

# Package: enviGCMS (via r-universe)

August 30, 2024

**Type** Package

**Title** GC/LC-MS Data Analysis for Environmental Science

**Version** 0.7.3

**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>

**BugReports** <https://github.com/yufree/enviGCMS/issues>

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**Suggests** knitr, testthat, xcms, MSnbase, plotly, shiny, rmarkdown, DT, crosstalk

**VignetteBuilder** knitr

**Depends** R (>= 2.10)

**Imports** Rdisop, BiocParallel, grDevices, graphics, stats, utils, methods, animation (>= 2.2.3), RColorBrewer, mixtools, data.table, igraph

**RoxygenNote** 7.2.3

**Repository** <https://yufree.r-universe.dev>

**RemoteUrl** <https://github.com/yufree/envigcms>

**RemoteRef** HEAD

**RemoteSha** 3188a2b78493f52abb915ead1a49ede59dd3f1c2

## Contents

batch . . . . .	4
cbmd . . . . .	4

dotpanno . . . . .	5
findline . . . . .	6
findlipid . . . . .	6
findmet . . . . .	7
findohc . . . . .	8
findpfc . . . . .	9
getalign . . . . .	10
getalign2 . . . . .	10
getarea . . . . .	11
getareastd . . . . .	12
getbgremove . . . . .	12
getbiotechrep . . . . .	13
getcompare . . . . .	14
getcsv . . . . .	15
getdata . . . . .	16
getdata2 . . . . .	17
getdoe . . . . .	18
getdwtus . . . . .	19
getfeaturesanova . . . . .	20
getfeaturest . . . . .	21
getfilter . . . . .	21
getformula . . . . .	22
getgroupprep . . . . .	23
getimputation . . . . .	24
GetIntegration . . . . .	24
Getisotopologues . . . . .	25
getmass . . . . .	26
getmassdefect . . . . .	27
getmd . . . . .	27
getmdh . . . . .	28
getmdr . . . . .	29
getmr . . . . .	29
getmslanno . . . . .	30
getMSP . . . . .	31
getmzrt . . . . .	31
getmzrt2 . . . . .	33
getmzrtcsv . . . . .	34
getoverlapeak . . . . .	34
getpn . . . . .	35
getpower . . . . .	35
getpqi . . . . .	36
getQCraw . . . . .	36
getrangecsv . . . . .	37
getretcor . . . . .	37
getrmd . . . . .	38
getscpp . . . . .	39
getsim . . . . .	40
gettechrep . . . . .	40

gettimegrouprep . . . . .	41
getupload . . . . .	42
getupload2 . . . . .	43
getupload3 . . . . .	43
gifmr . . . . .	44
Integration . . . . .	45
list . . . . .	46
ma . . . . .	46
Mode . . . . .	47
plotanno . . . . .	47
plotcc . . . . .	48
plotden . . . . .	48
plotdwtus . . . . .	49
plote . . . . .	49
plotgroup . . . . .	50
plothist . . . . .	51
plothm . . . . .	51
plotint . . . . .	52
plotintslope . . . . .	53
plotkms . . . . .	53
plotmr . . . . .	54
plotmrc . . . . .	55
plotsms . . . . .	55
plotsmrt . . . . .	56
plotmz . . . . .	57
plotpca . . . . .	57
plotpeak . . . . .	58
plotridge . . . . .	59
plotridges . . . . .	60
plotrla . . . . .	60
plotsrd . . . . .	61
plotrtms . . . . .	62
plotrug . . . . .	62
plotsms . . . . .	63
plotsub . . . . .	64
plott . . . . .	64
plottic . . . . .	65
qbatch . . . . .	65
runMDPlot . . . . .	66
runsccp . . . . .	66
sccp . . . . .	67
submd . . . . .	67
svabatch . . . . .	68
svacor . . . . .	69
svadata . . . . .	70
svapca . . . . .	71
svaplot . . . . .	72
svaupload . . . . .	73

