Package 'metaX'

April 23, 2016

Type Package

Title An R package for metabolomic data analysis

Version 1.0.3

Date 2015-02-10

Author Bo Wen <wenbo@genomics.cn>

Maintainer Bo Wen <wenbo@genomics.cn>

Description The package provides a integrated pipeline for mass spectrometry-based metabolomic data analysis. It includes the stages peak detection, data preprocessing, normalization, missing value imputation, univariate statistical analysis, multivariate statistical analysis such as PCA and PLS-DA, metabolite identification, pathway analysis, power analysis, feature selection and modeling, data quality assessment.

Depends R (>= 3.2.0),VennDiagram,pROC,SSPA,methods

Imports Nozzle.R1, ggplot2, parallel, pcaMethods, reshape2, plyr, BBmisc, mixOmics, preprocessCore, vsn, pls, impute, missForest, doParallel, DiscriMiner, xcms, ape, scatterplot3d, pheatmap, bootstrap, boot, caret, dplyr, stringr, RColorBrewer, DiffCorr, RCurl, lattice, faahKO, data.table, CAMERA, igraph

License LGPL-2

Suggests knitr, BiocStyle, R.utils, RUnit,BiocGenerics

VignetteBuilder knitr

biocViews Metabolomics, MassSpectrometry, QualityControl

NeedsCompilation no

PackageStatus Deprecated

R topics documented:

`addIdentInfo` *Add identification result into metaXpara object*

Description

Add identification result into metaXpara object

Usage

```
addIdentInfo(para, file, ...)

## S4 method for signature 'metaXpara,character'
addIdentInfo(para, file, ...)
```

Arguments

`para`	A metaXpara object.
`file`	The file name which contains the identification result
`...`	Other argument

Value

A metaXpara object.

Methods (by class)

- `para = metaXpara,file = character`:

Author(s)

Bo Wen <wenbo@genomics.cn>

addValueNorm<- *addValueNorm*

Description

addValueNorm

Usage

```
addValueNorm(para) <- value
```

Arguments

para	An object of metaXpara
value	An object of metaXpara

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
addValueNorm(para) <- para
```

autoRemoveOutlier *Automatically detect outlier samples*

Description

Automatically detect outlier samples

Usage

```
autoRemoveOutlier(para, outTol = 1.2, pcaMethod = "svdImpute",
  valueID = "valueNorm", scale = "none", center = FALSE, ...)

## S4 method for signature 'metaXpara'
autoRemoveOutlier(para, outTol = 1.2,
  pcaMethod = "svdImpute", valueID = "valueNorm", scale = "none",
  center = FALSE, ...)
```

Arguments

para	A metaXpara object
outTol	A factor to define the outlier tolerance, default is 1.2
pcaMethod	See pca in **pcaMethods**
valueID	The name of the column which will be used
scale	Scaling, see pca in **pcaMethods**
center	Centering, see pca in **pcaMethods**
...	Additional parameter

Value

The name of outlier samples

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
rs <- autoRemoveOutlier(para,valueID="value")
```

bootPLSDA *Fit predictive models for PLS-DA*

Description

Fit predictive models for PLS-DA

Usage

```
bootPLSDA(x, y, ncomp = 2, sample = NULL, test = NULL, split = 0,
  method = "repeatedcv", repeats = 250, number = 7, ...)
```

Arguments

x	An object where samples are in rows and features are in columns. This could be a simple matrix, data frame.
y	A numeric or factor vector containing the outcome for each sample.
ncomp	The maximal number of component for PLS-DA
sample	A vector contains the sample used for the model
test	The data set (data.frame) for testing. If the data contains a column with the name "class", this column is the sample class.
split	Whether split the data as train and test set. Default is 0 which indicates not split the data.
method	The resampling method: boot, boot632, cv, repeatedcv, LOOCV, LGOCV (for repeated training/test splits), none (only fits one model to the entire training set), oob (only for random forest, bagged trees, bagged earth, bagged flexible discriminant analysis, or conditional tree forest models), "adaptive_cv", "adaptive_boot" or "adaptive_LGOCV"
repeats	For repeated k-fold cross-validation only: the number of complete sets of folds to compute
number	Either the number of folds or number of resampling iterations
...	Arguments passed to the classification or regression routine

