(list)
plotdwtus(list)

plote

plot EIC and boxplot for all peaks and return diffreport

Description

plot EIC and boxplot for all peaks and return diffreport

Usage

plote(xset, name = "test", test = "t", nonpara = "n", ...)

Arguments

xset xcmsset object
name filebase of the sub dir

test 't' means two-sample welch t-test, 't.equalvar' means two-sample welch t-test
with equal variance, 'wilcoxon' means rank sum wilcoxon test, 'f' means F-test,
'pairt' means paired t test, 'blockf' means Two-way analysis of variance, default
't'

nonpara 'y' means using nonparametric ranked data, 'n' means original data

... other parameters for 'diffreport'

Value

diffreport and pdf figure for EIC and boxplot

Examples

## Not run:
library(faahKO)
cdfpath <- system.file("/quotesingle.Var/cdf/quotesingle.Var", package = "faahKO")
xset <- getdata(cdfpath, pmethod = ' ')
plote(xset)

## 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**

```r
## Not run:
data <- getmd(rawdata)
plotgroup(data)
## 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

```r
## Not run:
matrix <- getmd(rawdata)
plothist(matrix)
## End(Not run)
```

#### plothm

**Plot the heatmap of mzrt profiles**

#### Description

Plot the heatmap of mzrt profiles

#### Usage

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

#### Arguments

- `data`: mzrt profile with row peaks and column samples
- `lv`: group information
- `index`: index for selected peaks

#### Examples

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

### plotint

**plot the information of intergretion**

#### Description

plot the information of intergretion

#### Usage

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

#### Arguments

- `list`: list from getinteragtion
- `name`: the title of the plot
Examples

```r
## Not run:
list <- getinteragtion(rawdata)
plotint(list)
## End(Not run)
```

---

plotintslope  
*plot the slope information of intergretion*

Description

plot the slope information of intergretion

Usage

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

Arguments

- `list`: list from getinteragtion
- `name`: the title of the plot

Examples

```r
## 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

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

Arguments

- `data`: vector with the name m/z
- `cutoff`: remove the low intensity
plotmr

plot the scatter plot for peaks list with threshold

Description

plot the scatter plot for peaks list with threshold

Usage

```r
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

See Also

getmassdefect

Examples

```r
## 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)
```
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 = T)

plotmrc

plot the diff scatter plot for one xcmsset objects with threshold between two groups

Description

plot the diff scatter plot for one xcmsset objects 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 = F)

Examples

data(list)
plotmrc(list)
Arguments

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

Value

- heatmap

Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file("cdf", package = 'faahKO')
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
matrix <- getmd(cdffiles[1])
png("test.png")
plotms(matrix)
dev.off()
## End(Not run)
```

Description

Plot EIC of certain m/z and return dataframe for integration

Usage

```r
plotmsrt(data, ms, rt, n = F)
```

Arguments

- **data**: imported data matrix of GC-MS
- **ms**: m/z to be extracted
- **rt**: vector range of the retention time
- **n**: logical smooth or not

Value

dataframe with with the first column RT and second column intensity of the SIM ions.
Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file("cdf", package = "faahKO")
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
matrix <- getmd(cdffiles[1])
png('test.png')
plotmz(matrix)
dev.off()
## End(Not run)
```

Description

plot GC/LC-MS data as scatter plot

Usage

```r
plotmz(data, inscf = 5, ...)
```

Arguments

- `data`: imported data matrix of GC-MS
- `inscf`: Log intensity cutoff for peaks, default 5
- `...`: parameters for ‘plot’ function

Value

scatter plot

Examples

```r
## Not run:
matrix <- getmd(rawdata)
plotmsrt(matrix, rt = c(500,1000), ms = 300)
## 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 = T,
  scale = T,
  xrange = NULL,
  yrange = NULL,
  pch = NULL,
  ...
)

Arguments
- data: mzrt profile with row peaks and column 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))
plotridges  
*Relative Log Abundance Ridge (RLAR) plots*

**Description**
Relative Log Abundance Ridge (RLAR) plots

**Usage**
```r
plotridges(data, lv, type = "g")
```

**Arguments**
- `data`: data as mzrt profile
- `lv`: factor vector for the group information
- `type`: 'g' means group median based, other means all samples median based.