TBBPA . . . . .	73
writeMSP . . . . .	74
xrankanno . . . . .	75

<b>Index</b>	<b>76</b>
--------------	-----------

---

batch	<i>Get the MIR and related information from the files</i>
-------	---

---

### 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

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

## End(Not run)
```

---

cbmd	<i>Combine two data with similar retention time while different mass range</i>
------	--

---

### 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)
```

---

dotpanno

*Perform MS/MS dot product annotation for mgf file*

---

**Description**

Perform MS/MS dot product annotation for mgf file

**Usage**

```
dotpanno(file, db = NULL, ppm = 10, prems = 1.1, binstep = 1, consinc = 0.6)
```

**Arguments**

file	mgf file generated from MS/MS data
db	database could be list object from 'getMSP'
ppm	mass accuracy, default 10
prems	precursor mass range, default 1.1 to include M+H or M-H
binstep	bin step for consin similarity
consinc	consin similarity cutoff for annotation. Default 0.6.

**Value**

list with MSMS annotation results

---

findline	<i>find line of the regression model for GC-MS</i>
----------	--

---

**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)
```

---

findlipid	<i>Find lipid class of metabolites base on referenced Kendrick mass defect</i>
-----------	--

---

**Description**

Find lipid class of metabolites base on referenced Kendrick mass defect

**Usage**

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

**Arguments**

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

**Value**

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

**References**

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

**Examples**

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

---

findmet

*Screen metabolites by Mass Defect*

---

**Description**

Screen metabolites by Mass Defect

**Usage**

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

**Arguments**

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

**Value**

list with filtered metabolites mass to charge index of certain compound

---

findohc	<i>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.</i>
---------	---

---

### Description

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

### Usage

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

### Arguments

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

### Value

list with filtered organohalogen compounds



## References

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

---

findpfc

*Find PFCs based on mass defect analysis*

---

## Description

Find PFCs based on mass defect analysis

## Usage

```
findpfc(list)
```

## Arguments

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

## Value

list list with potential PFCs compounds index

## References

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

## Examples

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

---

getalign	<i>Align two peaks vectors by mass to charge ratio and/or retention time</i>
----------	--

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

data frame with aligned peaks table

**Examples**

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

---

getalign2	<i>Align mass to charge ratio and/or retention time to remove redundancy</i>
-----------	--

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

index for

**Examples**

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

---

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](#), [getscpp](#)

---

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  
)
```

**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**

```
## Not run:  
library(faahK0)  
cdfpath <- system.file("cdf", package = "faahK0")  
xset <- getdata(cdfpath, pmethod = ' ' )  
getbgremove(xset)  
  
## End(Not run)
```

---

getbiotechrep	<i>Get the report for biological replicates.</i>
---------------	--

---

**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

---

getcompare	<i>Align multiple peaks list to one peak list</i>
------------	---

---

**Description**

Align multiple peaks list to one peak list

**Usage**

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

**Arguments**

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

**Value**

list object with aligned mzrt objects

---

getcsv *Convert an list object to csv file.*

---

### Description

Convert an list object to csv file.

### Usage

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

### Arguments

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

### Value

NULL, csv file

### References

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

### Examples

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

---

`getdata`*Get xcmsset object in one step with optimized methods.*

---

**Description**

Get xcmsset object in one step with optimized methods.

**Usage**

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

**Arguments**

<code>path</code>	the path to your data
<code>index</code>	the index of the files
<code>BPPARAM</code>	used for BiocParallel package
<code>pmethod</code>	parameters used for different instrumentals such as 'hplcorbitrap', 'uplcorbitrap', 'hplcqtof', 'hplchqtof', 'uplcqtof', 'uplchqtof'. The parameters were from the reference
<code>minfrac</code>	minimum fraction of samples necessary in at least one of the sample groups for it to be a valid group, default 0.67
<code>...</code>	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 submit the results to a paper, remember to include those parameters.

**Value**

a xcmsset object for that path or selected samples

**References**

Patti, G. J.; Tautenhahn, R.; Siuzdak, G. Nat. Protocols 2012, 7 (3), 508–516.