Value

A list object

calcAUROC *Classical univariate ROC analysis*

Description

Classical univariate ROC analysis

Usage

```
calcAUROC(x, y, cgroup, plot, ...)

## S4 method for signature 'numeric'
calcAUROC(x, y, cgroup, plot, ...)
```

Arguments

x	A numeric vector
y	A response vector
cgroup	Sample class used
plot	A logical indicates whether plot
...	Additional parameter

Value

A data.frame

Methods (by class)

- numeric:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
x <- rnorm(50,2,2)
y<- rep(c("c","t"),25)
calcAUROC(x,y)
```

calcVIP *Calculate the VIP for PLS-DA*

Description

Calculate the VIP for PLS-DA

Usage

```
calcVIP(x, ncomp, ...)

## S4 method for signature 'ANY'
calcVIP(x, ncomp, ...)
```

Arguments

x	An object of output from plsr
ncomp	The number of component used in PLS-DA
...	Additional parameters

Value

An vector of VIP value

Methods (by class)

- ANY:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(pls)
x <- matrix(rnorm(1000),nrow = 10,ncol = 100)
y <- rep(0:1,5)
res <- plsr(y~x)
calcVIP(res,2)
```

center<- *center*

Description

center

Usage

```
center(para) <- value
```

Arguments

para	An object of plsDAPara
value	value

Value

An object of plsDAPara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("plsDAPara")
center(para) <- TRUE
```

cor.network *Correlation network analysis*

Description

Correlation network analysis

Usage

```
cor.network(para, group, valueID = "value", cor.method = "spearman",
  threshold = 0.1, p.adjust.methods = "BH")
```

Arguments

para	A metaXpara object
group	Samples used for plot
valueID	The name of the column that used for plot
cor.method	Method for correlation:"pearson","spearman" or "kendall"
threshold	A threshold of significance levels of differential correlation
p.adjust.methods	c("local", holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none")
...	Additional parameter

Value

The name of result file

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
resfile <- cor.network(para,group=c("S","C"))
```

---

createModels *Create predictive models*

Description

Create predictive models

Usage

```
createModels(para, method = "plsda", group = NA, valueID = "value", ...)
```

Arguments

para	An object of metaXpara
method	Method for model construction
group	Sample class used
valueID	The name of column used
...	Additional arguments.

Value

A list object

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
para <- transformation(para,method = 1,valueID = "value")
para <- metaX::preProcess(para,scale = "uv",center = TRUE,
                          valueID = "value")
rs <- createModels(para,method="plsda",group=c("S","C"),valueID="value")
```

---

dataClean *dataClean*

Description

dataClean

Usage

```
dataClean(para, valueID = "value", sd.factor = 3, snr = 1, ...)

## S4 method for signature 'metaXpara'
dataClean(para, valueID = "value", sd.factor = 3,
  snr = 1, ...)
```

Arguments

para	A metaXpara object.
valueID	The name of the column used
sd.factor	The factor used to filter peak based on SD
snr	The threshold to filter peak
...	Other argument

Value

A metaXpara object.

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
p <- dataClean(para)
```

`dir.case<-` *dir.case*

Description

dir.case

Usage

```
dir.case(para) <- value
```

Arguments

`para`	An object of metaXpara
`value`	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
dir.case(para) <- "./"
```

dir.ctrl<- *dir.ctrl*

Description

dir.ctrl

Usage

```
dir.ctrl(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
dir.ctrl(para) <- "./"
```

doQCRLSC *Using the QC samples to do the quality control-robust spline signal correction*

Description

Using the QC samples to do the quality control-robust spline signal correction.

Usage

```
doQCRLSC(para, cvFilter = 0.3, impute = TRUE, cpu = 0, ...)

