Package ‘enviGCMS’

April 26, 2018

Type Package
Title GC/LC-MS Data Analysis for Environmental Science
Version 0.5.0
Date 2018-04-25
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 as raw data process, 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
VignetteBuilder knitr

biocViews

Depends R (>= 2.10)
Imports xcms, MSnbase, rcdn, RColorBrewer, mixtools, BiocParallel, genefilter, grDevices, graphics, stats, utils, methods, reshape2, animation (>= 2.2.3), rmarkdown, shiny, broom

RoxygenNote 6.0.1

NeedsCompilation no

Author Miao YU [aut, cre] (<https://orcid.org/0000-0002-2804-6014>)

Repository CRAN

Date/Publication 2018-04-25 22:53:04 UTC
R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>batch</td>
<td>3</td>
</tr>
<tr>
<td>cbmd</td>
<td>4</td>
</tr>
<tr>
<td>findline</td>
<td>4</td>
</tr>
<tr>
<td>getarea</td>
<td>5</td>
</tr>
<tr>
<td>getareastd</td>
<td>6</td>
</tr>
<tr>
<td>getbgremove</td>
<td>6</td>
</tr>
<tr>
<td>getbiotechrep</td>
<td>7</td>
</tr>
<tr>
<td>getdata</td>
<td>8</td>
</tr>
<tr>
<td>getdata2</td>
<td>9</td>
</tr>
<tr>
<td>getdoe</td>
<td>10</td>
</tr>
<tr>
<td>getfeaturesanova</td>
<td>11</td>
</tr>
<tr>
<td>getfeaturest</td>
<td>11</td>
</tr>
<tr>
<td>getgrouprep</td>
<td>12</td>
</tr>
<tr>
<td>getimputation</td>
<td>13</td>
</tr>
<tr>
<td>GetIntegration</td>
<td>14</td>
</tr>
<tr>
<td>Getisotopologues</td>
<td>15</td>
</tr>
<tr>
<td>getmassdefect</td>
<td>15</td>
</tr>
<tr>
<td>getmd</td>
<td>16</td>
</tr>
<tr>
<td>getmr</td>
<td>17</td>
</tr>
<tr>
<td>getmzrt</td>
<td>18</td>
</tr>
<tr>
<td>getmzr2</td>
<td>18</td>
</tr>
<tr>
<td>getmzr2csv</td>
<td>18</td>
</tr>
<tr>
<td>getQCraw</td>
<td>19</td>
</tr>
<tr>
<td>getsccp</td>
<td>20</td>
</tr>
<tr>
<td>getsim</td>
<td>20</td>
</tr>
<tr>
<td>getsccp</td>
<td>21</td>
</tr>
<tr>
<td>gettechrep</td>
<td>21</td>
</tr>
<tr>
<td>gettimegrouprep</td>
<td>22</td>
</tr>
<tr>
<td>getupload</td>
<td>23</td>
</tr>
<tr>
<td>getupload2</td>
<td>24</td>
</tr>
<tr>
<td>gifmr</td>
<td>25</td>
</tr>
<tr>
<td>Integration</td>
<td>26</td>
</tr>
<tr>
<td>ma</td>
<td>26</td>
</tr>
<tr>
<td>Mode</td>
<td>27</td>
</tr>
<tr>
<td>plot</td>
<td>27</td>
</tr>
<tr>
<td>plotgroup</td>
<td>27</td>
</tr>
<tr>
<td>plotist</td>
<td>28</td>
</tr>
<tr>
<td>plot</td>
<td>29</td>
</tr>
<tr>
<td>plothist</td>
<td>29</td>
</tr>
<tr>
<td>plot</td>
<td>29</td>
</tr>
<tr>
<td>plotint</td>
<td>30</td>
</tr>
<tr>
<td>plotintslope</td>
<td>30</td>
</tr>
<tr>
<td>plotkms</td>
<td>31</td>
</tr>
<tr>
<td>plotmr</td>
<td>31</td>
</tr>
<tr>
<td>plotmrc</td>
<td>31</td>
</tr>
<tr>
<td>plotms</td>
<td>32</td>
</tr>
<tr>
<td>plotms</td>
<td>32</td>
</tr>
<tr>
<td>plotmrsr</td>
<td>33</td>
</tr>
<tr>
<td>plotmz</td>
<td>34</td>
</tr>
<tr>
<td>plotpca</td>
<td>34</td>
</tr>
</tbody>
</table>
**Description**