**See Also**

[getdata2](#), [getmzrt](#)

**Examples**

```
## Not run:
library(faahK0)
cdfpath <- system.file('cdf', package = 'faahK0')
xset <- getdata(cdfpath, pmethod = ' ')

## End(Not run)
```

---

getdata2	<i>Get XCMSnExp object in one step from structured folder path for xcms 3.</i>
----------	--

---

**Description**

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

**Usage**

```
getdata2(
  path,
  index = FALSE,
  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 sub-directories 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. <code>xcms::CentWaveParam()</code>
rtp	parameters for retention time correction, e.g. <code>xcms::ObiwrapParam()</code>
gpp	parameters for peaks grouping, e.g. <code>xcms::PeakDensityParam()</code>
fpp	parameters for peaks filling, e.g. <code>xcms::FillChromPeaksParam()</code> , <code>PeakGroupsParam()</code>

### Details

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

### Value

a XCMSnExp object with processed data

### See Also

[getdata](#), [getmzrt](#)

---

getdoe	<i>Generate the group level rsd and average intensity based on DoE,</i>
--------	---

---

### Description

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

### Usage

```
getdoe(
  list,
  inscf = 5,
  rsdcf = 100,
  rsdcft = 30,
  imputation = "1",
  tr = FALSE,
  BPPARAM = BiocParallel::bpparam()
)
```

### Arguments

list	list with data as peaks list, mz, rt and group information
inscf	Log intensity cutoff for peaks across samples. If any peaks show a intensity higher than the cutoff in any samples, this peaks would not be filtered. default 5
rsdcf	the rsd cutoff of all peaks in all group
rsdcft	the rsd cutoff of all peaks in technical replicates

imputation	parameters for 'getimputation' function method
tr	logical. TRUE means dataset with technical replicates at the base level folder
BPPARAM	An optional BiocParallelParam instance determining the parallel back-end to be used during evaluation.

**Value**

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

**See Also**

[getdata2](#), [getdata](#), [getmzrt](#), [getimputation](#), [getmr](#), [getpower](#)

**Examples**

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

---

getdwtus	<i>Density weighted intensity for one sample</i>
----------	--

---

**Description**

Density weighted intensity for one sample

**Usage**

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

**Arguments**

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

**Value**

Density weighted intensity for one sample

**Examples**

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

---

getfeaturesanova	<i>Get the features from anova, with p value, q value, rsd and power restriction</i>
------------------	--

---

### 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

---

getfeaturest	<i>Get the features from t test, with p value, q value, rsd and power restriction</i>
--------------	---

---

**Description**

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

**Usage**

```
getfeaturest(list, power = 0.8, pt = 0.05, qt = 0.05, n = 3, imputation = "l")
```

**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	<i>Filter the data based on row and column index</i>
-----------	--

---

**Description**

Filter the data based on row and column index

**Usage**

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

**Arguments**

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

**Value**

list with remain peaks, and filtered peaks index

**See Also**

[getdata2](#), [getdata](#), [getmzrt](#), [getimputation](#), [getmr](#), [getcsv](#)

**Examples**

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

---

getformula

*Get chemical formula for mass to charge ratio.*

---

**Description**

Get chemical formula for mass to charge ratio.

**Usage**

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

**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

---

getgrouprep	<i>Get the report for samples with biological and technique replicates in different groups</i>
-------------	--

---

**Description**

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

**Usage**

```
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

```
getimputation(list, method = "1")
```

### 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 '1'.

### 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 integration of certain ion's chromatogram data and plot the data*

---

### Description

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



**Usage**

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

**Arguments**

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

**Value**

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

**Examples**

```
## Not run:  
list <- GetIntegration(data)  
  
## End(Not run)
```

---

Getisotopologues

*Get the selected isotopologues at certain MS data*

---

**Description**

Get the selected isotopologues at certain MS data

**Usage**

```
Getisotopologues(formula = "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**

```
# 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)
```

**Arguments**

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

**Value**

numerical vector

**Examples**

```
getmass('CH2')
```

---

getmassdefect	<i>Get mass defect with certain scaled factor</i>
---------------	---

---

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

---

**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 = FALSE, rtrange = FALSE)
```