## S4 method for signature 'metaXpara'
doQCRLSC(para, cvFilter = 0.3, impute = TRUE,
  cpu = 0, ...)
```

Arguments

para	An object of metaXpara
cvFilter	The threshold of CV filter
impute	A logical indicates whether impute the result
cpu	The number of cpu used for processing
...	Additional parameters

Details

The smoothing parameter is optimised using leave-one-out cross validation to avoid overfitting.

Value

A list object

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

See Also

plotQCRLSC

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)[1:20,]
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
res <- doQCRLSC(para,cpu=1)
```

featureSelection *Feature selection and modeling*

Description

Feature selection and modeling

Usage

```
featureSelection(para, group, method = "rf", valueID = "value", fold = 5,
  repeats = 10, verbose = FALSE, ...)
```

Arguments

para	An object of metaXpara
group	The sample class used
method	Method for feature selection and modeling
valueID	The column name used
fold	k-fold
repeats	The repeat number
verbose	Whether output or not
...	Additional parameters

Value

The result of feature selection and modeling

filterPeaks *filterPeaks*

Description

filter peaks according to the non-QC sample

Usage

```
filterPeaks(para, ratio = 0.8, omit.negative = TRUE, ...)

## S4 method for signature 'metaXpara'
filterPeaks(para, ratio = 0.8, omit.negative = TRUE,
  ...)
```

Arguments

para	An object of metaXpara
ratio	filter peaks which have missing value more than percent of "ratio", default is 0.8
omit.negative	A logical value indicates whether omit the negative value
...	Additional parameters

Value

An object of metaXpara

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
p <- filterPeaks(para,ratio=0.2)
```

---

filterQCPeaks *filterQCPeaks*

Description

filter peaks according to the QC sample

Usage

```
filterQCPeaks(para, ratio = 0.5, omit.negative = TRUE, ...)

## S4 method for signature 'metaXpara'
filterQCPeaks(para, ratio = 0.5, omit.negative = TRUE,
  ...)
```

Arguments

para	An object of metaXpara
ratio	filter peaks which have missing value more than percent of "ratio", default is 0.5
omit.negative	A logical value indicates whether omit the negative value
...	Additional parameters

Value

An object of metaXpara

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
p <- filterQCPeaks(para,ratio=0.5)
```

---

filterQCPeaksByCV *Filter peaks according to the RSD of peaks in QC samples*

Description

Filter peaks according to the RSD of peaks in QC samples. Usually used after missing value imputation.

Usage

```
filterQCPeaksByCV(para, cvFilter = 0.3, valueID = "value", ...)

## S4 method for signature 'metaXpara'
filterQCPeaksByCV(para, cvFilter = 0.3,
  valueID = "value", ...)
```

Arguments

para	An object of metaXpara
cvFilter	Filter peaks with the RSD in QC samples > cvFilter.
valueID	The name of the column which will be used.
...	Additional parameter

Value

An object of metaXpara

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
p <- filterQCPeaksByCV(para)
```

---

getPeaksTable *Get a data.frame which contained the peaksData in metaXpara*

Description

Get a data.frame which contained the peaksData in metaXpara

Usage

```
getPeaksTable(para, sample = NULL, valueID = "value")

## S4 method for signature 'metaXpara'
getPeaksTable(para, sample = NULL, valueID = "value")
```

Arguments

para	An object of data
sample	Sample class used
valueID	The column name used

Value

A data.frame

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
res <- getPeaksTable(para)
```

---

group.bw0<- *group.bw0*

Description

group.bw0

Usage

```
group.bw0(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
group.bw0(para) <- 10
```

group.bw<- *group.bw*

Description

group.bw

Usage

```
group.bw(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
group.bw(para) <- 5
```

group.max<- *group.max*

Description

group.max

Usage

```
group.max(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
group.max(para) <- 1000
```

`group.minfrac<-` *group.minfrac*

Description

group.minfrac

Usage

```
group.minfrac(para) <- value
```

Arguments

`para`	An object of metaXpara
`value`	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
group.minfrac(para) <- 0.3
```

group.minsamp<- *group.minsamp*

Description

group.minsamp

Usage

```
group.minsamp(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
group.minsamp(para) <- 1
```

group.mzwid0<- *group.mzwid0*

Description

group.mzwid0

Usage

```
group.mzwid0(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
group.mzwid0(para) <- 0.015
```

`group.mzwid<-` *group.mzwid*

Description

group.mzwid

Usage

```
group.mzwid(para) <- value
```

Arguments

`para`	An object of metaXpara
`value`	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
group.mzwid(para) <- 0.015
```

group.sleep<- *group.sleep*

Description

group.sleep

Usage

```
group.sleep(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
group.sleep(para) <- 0
```

hasQC *Judge whether the data has QC samples*

Description

Judge whether the data has QC samples

Usage

```
hasQC(para, ...)

## S4 method for signature 'metaXpara'
hasQC(para, ...)
```