**Value**
Relative Log Abundance Ridge (RLAR) plots

**Examples**
```r
data(list)
plotridges(list$data, as.factor(list$group))
```

plotrla  
*Relative Log Abundance (RLA) plots*

**Description**
Relative Log Abundance (RLA) plots

**Usage**
```r
plotrla(data, lv, type = "g")
```

**Arguments**
- `data`: data as mzrt profile
- `lv`: factor vector for the group information
- `type`: 'g' means group median based, other means all samples median based.

**Value**
Relative Log Abundance (RLA) plots
plotrsd

Examples

data(list)
plotrla(list$data, as.factor(list$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)

plotrtms

Plot mass spectrum of certain retention time and return mass spectrum vector (MSP file) for NIST search

Description

Plot mass spectrum of certain retention time and return mass spectrum vector (MSP file) for NIST search

Usage

plotrtms(data, rt, ms, msp = F)
Arguments

- **data**: imported data matrix of GC-MS
- **rt**: vector range of the retention time
- **ms**: vector range of the m/z
- **msp**: logical, return MSP files or not, default False

Value

plot, vector and MSP files for NIST search

Examples

```r
## Not run:
matrix <- getmd(rawdata)
plotrtms(matrix, rt = c(500, 1000), ms = (300, 500))
## End(Not run)
```

--

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

Description

Plot the intensity distribution of GC-MS

Usage

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

Arguments

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

Examples

```r
## Not run:
data1 <- getmd('sample1-1')
data2 <- getmd('sample1-2')
data3 <- getmd('sample1-3')
data4 <- getmd('sample1-4')
data5 <- getmd('sample1-5')
data <- (data1+data2+data3+data4+data5)/5
datasd <- sqrt(((data1-data)^2+(data2-data)^2+(data3-data)^2+(data4-data)^2+(data5-data)^2)/4)
databrsd <- datasd/data
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:
matrix <- getmd(rawdata)
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 = F, 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:
matrix <- getmd(rawdata)
plott(matrix)

## End(Not run)

plottic

Plot Total Ion Chromatogram (TIC)

Description

Plot Total Ion Chromatogram (TIC)

Usage

plottic(data, n = F)

Arguments

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

Value

plot

Examples

## Not run:
matrix <- getmd(rawdata)
plottic(matrix)

## End(Not run)

qbatch

Get the MIR from the file

Description

Get the MIR from the file

Usage

qbatch(file, mz1, mz2, rt = c(8.65, 8.74), brt = c(8.74, 8.85))
runMDPlot

Arguments

- **file**: data file, CDF or other format supported by xcmsRaw
- **mz1**: the lowest mass
- **mz2**: the highest mass
- **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

Value

- **arearatio**

Examples

```r
## Not run:
arearatio <- qbatch(datafile)
## End(Not run)
```

Description

Shiny application for interactive mass defect plots analysis

Usage

```r
runMDPlot()
```

runsccp

Shiny application for Short-Chain Chlorinated Paraffins analysis

Description

Shiny application for Short-Chain Chlorinated Paraffins analysis

Usage

```r
runsccp()
```
**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

---

**Description**

Get the differences of two GC/LC-MS data

**Usage**

```
submd(data1, data2, mzstep = 0.1, rtstep = 0.01)
```

**Arguments**

- **data1** data file path of first data
- **data2** data file path of second data
- **mzstep** the m/z step for generating matrix data from raw mass spectral data
- **rtstep** the alignment accuracy of retention time, e.g. 0.01 means the retention times of combined data should be the same at the accuracy 0.01s. Higher rtstep would return less scans for combined data
svabatch

Value

list four matrix with the row as scantime in second and column as m/z, the first matrix refer to data 1, the second matrix refer to data 2, the third matrix refer to data1 - data2 while the fourth refer to data2 - data1, minus values are imputed by 0

Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file('cdf', package = 'faahKO')
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
matrix <- submd(cdffiles[1], cdffiles[7])

## End(Not run)
```

svabatch

Plot the influences of DoE and Batch effects on each peaks

Description

Plot the influences of DoE and Batch effects on each peaks

Usage

```r
svabatch(df, dfsv, dfanova)
```

Arguments

- `df`: data output from `svacor` function
- `dfsv`: data output from `svaplot` function for corrected data
- `dfanova`: data output from `svaplot` function for raw data