**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
rtrange	vector range of the retention time, default all

**Value**

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

**Examples**

```
## Not run:
library(faahK0)
cdfpath <- system.file('cdf', package = 'faahK0')
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**

```
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**

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

---

getmdr	<i>Get the raw Mass Defect</i>
--------	--------------------------------

---

**Description**

Get the raw Mass Defect

**Usage**

```
getmdr(mz)
```

**Arguments**

mz                    numeric vector for exact mass

**Value**

raw Mass Defect

**Examples**

```
getmdr(getmass('C2H4'))
```

---

getmr	<i>Get the mzrt profile and group information for batch correction and plot as a list directly from path with default setting</i>
-------	---

---

**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 = FALSE,  
  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**

```
## Not run:
library(faahK0)
cdfpath <- system.file('cdf', package = 'faahK0')
list <- getmr(cdfpath, pmethod = ' ')

## End(Not run)
```

---

getms1anno	<i>Annotation of MS1 data by compounds database by predefined paired mass distance</i>
------------	--

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

list or data frame

---

getMSP	<i>read in MSP file as list for ms/ms or ms(EI) annotation</i>
--------	--

---

**Description**

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

**Usage**

```
getMSP(file)
```

**Arguments**

file            the path to your MSP file

**Value**

list a list with MSP information for annotation

---

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

---

**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,  
  rtdigit = 1,  
  method = "medret",  
  value = "into",  
  eic = FALSE,  
  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
rtdigit	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 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.

## Value

mzrt object, a list with mzrt profile and group information

## References

Smith, C.A., Want, E.J., O'Maille, G., Abagyan, R., Siuzdak, G., 2006. XCMS: Processing Mass Spectrometry Data for Metabolite Profiling Using Nonlinear Peak Alignment, Matching, and Identification. *Anal. Chem.* 78, 779–787.

## See Also

[getdata](#), [getdata2](#), [getdoe](#), [getcsv](#), [getfilter](#)

## Examples

```
## Not run:
library(faahK0)
cdfpath <- system.file('cdf', package = 'faahK0')
xset <- getdata(cdfpath, pmethod = ' ')
getmzrt(xset, name = 'demo', type = 'mapo')

## End(Not run)
```



---

getmzrt2	<i>Get the mzrt profile and group information for batch correction and plot as a list for xcms 3 object</i>
----------	---

---

## 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](#)

## Examples

```
## Not run:  
library(faahK0)  
cdfpath <- system.file('cdf', package = 'faahK0')  
xset <- getdata2(cdfpath,  
ppp = xcms::MatchedFilterParam(),  
rtp = xcms::ObiwarpParam(),  
gpp = xcms::PeakDensityParam())  
getmzrt2(xset)  
  
## End(Not run)
```

---

getmzrtcsv	<i>Covert the peaks list csv file into list</i>
------------	---

---

**Description**

Covert the peaks list csv file into list

**Usage**

```
getmzrtcsv(path)
```

**Arguments**

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

**Value**

list with rtmz profile and group information as the first row

**See Also**

[getmzrt](#)

---

getoverlapeak	<i>Get the overlap peaks by mass and retention time range</i>
---------------	---

---

**Description**

Get the overlap peaks by mass and retention time range

**Usage**

```
getoverlapeak(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](#)

---

getpn	<i>Merge positive and negative mode data</i>
-------	--

---

**Description**

Merge positive and negative mode data

**Usage**

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

**Arguments**

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

**Value**

mzrt object with group information from pos mode

---

getpower	<i>Get the index with power restriction for certain study with BH adjusted p-value and certain power.</i>
----------	---

---

**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, power = 0.8, imputation = "1")
```

**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
power	power cutoff, default 0.8
imputation	parameters for 'getimputation' function method

**Value**

list with current power and sample numbers for each peaks

**See Also**

[getdata2](#), [getdata](#), [getmzrt](#), [getimputation](#), [getmr](#), [getdoe](#)

**Examples**

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

---

getpqsi	<i>Compute pooled QC linear index according to run order</i>
---------	--

---

**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	<i>get the data of QC compound for a group of data</i>
----------	--