Arguments

para	An object of data
...	Additional parameters

Value

A logical value

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
hasQC(para)
```

---

idres<- *idres*

Description

idres

Usage

```
idres(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
idres(para) <- data.frame()
```

---

kfold<- *kfold*

Description

kfold

Usage

```
kfold(para) <- value
```

Arguments

para	An object of plsDAPara
value	value

Value

An object of plsDAPara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("plsDAPara")
kfold(para) <- 5
```

makeDirectory *Create directory*

Description

Create directory

Usage

```
makeDirectory(para)

## S4 method for signature 'metaXpara'
makeDirectory(para)
```

Arguments

para	A metaXpara object

Value

none

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
outdir(para) <- "outdir"
makeDirectory(para)
```

makeMetaboAnalystInput *Export a csv file which can be used for MetaboAnalyst*

Description

Export a csv file which can be used for MetaboAnalyst

Usage

```
makeMetaboAnalystInput(para, rmQC = TRUE, valueID = "valueNorm",
  zero2NA = TRUE, prefix = NA, ...)

## S4 method for signature 'metaXpara'
makeMetaboAnalystInput(para, rmQC = TRUE,
  valueID = "valueNorm", zero2NA = TRUE, prefix = NA, ...)
```

Arguments

para	A metaXpara object
rmQC	A logical indicates whether remove the QC data
valueID	The name of the column which will be used
zero2NA	A logical indicates whether convert the value <=0 to NA
prefix	The prefix of output file
...	Additional parameter

Value

none

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
makeMetaboAnalystInput(para,valueID="value")
```

metaboliteAnnotation *Metabolite identification*

Description

Metabolite identification

Usage

```
metaboliteAnnotation(para, db, delta, mode, unit)
```

Arguments

para	An object of metaXpara
db	The file name of database
delta	The delta
mode	The mode of data, positive or negative
unit	The unit of the delta

Value

The name of output file

metaXpara-class *An S4 class to represent the parameters and data for data processing*

Description

An S4 class to represent the parameters and data for data processing

Usage

```
## S4 replacement method for signature 'metaXpara'
dir.case(para) <- value

## S4 replacement method for signature 'metaXpara'
dir.ctrl(para) <- value

## S4 replacement method for signature 'metaXpara'
sampleListFile(para) <- value

## S4 replacement method for signature 'metaXpara'
ratioPairs(para) <- value
```

```
## S4 replacement method for signature 'metaXpara'
missValueImputeMethod(para) <- value

## S4 replacement method for signature 'metaXpara'
outdir(para) <- value

## S4 replacement method for signature 'metaXpara'
prefix(para) <- value

## S4 replacement method for signature 'metaXpara'
peaksData(para) <- value

## S4 replacement method for signature 'metaXpara,metaXpara'
addValueNorm(para) <- value

## S4 replacement method for signature 'metaXpara'
rawPeaks(para) <- value

## S4 replacement method for signature 'metaXpara'
idres(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.method(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.ppm(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.peakwidth(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.snthresh(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.prefilter(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.mzCenterFun(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.integrate(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.mzdiff(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.noise(para) <- value
```

```
## S4 replacement method for signature 'metaXpara'
xcmsSet.verbose.columns(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.polarity(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.profparam(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.nSlaves(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.fitgauss(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.sleep(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.fwhm(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.max(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSet.step(para) <- value

## S4 replacement method for signature 'metaXpara'
group.bw0(para) <- value

## S4 replacement method for signature 'metaXpara'
group.mzwid0(para) <- value

## S4 replacement method for signature 'metaXpara'
group.bw(para) <- value

## S4 replacement method for signature 'metaXpara'
group.mzwid(para) <- value

## S4 replacement method for signature 'metaXpara'
group.minfrac(para) <- value

## S4 replacement method for signature 'metaXpara'
group.minsamp(para) <- value

## S4 replacement method for signature 'metaXpara'
group.max(para) <- value
```

```
## S4 replacement method for signature 'metaXpara'
group.sleep(para) <- value

## S4 replacement method for signature 'metaXpara'
retcor.method(para) <- value

## S4 replacement method for signature 'metaXpara'
retcor.profStep(para) <- value

## S4 replacement method for signature 'metaXpara'
retcor.plottype(para) <- value

## S4 replacement method for signature 'metaXpara'
qcRlscSpan(para) <- value

## S4 replacement method for signature 'metaXpara'
xcmsSetObj(para) <- value
```