Get the MIR and related information from the files

**Usage**

`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(file, 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

`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

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))`
getarea

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

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

## End(Not run)
```

Description

Get the peak information from samples for SCCPs detection

Usage

```r
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`, `getscsp`
getareastd

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

getbgremove

Get the peak list with blank samples’ peaks removed

Description
Get the peak list with blank samples’ peaks removed

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

Get the report for biological replicates.

**Description**

Get the report for biological replicates.

**Usage**

```r
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

**Examples**

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

## End(Not run)
```
Value
dataframe with mean, standard deviation and RSD for those technique replicates & biological replicates combined with raw data

getdata

Description
Get xcmsset object in one step with optimized methods.

Usage
getdata(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

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, getupload, 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

```r
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(),
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

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, index = NULL)

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
- **index**: the index of peaks considered, default NULL

Value
list with group information, filtered peaks and index

See Also
`getdata`, `getupload2`, `getmzrt2`, `getimputation`, `getmr`

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

## End(Not run)
```
getfeaturesanova

Description

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

Usage

getfeaturesanova(list, power = 0.8, pt = 0.05, qt = 0.05, n = 3,
ng = 3, rsdcf = 100, inscf = 5, imputation = "1", 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: 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

getfeatures t

Description

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

Usage

getfeatures t(list, power = 0.8, pt = 0.05, qt = 0.05, n = 3,
inscf = 5, rsdcf = 30, imputation = "1", index = NULL)
**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
- **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
- **index**: the index of peaks considered, default NULL

**Value**

dataframe with peaks fit the setting above

**Examples**

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

---

**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

**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
getimputation

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
getimputation(list, method = "l")

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</td>
</tr>
<tr>
<td>method</td>
<td>'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'.</td>
</tr>
</tbody>
</table>

Value
list with imputed peaks

See Also
getdata2, getdata, getmzrt, getmzrt2, getdooe, getmr

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

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 = "C12OH6Br4", charge = 1, width = 0.3)

Arguments

formula the molecular formula. C12OH6Br4 means BDE-47 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.

Examples

## Not run:
# show isotopologues for BDE-47
ir <- Getisotopologues(formula = 'C12OH6Br4')

## End(Not run)

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
getmd

Description

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

Usage

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 scan time

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)
```
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

gedata, 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 for batch correction and plot as a list

Description
Get the mzrt profile and group information for batch correction and plot as a list

Usage
getmzrt(xset, name = NULL)

Arguments
  xset  xcmsSet objects
  name  file name for csv file, default NULL

Value
list with rtmz profile and group information

See Also
gedata, getupload, getmzrt2, getdoe, getmzrt

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

## 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
gemat2(xset, name = NULL)
getmzrtcsv

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`

Examples

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

# End(Not run)
```

getmzrtcsv  |  **Covert the peaks list csv file into list**

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

See Also

- `getmzrt`, `getmzrt2`
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, rtrange, 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

getsccep  
Quantitative analysis for short-chain chlorinated paraffins (SCCPs)

Description
Quantitative analysis for short-chain chlorinated paraffins (SCCPs)

Usage
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
getsim

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

gettechrep Get the report for technique replicates.

Description

Get the report for technique replicates.