---

**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

---

getrangecsv	<i>Get a mzrt list and/or save mz and rt range as csv file.</i>
-------------	---

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

NULL, csv file

---

getretcor	<i>Perform peaks list alignment and return features table</i>
-----------	---

---

**Description**

Perform peaks list alignment and return features table

**Usage**

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

**Arguments**

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

**Value**

mzrt object without group information

---

getrmd *Get the Relative Mass Defect*

---

**Description**

Get the Relative Mass Defect

**Usage**

```
getrmd(mz)
```

**Arguments**

mz numeric vector for exact mass

**Value**

Relative Mass Defect

**Examples**

```
getrmd(getmass('C2H4'))
```

---

`getscpp`*Quantitative analysis for short-chain chlorinated paraffins(SCCPs)*

---

**Description**

Quantitative analysis for short-chain chlorinated paraffins(SCCPs)

**Usage**

```
getscpp(  
  pathstds,  
  pathsample,  
  ismz = 323,  
  ppm = 5,  
  con = 2000,  
  rt = NULL,  
  rts = NULL,  
  log = TRUE  
)
```

**Arguments**

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

**Value**

list with peak information

**See Also**

[getareastd](#), [getarea](#)

---

getsim	<i>output the similarity of two dataset</i>
--------	---

---

**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	<i>Get the report for technique replicates.</i>
------------	---

---

**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
inscf	intensity cutoff for peaks, default 1000



**Value**

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

---

gettimegrouprep	<i>Get the time series or two factor DoE report for samples with biological and technique replicates in different groups</i>
-----------------	--

---

**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**

<code>xset</code>	the xcmsset/XCMSnExp/list object which you want to submitted to Metaboanalyst
<code>method</code>	parameter for groupval function
<code>value</code>	parameter for groupval function
<code>name</code>	file name
<code>type</code>	m means Metaboanalyst, a means xMSannotator, o means full information csv
<code>mzdigit</code>	m/z digits of row names of data frame
<code>rtdigit</code>	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(faahK0)  
cdfpath <- system.file('cdf', package = 'faahK0')  
xset <- getdata(cdfpath, pmethod = ' ' )  
getupload(xset)  
  
## End(Not run)
```

---

getupload2	<i>Get the csv files to be submitted to Metaboanalyst</i>
------------	---

---

**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(faahK0)  
cdfpath <- system.file('cdf', package = 'faahK0')  
xset <- getdata2(cdfpath)  
getupload2(xset)  
  
## End(Not run)
```

---

getupload3	<i>Get the csv files to be submitted to Metaboanalyst</i>
------------	---

---

**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 your want to perform local analysis.

**See Also**

[getmzrt](#), [getmzrt2](#)

**Examples**

```
## Not run:
library(faahK0)
cdfpath <- system.file('cdf', package = 'faahK0')
xset <- getdata2(cdfpath,
ppp = xcms::MatchedFilterParam(),
rtp = xcms::ObiwarpParam(),
gpp = xcms::PeakDensityParam())
xset <- enviGCMS::getmzrt2(xset)
getupload3(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",
  name = "test",
  ...
)
```

**Arguments**

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

**Value**

gif file

**Examples**

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

## End(Not run)
```

---

Integration

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

---

**Description**

Just integrate data according to fixed rt and fixed noise area

**Usage**

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

**Arguments**

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

**Value**

area integration data

**Examples**

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

---

list	<i>Demo data</i>
------	------------------

---

**Description**

Demo data

**Usage**