Arguments

para	A metaXpara object
value	New value

Value

A object of metaXpara

Methods (by generic)

- dir.case<-:
- dir.ctrl<-:
- sampleListFile<-:
- ratioPairs<-:
- missValueImputeMethod<-:
- outdir<-:
- prefix<-:
- peaksData<-:
- addValueNorm<-:
- rawPeaks<-:
- idres<-:
- xcmsSet.method<-:
- xcmsSet.ppm<-:
- xcmsSet.peakwidth<-:
- xcmsSet.snthresh<-:

- xcmsSet.prefilter<-:
- xcmsSet.mzCenterFun<-:
- xcmsSet.integrate<-:
- xcmsSet.mzdiff<-:
- xcmsSet.noise<-:
- xcmsSet.verbose.columns<-:
- xcmsSet.polarity<-:
- xcmsSet.profparam<-:
- xcmsSet.nSlaves<-:
- xcmsSet.fitgauss<-:
- xcmsSet.sleep<-:
- xcmsSet.fwhm<-:
- xcmsSet.max<-:
- xcmsSet.step<-:
- group.bw0<-:
- group.mzwid0<-:
- group.bw<-:
- group.mzwid<-:
- group.minfrac<-:
- group.minsamp<-:
- group.max<-:
- group.sleep<-:
- retcor.method<-:
- retcor.profStep<-:
- retcor.plottype<-:
- qcRlscSpan<-:
- xcmsSetObj<-:

Slots

`dir.case` The path names of the NetCDF/mzXML files to read

`dir.ctrl` The path names of the NetCDF/mzXML files to read

`sampleListFile` The file name of containing the experiment design

`sampleList` A data.frame containing the experiment design

`ratioPairs` A character containing the ratio pairs, such as "A:B;A:C"

`missValueImputeMethod` A character of missing value imputation method

`sampleListHead` The name of head of `sampleListFile`

`outdir` The output directory

`prefix` The prefix of output file

`xcmsPeakListFile` The file of output from **XCMS**

`fig` A `list` of file names of figures

`peaksData` A `data.frame` containing the peaks data

`VIP` A `data.frame` containing the VIP

`rawPeaks` A `data.frame` containg the raw peaks data

`xcmsSetObj` An object of `xcmsSet`

`quant` A `data.frame` containing the quantification result

`idres` A `data.frame` containing the identification result

`xcmsSet.method` Method to use for peak detection. See details `findPeaks` in package **XCMS**

`xcmsSet.ppm` The maxmial tolerated m/z deviation in consecutive scans, in ppm (parts per million)

`xcmsSet.peakwidth` Chromatographic peak width, given as range (min,max) in seconds

`xcmsSet.snthresh` The signal to noise ratio cutoff, definition see `findPeaks.centWave`

`xcmsSet.prefilter` prefilter=c(k,I), see `findPeaks.centWave`

`xcmsSet.mzCenterFun` See `findPeaks.centWave`

`xcmsSet.integrate` See `findPeaks.centWave`

`xcmsSet.mzdiff` See `findPeaks.centWave`

`xcmsSet.noise` See `findPeaks.centWave`

`xcmsSet.verbose.columns` See `findPeaks.centWave`

`xcmsSet.polarity` Filter raw data for positive/negative scans. See `xcmsSet`

`xcmsSet.profparam` Parameters to use for profile generation. See `xcmsSet`

`xcmsSet.nSlaves` The number of slaves/cores to be used for parallel peak detection. See `xcmsSet`

`xcmsSet.fitgauss` See `findPeaks.centWave`

`xcmsSet.sleep` The number of seconds to pause between plotting peak finding cycles. See `findPeaks.centWave`