Value

influences plot

See Also

`svacor`, `svaplot`, `svapca`
Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file("cdf", package = "faahKO")
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
xset <- xcmsSet(cdffiles)
xset <- group(xset)
xset2 <- retcor(xset, family = "symmetric", plottype = "mdevden")
xset2 <- group(xset2, bw = 10)
xset3 <- fillPeaks(xset2)
df <- svacor(xset3)
dfsv <- svaplot(xset3)
dfanova <- svaplot(xset3, pqvalues = "anova")
svabatch(df, dfsv, dfanova)

## End(Not run)
```

svacor

Surrogate variable analysis (SVA) to correct the unknown batch effects

Description

Surrogate variable analysis (SVA) to correct the unknown batch effects

Usage

```r
svacor(xset, lv = NULL, method = "medret", intensity = "into")
```

Arguments

- `xset`: xcmsset object
- `lv`: group information
- `method`: parameter for groupval function
- `intensity`: parameter for groupval function

Details

This is used for revised version of SVA to correct the unknown batch effects

Value

List object with various components such as raw data, corrected data, signal part, random errors part, batch part, p-values, q-values, mass,.rt, Posterior Probabilities of Surrogates variables and Posterior Probabilities of Mod. If no surrogate variable found, corresponding part would miss.

See Also

svapca, svaplot, svabatch
svadata

Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file("cdf", package = "faahKO")
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
xset <- xcmsSet(cdffiles)
xset <- group(xset)
xset2 <- retcor(xset, family = "symmetric", plottype = "mdevden")
xset2 <- group(xset2, bw = 10)
xset3 <- fillPeaks(xset2)
df <- svacor(xset3)
## End(Not run)
```

svadata

Filter the data with p value and q value

Description

Filter the data with p value and q value

Usage

```r
svadata(list, pqvalues = "sv", pt = 0.05, qt = 0.05)
```

Arguments

- `list`: results from `svacor` function
- `pqvalues`: method for ANOVA or SVA
- `pt`: threshold for p value, default is 0.05
- `qt`: threshold for q value, default is 0.05

Value

data, corrected data, mz and retention for fileted data

Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file("cdf", package = "faahKO")
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
xset <- xcmsSet(cdffiles)
xset <- group(xset)
xset2 <- retcor(xset, family = "symmetric", plottype = "mdevden")
xset2 <- group(xset2, bw = 10)
xset3 <- fillPeaks(xset2)
df <- svacor(xset3)
```
svadata(df)

## End(Not run)

svapca

Principal component analysis(PCA) for SVA corrected data and raw data

Description
Principal component analysis(PCA) for SVA corrected data and raw data

Usage
svapca(list, center = T, scale = T, lv = NULL)

Arguments
- **list**: results from svacor function
- **center**: parameters for PCA
- **scale**: parameters for scale
- **lv**: group information

Value
- plot

See Also
svacor, svaplot, svabatch

Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file("cdf", package = "faahKO")
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
xset <- xcmsSet(cdffiles)
xset <- group(xset)
xset2 <- retcor(xset, family = "symmetric", plottype = "mdevden")
xset2 <- group(xset2, bw = 10)
xset3 <- fillPeaks(xset2)
df <- svacor(xset3)
svapca(df)

## End(Not run)
```
Filter the data with p value and q value and show them

Usage

svaplot(list, pqvalues = "sv", pt = 0.05, qt = 0.05, lv = NULL, index = NULL)

Arguments

- **list**: results from svacor function
- **pqvalues**: method for ANOVA or SVA
- **pt**: threshold for p value, default is 0.05
- **qt**: threshold for q value, default is 0.05
- **lv**: group information
- **index**: index for selected peaks