```
data(list)
```

**Format**

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

---

ma	<i>filter data by average moving box</i>
----	--

---

**Description**

filter data by average moving box

**Usage**

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

**Arguments**

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

**Value**

The filtered data

**Examples**

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

---

Mode	<i>define the Mode function</i>
------	---------------------------------

---

**Description**

define the Mode function

**Usage**

Mode(x)

**Arguments**

x                    vector

**Value**

Mode of the vector

---

plotanno	<i>Show MS/MS pmd annotation result</i>
----------	---

---

**Description**

Show MS/MS pmd annotation result

**Usage**

plotanno(anno, ...)

**Arguments**

anno                    list from MSMS anno function  
...                    other parameter for plot function

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

---

**Description**

plot the calibration curve with error bar, r squared and equation.

**Usage**

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

**Arguments**

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

**Examples**

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

---

plotden *plot the density for multiple samples*

---

**Description**

plot the density for multiple samples

**Usage**

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

**Arguments**

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



**Examples**

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

---

plotdwtus	<i>plot density weighted intensity for multiple samples</i>
-----------	---

---

**Description**

plot density weighted intensity for multiple samples

**Usage**

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

**Arguments**

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

**Value**

Density weighted intensity for multiple samples

**Examples**

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

---

plote	<i>plot EIC and boxplot for all peaks and return diffreport</i>
-------	---

---

**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, 'pair' 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(faahK0)
cdfpath <- system.file('cdf', package = 'faahK0')
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**

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

---

plothist	<i>plot the density of the GC-MS data with EM algorithm to separate the data into two log normal distribution.</i>
----------	--

---

**Description**

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

**Usage**

```
plothist(data)
```

**Arguments**

data	imported data matrix of GC-MS
------	-------------------------------

**Examples**

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

---

plothm	<i>Plot the heatmap of mzrt profiles</i>
--------	--

---

**Description**

Plot the heatmap of mzrt profiles

**Usage**

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

**Arguments**

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

**Examples**

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

---

plotint	<i>plot the information of integration</i>
---------	--

---

**Description**

plot the information of integration

**Usage**

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

**Arguments**

list	list from getinteagtion
name	the title of the plot

**Examples**

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

## End(Not run)
```

---

plotintslope                    *plot the slope information of integration*

---

**Description**

plot the slope information of integration

**Usage**

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

**Arguments**

list	list from getintegration
name	the title of the plot

**Examples**

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

---

plotkms                        *plot the kendrick mass defect diagram*

---

**Description**

plot the kendrick mass defect diagram

**Usage**

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

**Arguments**

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

**See Also**

[getmassdefect](#)

**Examples**

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

## End(Not run)
```

---

plotmr

*plot the scatter plot for peaks list with threshold*


---

**Description**

plot the scatter plot for peaks list with threshold

**Usage**

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

**Arguments**

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

**Value**

data fit the cutoff

**Examples**

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

---

plotmrc	<i>plot the diff scatter plot for peaks list with threshold between two groups</i>
---------	--

---

**Description**

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

**Usage**

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

**Arguments**

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

**Examples**

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

---

plotms	<i>plot GC/LC-MS data as a heatmap with TIC</i>
--------	---

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

heatmap

**Examples**

```
## Not run:
library(faahK0)
cdfpath <- system.file('cdf', package = 'faahK0')
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
matrix <- getmd(cdffiles[1])
png('test.png')
plotms(matrix)
dev.off()

## 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**

```
plotmsrt(data, ms, rt, n = FALSE)
```

**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**

```
## 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**

```
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**

```
## Not run:  
library(faahK0)  
cdfpath <- system.file('cdf', package = 'faahK0')  
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 for multiple samples*

---

**Description**

plot the PCA for multiple samples

**Usage**

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

**Arguments**

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

**Value**

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

**Examples**

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

---

plotpeak

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

---

**Description**

plot intensity of peaks across samples or samples across peaks

**Usage**

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

**Arguments**

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

**Value**

parallel coordinates plot

**Examples**

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

---

plotridge	<i>plot ridgeline density plot</i>
-----------	------------------------------------

---

**Description**

plot ridgeline density plot

**Usage**

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

**Arguments**

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

**Value**

ridgeline density plot

**Examples**

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

---

plotridges	<i>Relative Log Abundance Ridge (RLAR) plots for samples or peaks</i>
------------	---

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

Relative Log Abundance Ridge(RLA) plots

**Examples**

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

---

plotrla	<i>Relative Log Abundance (RLA) plots</i>
---------	---

---

**Description**

Relative Log Abundance (RLA) plots

**Usage**

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

**Arguments**

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

**Value**

Relative Log Abundance (RLA) plots

**Examples**

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

---

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

---

**Description**

plot the rsd influences of data in different groups

**Usage**

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

**Arguments**

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

**Examples**

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

---

plotrtms	<i>Plot mass spectrum of certain retention time and return mass spectrum vector (MSP file) for NIST search</i>
----------	--

---

**Description**

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

**Usage**