`xcmsSet.fwhm` See `findPeaks.matchedFilter`

`xcmsSet.max` See `findPeaks.matchedFilter`

`xcmsSet.step` See `findPeaks.matchedFilter`

`group.bw0` See `group.density`

`group.mzwid0` See `group.density`

`group.bw` See `group.density`

`group.mzwid` See `group.density`

`group.minfrac` See `group.density`

`group.minsamp` See `group.density`

`group.max` See `group.density`

`group.sleep` See `group.density`

`retcor.method` See `retcor`

`retcor.profStep` See `retcor.obiwarp`

`retcor.plottype` See `retcor.obiwarp`

`qcRlscSpan` The value of span for QC-RLSC

Author(s)

Bo Wen <wenbo@genomics.cn>

`metaXpipe` *metaXpipe*

Description

metaXpipe

Usage

```
metaXpipe(para, plsdaPara, cvFilter = 0.3, remveOutlier = TRUE,
  outTol = 1.2, doQA = TRUE, doROC = TRUE, qcsc = FALSE,
  nor.method = "pqn", pclean = TRUE, t = 1, scale = "uv",
  idres = NULL, nor.order = 1, out.rmqc = FALSE, saveRds = FALSE, ...)

## S4 method for signature 'metaXpara'
metaXpipe(para, plsdaPara, cvFilter = 0.3,
  remveOutlier = TRUE, outTol = 1.2, doQA = TRUE, doROC = TRUE,
  qcsc = FALSE, nor.method = "pqn", pclean = TRUE, t = 1,
  scale = "uv", idres = NULL, nor.order = 1, out.rmqc = FALSE,
  saveRds = FALSE, ...)
```

Arguments

`para`	A metaXpara object.
`plsdaPara`	A plsDAPara object.
`cvFilter`	Filter peaks which cv > cvFilter in QC samples.
`remveOutlier`	Remove outlier samples.
`outTol`	The threshold to remove outlier samples.
`doQA`	Boolean, setting the argument to TRUE will perform plot quality figures.
`doROC`	A logical indicates whether to calculate the ROC
`qcsc`	Boolean, setting the argument to TRUE to perform quality control-robust loess signal correction.
`nor.method`	Normalization method.
`pclean`	Boolean, setting the argument to TRUE to perform data cleaning
`t`	Data transformation method. See `transformation`.
`scale`	Data scaling method.
`idres`	A file containing the metabolite identification result
`nor.order`	The order of normalization, only valid when `qcsc` is TRUE. 1: before QC-RLSC, 2: after QC-RLSC.

out.rmqc	Boolean, setting the argument to TRUE to remove the QC samples for the csv file.
saveRds	Boolean, setting the argument to TRUE to save some objects to disk for debug. Only useful for developer. Default is FALSE.
...	Other argument

Value

A metaXpara object.

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
## Not run:
## example 1: no QC sample
library(faahKO)
xset <- group(faahko)
xset <- retcor(xset)
xset <- group(xset)
xset <- fillPeaks(xset)
peaksData <- as.data.frame(groupval(xset,"medret",value="into"))
peaksData$name <- row.names(peaksData)
para <- new("metaXpara")
rawPeaks(para) <- peaksData
ratioPairs(para) <- "KO:WT"
outdir(para) <- "test"
sampleListFile(para) <- system.file("extdata/faahKO_sampleList.txt",
    package = "metaX")
plsdaPara <- new("plsDAPara")
p <- metaXpipe(para,plsdaPara=plsdaPara)

## example 2: has QC samples
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
ratioPairs(para) <- "S:C"
plsdaPara <- new("plsDAPara")
p <- metaXpipe(para,plsdaPara=plsdaPara)

## End(Not run)
```

method<- *method*

Description

method

Usage

```
method(para) <- value
```

Arguments

para	An object of plsDAPara
value	value

Value

An object of plsDAPara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("plsDAPara")
method(para) <- "oscorespls"
```

missingValueImpute *Missing value imputation*

Description

Missing value imputation

Usage

```
missingValueImpute(x, valueID = "value", method = "knn", negValue = TRUE,
  cpu = 1, ...)

## S4 method for signature 'metaXpara'
missingValueImpute(x, valueID = "value",
  method = "knn", negValue = TRUE, cpu = 1, ...)

## S4 method for signature 'data.frame'
missingValueImpute(x, valueID = "value",
  method = "knn", negValue = TRUE, cpu = 1, ...)
```