Value

heatmap for the data

See Also

svacor, svapca, svabatch

Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file("cdf", package = "faahKO")
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
xset <- xcmsSet(cdffiles)
xset <- group(xset)
xset2 <- retcor(xset, family = "symmetric", plottype = "mdevden")
xset2 <- group(xset2, bw = 10)
xset3 <- fillPeaks(xset2)
df <- svacor(xset3)
svaplot(df)

## End(Not run)
```
svaupload  
*Get the corrected data after SVA for metabolanalyst*

**Description**

Get the corrected data after SVA for metabolanalyst

**Usage**

```r
data(TBBPA)
```

**Arguments**

- `xset`  
  xcmsset object

- `lv`  
  group information

**Value**

csv files for both raw and corrected data for metabolanalyst if SVA could be applied

**Examples**

```r
## Not run:
library(faahKO)
cdfpath <- system.file("cdf", package = "faahKO")
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
xset <- xcmsSet(cdffiles)
xset <- group(xset)
xset2 <- retcor(xset, family = "symmetric", plottype = "mdevden")
xset2 <- group(xset2, bw = 10)
xset3 <- fillPeaks(xset2)
svaupload(xset3)
## End(Not run)
```
writeMSP

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


writeMSP  Write MSP files for NIST search

Description

Write MSP files for NIST search

Usage

writeMSP(mz, outfilename = "unknown")

Arguments

mz a intensity vector, who name is the mass in m/z
outfilename the name of the MSP file, default is 'unknown'

Value

none a MSP file will be created at the subfolder working dictionary with name 'MSP'

Examples

## Not run:
mz <- c(10000,20000,10000,30000,5000)
names(mz) <- c("101","143","189","221","234")
writeMSP(mz,'test')

## End(Not run)
Index

*Topic datasets
  list, 40
  sccp, 58
  TBBPA, 64

batch, 4

cbmd, 4

findline, 5
findmet, 6
findohc, 6

getarea, 7, 8, 33
getareastd, 8, 33
getbgremove, 9
getbiotechrep, 10
getcsv, 10, 17, 27
getdata, 11, 14, 17, 20, 25, 27, 31, 36
getdata2, 12, 13, 14, 17, 20, 27, 31, 37
getdow, 14, 20, 25, 27, 29–31
getdwtus, 15
getfeaturesanova, 15
getfeaturesrest, 16
getfilter, 17, 27
getformula, 18
getgrouprep, 18
getimputation, 14, 17, 19, 29–31
getIntegration, 20
GetIsotopologues, 21
getmass, 21
getmassdefect, 22, 47
getmd, 23
getmdh, 23
getmdr, 24
getmr, 14, 17, 20, 25, 29–31
getmzr, 12, 14, 17, 20, 25, 26, 27–31, 36, 38
getmzr2, 27, 37, 38
getmzrtcsv, 27, 28
getoverlapmass, 28, 29, 30
getoverlappeak, 29, 30
getoverlaprpt, 29, 30
getpower, 14, 30
getpqi, 31
getQCraw, 31
getrmdd, 32
getccp, 8, 33
getsim, 34
gettechrep, 34
gettimegrouprep, 35
getupload, 25, 36, 37
getupload2, 27, 37
getupload3, 37
gifmr, 38
Integration, 39

list, 40

ma, 40
Mode, 41

plotcc, 41
plotden, 42
plotdwtus, 42
plote, 43
plotgroup, 44
plothist, 44
plotm, 45
plotint, 45
plotintslope, 46
plotkms, 22, 46
plotmr, 47
plotmrc, 48
plotms, 48
plotmsr, 49
plotmz, 50
plotpca, 51
plotridges, 52
plotrla, 52

66
plotrsd, 53
plotrtms, 53
plotsms, 54
plotsub, 55
plott, 55
plottic, 56
qbatch, 56
runMDPlot, 57
runsccp, 57
sccp, 58
submd, 58
svabatch, 59, 60, 62, 63
svacor, 59, 60, 62, 63
svadata, 61
svapca, 59, 60, 62, 63
svaplot, 59, 60, 62, 63
svaupload, 64
TBBPA, 64
writeMSP, 65