Usage

gettechrep(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
in scf intensity cutoff for peaks, default 1000
Value
dataframe with mean, standard deviation and RSD for those technique replicates combined with raw data

gettimegrouprep

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

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 to be submitted to Metaboanalyst

Description

Get the csv files to be submitted to Metaboanalyst

Usage

getupload(xset, method = "medret", value = "into", name = "Peaklist")

Arguments

- **xset**: the xcmsset object which you want to submitted to Metaboanalyst
- **method**: parameter for groupval function
- **value**: parameter for groupval function
- **name**: file name

Value

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

See Also

gedata, upload2, getmzrt

Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file('cdf', 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

```r
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

```r
## Not run:
library(faahKO)
cdfpath <- system.file('cdf', package = 'faahKO')
xset <- getdata2(cdfpath,
    ppp = xcms::MatchedFilterParam(),
    rtp = xcms::ObiwrapParam(),
    gpp = xcms::PeakDensityParam())
getupload2(xset)

## End(Not run)
```
**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 = "i",
     index = NULL, file = "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
- `index` the index of peaks considered, default NULL
- `file` file name for gif file, default NULL
- `...` parameters for ‘plot’ function

**Value**

gif file

**Examples**

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

## 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

```r
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 integration data

#### Examples

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

### 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 identify the size of the moving box.
**Mode**

**Value**

The filtered data

**Examples**

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

---

**Description**

define the Mode function

**Usage**

```r
Mode(x)
```

**Arguments**

- `x` vector

**Value**

Mode of the vector

---

**plote**

plot EIC and boxplot for all peaks and return diffreport

---

**Description**

plot EIC and boxplot for all peaks and return diffreport

**Usage**

```r
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

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

## End(Not run)
```

---

**plotgroup**

*Plot the response group of GC-MS*

Description

Plot the response group of GC-MS

Usage

```r
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)
plotmd(data)

## End(Not run)
```
plotHist

plot the density of the GC-MS data with EM algorithm to separate the data into two log normal distributions.

Description

plot the density of the GC-MS data with EM algorithm to separate the data into two log normal distributions.

Usage

plotHist(data)

Arguments

data imported data matrix of GC-MS

Examples

## 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

plothm(data, lv, index = NULL)

Arguments

data mzrt profile with row peaks and column samples
lv group information
index index for selected peaks
plotint

plot the information of integration

Description
plot the information of integration

Usage
plotint(list, name = NULL)

Arguments
list list from getinteragtion
name the title of the plot

Examples
## Not run:
list <- getinteragtion(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 getinteragtion
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

gettmassdefect

Examples

## Not run:
mz <- c(10000, 50000, 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", index = NULL, ...)

## Not run:
mz <- c(10000, 50000, 20000, 100, 40000)
names(mz) <- c(100.1022, 245.2122, 267.3144, 400.1222, 707.2294)
plotmr(mz)

## End(Not run)
plotmrc

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
- imputation: parameters for ‘getimputation’ function method
- index: the index of peaks considered, default NULL
- ... parameters for ‘plot’ function

Value

data fit the cutoff

Examples

```r
## Not run:
library(faahKO)
cdfpath <- system.file('cdf', package = 'faahKO')
list <- getmr(cdfpath, pmethod = '1')
plotmrc(list)
## End(Not run)
```

Description

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

Usage

```r
plotmrc(list, ms = c(100, 800), inscf = 5, rsdcf = 30, imputation = "1", index = NULL, ...)
```

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
index             the index of peaks considered, default NULL
...                parameters for 'plot' function

Examples

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

## End(Not run)
### plotmsrt

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

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

### plotmz

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

**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
plotpca

Value

scatter plot

Examples

## 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)

---

plotpca  
*plot the PCA of list*

---

Description

plot the PCA of list

Usage

plotpca(data, lv = NULL, index = NULL, center = T, scale = T, ...)