```
plotrtms(data, rt, ms, msp = FALSE)
```

**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**

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

## End(Not run)
```

---

plotrug	<i>plot 1-d density for multiple samples</i>
---------	--

---

**Description**

plot 1-d density for multiple samples

**Usage**

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

**Arguments**

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

**Examples**

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

---

plotsms	<i>Plot the intensity distribution of GC-MS</i>
---------	---

---

**Description**

Plot the intensity distribution of GC-MS

**Usage**

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

**Arguments**

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

**Examples**

```
## Not run:
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	<i>Plot the background of data</i>
---------	------------------------------------

---

**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	<i>plot GC-MS data as a heatmap for constant speed of temperature rising</i>
-------	--

---

**Description**

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

**Usage**

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

**Arguments**

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

**Value**

heatmap



**Examples**

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

---

plottic

*Plot Total Ion Chromatogram (TIC)*

---

**Description**

Plot Total Ion Chromatogram (TIC)

**Usage**

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

**Arguments**

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

**Value**

plot

**Examples**

```
## Not run:  
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**

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

---

runMDPplot

*Shiny application for interactive mass defect plots analysis*

---

**Description**

Shiny application for interactive mass defect plots analysis

**Usage**

```
runMDPplot()
```

---

runsccp

*Shiny application for Short-Chain Chlorinated Paraffins analysis*

---

**Description**

Shiny application for Short-Chain Chlorinated Paraffins analysis

**Usage**

```
runsccp()
```

---

sccp	<i>Short-Chain Chlorinated Paraffins(SCCPs) peaks information for quantitative analysis</i>
------	---

---

**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

---

submd	<i>Get the differences of two GC/LC-MS data</i>
-------	---

---

**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

**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**

```
## Not run:
library(faahK0)
cdfpath <- system.file('cdf', package = 'faahK0')
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**

```
## Not run:
library(faahK0)
cdfpath <- system.file("cdf", package = "faahK0")
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
xset <- xcmsSet(cdffiles)
xset <- group(xset)
xset2 <- retcor(xset, family = "symmetric", plotype = "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**

```
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 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

```
## Not run:
library(faahK0)
cdfpath <- system.file("cdf", package = "faahK0")
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 filtered data

## Examples

```
## Not run:
library(faahK0)
cdfpath <- system.file("cdf", package = "faahK0")
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	<i>Principal component analysis(PCA) for SVA corrected data and raw data</i>
--------	--

---

### Description

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

### Usage

```
svapca(list, center = TRUE, scale = TRUE, 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

```
## Not run:
library(faahK0)
cdfpath <- system.file("cdf", package = "faahK0")
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)
xset <- xcmsSet(cdffiles)
xset <- group(xset)
xset2 <- retcor(xset, family = "symmetric", plotype = "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**

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

**Arguments**

<code>list</code>	results from svacor function
<code>pqvalues</code>	method for ANOVA or SVA
<code>pt</code>	threshold for p value, default is 0.05
<code>qt</code>	threshold for q value, default is 0.05
<code>lv</code>	group information
<code>index</code>	index for selected peaks

**Value**

heatmap for the data

**See Also**

[svacor](#), [svapca](#), [svabatch](#)

**Examples**

```
## Not run:
library(faahK0)
cdfpath <- system.file("cdf", package = "faahK0")
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**

<code>xset</code>	xcmsset object
<code>lv</code>	group information

**Value**

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

**Examples**

```
## Not run:  
library(faahK0)  
cdfpath <- system.file("cdf", package = "faahK0")  
cdffiles <- list.files(cdfpath, recursive = TRUE, full.names = TRUE)  
xset <- xcmsSet(cdffiles)  
xset <- group(xset)  
xset2 <- retcor(xset, family = "symmetric", plotype = "mdevden")  
xset2 <- group(xset2, bw = 10)  
xset3 <- fillPeaks(xset2)  
svaupload(xset3)  
  
## End(Not run)
```

---

`TBBPA`*Demo data for TBBPA metabolism in Pumpkin*

---

**Description**

Demo data for TBBPA metabolism in Pumpkin

**Usage**

```
data(TBBPA)
```

**Format**

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

**References**

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

---

writeMSP

*Write MSP file for NIST search*

---

**Description**

Write MSP file for NIST search

**Usage**

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

**Arguments**

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

**Value**

none a MSP file will be created.

**Examples**

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

---

`xrankanno`*Perform MS/MS X rank annotation for mgf file*

---

**Description**

Perform MS/MS X rank annotation for mgf file

**Usage**

```
xrankanno(file, db = NULL, ppm = 10, prems = 1.1, intc = 0.1, quantile = 0.75)
```

**Arguments**

<code>file</code>	mgf file generated from MS/MS data
<code>db</code>	database could be list object from 'getms2pmd'
<code>ppm</code>	mass accuracy, default 10
<code>prems</code>	precursor mass range, default 1.1 to include M+H or M-H
<code>intc</code>	intensity cutoff for peaks. Default 0.1
<code>quantile</code>	X rank quantiles cutoff for annotation. Default 0.75.

**Value**

list with MSMS annotation results

# Index

## \* datasets

- list, 46
- sccp, 67
- TBBPA, 73

batch, 4

cbmd, 4

dotpanno, 5

findline, 6

findlipid, 6

findmet, 7

findohc, 8

findpfc, 9

getalign, 10

getalign2, 10

getarea, 11, 12, 39

getareastd, 11, 12, 39

getbgremove, 12

getbiotechrep, 13

getcompare, 14

getcsv, 15, 22, 32

getdata, 16, 18, 19, 22, 24, 30, 32, 36, 42

getdata2, 17, 17, 19, 22, 24, 32, 33, 36, 43

getdoe, 18, 24, 30, 32–34, 36

getdwtus, 19

getfeaturesanova, 20

getfeaturest, 21

getfilter, 21, 32

getformula, 22

getgrouprep, 23

getimputation, 19, 22, 24, 34, 36

GetIntegration, 24

Getisotopologues, 25

getmass, 26

getmassdefect, 27, 53

getmd, 27

getmdh, 28

getmdr, 29

getmr, 19, 22, 24, 29, 34, 36

getms1anno, 30

getMSP, 31

getmzrt, 17–19, 22, 24, 30, 31, 33, 34, 36, 42, 44

getmzrt2, 33, 43, 44

getmzrtcsv, 33, 34

getoverlappeak, 34

getpn, 35

getpower, 19, 35

getpqi, 36

getQCraw, 36

getrangecsv, 37

getretcor, 37

getrmd, 38

getsccp, 11, 12, 39

getsim, 40

gettechrep, 40

gettimegrouprep, 41

getupload, 30, 42, 43

getupload2, 33, 43

getupload3, 43

gifmr, 44

Integration, 45

list, 46

ma, 46

Mode, 47

plotanno, 47

plotcc, 48

plotden, 48

plotdwtus, 49

plote, 49

plotgroup, 50

plothist, 51

plotmh, 51

plotint, 52  
plotintslope, 53  
plotkms, 27, 53  
plotmr, 54  
plotmrc, 55  
plotms, 55  
plotmsrt, 56  
plotmz, 57  
plotpca, 57  
plotpeak, 58  
plotridge, 59  
plotridges, 60  
plotrla, 60  
plotrsd, 61  
plotrtms, 62  
plotrug, 62  
plotsms, 63  
plotsub, 64  
plott, 64  
plottic, 65

qbatch, 65

runMDPlot, 66  
runscpp, 66

sccp, 67  
submd, 67  
svabatch, 68, 69, 71, 72  
svacor, 68, 69, 71, 72  
svadata, 70  
svapca, 68, 69, 71, 72  
svaplot, 68, 69, 71, 72  
svaupload, 73

TBBPA, 73

writeMSP, 74

xrankanno, 75