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
...  other parameters for 'plot' function

Examples

## Not run:
library(faahKO)
cdfpath <- system.file('cdf', package = 'faahKO')
list <- getmr(cdfpath, pmethod = '')
data <- list$data
lv <- as.character(list$group$class)
plotpca(data, lv)

## End(Not run)
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 = "1", index = NULL, ...)

Arguments

list  list with data as peaks list, mz, rt and group information
ms    the mass range to plot the data
inacf  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
index  the index of peaks considered, default NULL
...    other parameters for `plot` function

Examples

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

## End(Not run)

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)
**plotsms**

**Arguments**

- **data**: imported data matrix of GC-MS
- **rt**: vector range of the retention time
- **ms**: vector range of the m/z

**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)
```

**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**

```r
## 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)
plottic(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))
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 Short-Chain Chlorinated Paraffins analysis

**Usage**

```r
runsccep()
```

---

**Description**

A dataset containing the ions, formula, Cl

**Usage**

```r
data(s CCP)
```
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

---

**submd**

Get the differences of two GC/LC-MS data

---

Description

Get the differences of two GC/LC-MS data

Usage

```r
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

Value

A list of four matrices with the row as scantime in second and column as m/z, the first matrix refers to data 1, the second matrix refers to data 2, the third matrix refers to data1 - data2 while the fourth refers 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

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, pvalues = "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

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 surrogate variables, and posterior probabilities of mod. If no surrogate variable found, corresponding part would miss.

See Also

svapca, svaplot, svabatch

Examples

```r
## Not run:
library(fahKO)
cdfpath <- system.file("cdf", package = "fahKO")
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
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)
```
svaplot

Filter the data with p value and q value and show them

Description

Filter the data with p value and q value and show them

Usage

```r
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
svaupload(xset, lv = NULL)

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)
savaupload(xset3)

## End(Not run)
```

writeMSP

Write MSP files for NIST search

Description
Write MSP files for NIST search

Usage
writeMSP(mz, outfilename = "unknown")
writeMSP

Arguments

mz          a intensity vector, whose name is the mass in m/z
out filename 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

```R
## Not run:
mz <- c(10000, 20000, 10000, 30000, 50000)
names(mz) <- c(101, 143, 189, 221, 234)
writeMSP(mz, 'test')

## End(Not run)
```
Index

*Topic datasets
  sccp, 40

batch, 3

cbmd, 4

findline, 4

getarea, 5, 6, 21
getareastd, 5, 6, 21
getbgremove, 6
getbiotechrep, 7
getdata, 8, 10, 13, 17, 18, 23
getdata2, 8, 9, 10, 13, 19, 24
getdeo, 10, 13, 17–19
getfeaturesanova, 11
getfeaturesrest, 11
getgrouprep, 12
getimputation, 10, 13
GetIntegration, 14
Getisotopologues, 15
getmassdefect, 15, 31
getmd, 16
getmr, 10, 13, 17
getmzrt, 8, 10, 13, 17, 18, 19, 23
getmzrt2, 10, 13, 18, 19, 24
getmzrtcsv, 19, 19
getQcraw, 20
getscpp, 5, 6, 20
getsim, 21
gettechrep, 21
gettimegrouprep, 22
getupload, 8, 17, 18, 23, 24
getupload2, 10, 19, 23, 24
gifmr, 25

Integration, 26

ma, 26
Mode, 27

plote, 27
plotgroup, 28
plothist, 29
plothm, 29
plotint, 30
plotintslope, 30
plotkms, 16, 31
plotmr, 31
plotmrc, 32
plotsms, 33
plotmsrt, 34
plotmsz, 34
plotpca, 35
plotrsd, 36
plotrtrms, 36
plotsms, 37
plotsub, 38
plott, 38
plottie, 39

qbatch, 39
runscpp, 40

sccp, 40
submd, 41
svabatch, 42, 43, 45, 46
svacor, 42, 43, 45, 46
svadata, 44
svapca, 42, 43, 45, 46
svaplot, 42, 43, 45, 46
svaupload, 47

writeMSP, 47

49