Arguments

x	The value needed to be imputated
valueID	The name of the column which will be used
method	Method for imputation: bpca,knn,svdImpute,rf,min
negValue	A logical indicates whether convert <=0 value to NA
cpu	The number of cpus used
...	Additional parameters

Value

The imputation data

Methods (by class)

- metaXpara:
- data.frame:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
```

---

missValueImputeMethod<- *missValueImputeMethod*

Description

missValueImputeMethod

Usage

```
missValueImputeMethod(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
missValueImputeMethod(para) <- "knn"
```

`myCalcAUROC` *Classical univariate ROC analysis*

Description

Classical univariate ROC analysis

Usage

```
myCalcAUROC(para, cgroup, cpu = 0, plot = FALSE, ...)
```

Arguments

`para`	A metaXpara object
`cgroup`	Samples used
`cpu`	The number of CPU used
`plot`	A logical indicates whether plot
`...`	Additional parameter

Value

A metaXpara object

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
## Not run:
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
addValueNorm(para) <- para
res <- myCalcAUROC(para,cgroup=c("S","C"))

## End(Not run)
```

---

myPLSDA *Perform PLS-DA analysis*

Description

Perform PLS-DA analysis

Usage

```
myPLSDA(x, y, save, select, ...)

## S4 method for signature 'ANY,ANY,logical,ANY'
myPLSDA(x, y, ncomp = 10, validation = "CV",
  k = 7, method = "oscorespls", save = TRUE)

## S4 method for signature 'ANY,ANY,ANY,ANY'
myPLSDA(x, y, ncomp = 10, validation = "CV",
  k = 7, method = "oscorespls")

## S4 method for signature 'ANY,ANY,missing,logical'
myPLSDA(x, y, ncomp = 10,
  validation = "CV", k = 7, method = "oscorespls", select = TRUE)
```

Arguments

x	A matrix of observations
y	a vector or matrix of responses
save	A logical indicates whether save the plsr result
select	A logical indicates whether select the best component
...	Additional parameters
ncomp	The number of component used for PLS-DA
validation	See plsr

k	k-fold
method	See plsr

Value

The PLS-DA result

Methods (by class)

- x = ANY,y = ANY,save = logical,select = ANY:
- x = ANY,y = ANY,save = ANY,select = ANY:
- x = ANY,y = ANY,save = missing,select = logical:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
x <- matrix(rnorm(1000),nrow = 10,ncol = 100)
y <- rep(0:1,5)
res <- myPLSDA(x,y,save=TRUE,ncomp=2,validation="CV",k=7,
    method="oscorespls")
```

---

ncomp<- *ncomp*

Description

ncomp

Usage

```
ncomp(para) <- value
```

Arguments

para	An object of plsDAPara
value	value

Value

An object of plsDAPara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("plsDAPara")
ncomp(para) <- 5
```

`normalize` *Normalisation of peak intensity*

Description

The normalize method performs normalisation on peak intensities.

Usage

```
normalize(para, method = "sum", valueID = "value", ...)

## S4 method for signature 'metaXpara'
normalize(para, method = "sum", valueID = "value",
  ...)
```

Arguments

`para`	A metaXpara object.
`method`	The normalization method: sum, vsn, quantiles, quantiles.robust, sum, pqn. Default is sum.
`valueID`	The name of the column which will be normalized.
`...`	Additional parameter

Value

A metaXpara object.

Methods (by class)

- `metaXpara`:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
xset <- group(faahko)
xset <- retcor(xset)
xset <- group(xset)
xset <- fillPeaks(xset)
peaksData <- as.data.frame(groupval(xset,"medret",value="into"))
peaksData$name <- row.names(peaksData)
para <- new("metaXpara")
rawPeaks(para) <- peaksData
sampleListFile(para) <- system.file("extdata/faahKO_sampleList.txt",
    package = "metaX")
para <- reSetPeaksData(para)
para <- metaX::normalize(para)
```

---

nperm<- *nperm*

Description

nperm

Usage

```
nperm(para) <- value
```

Arguments

para	An object of plsDAPara
value	value

Value

An object of plsDAPara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("plsDAPara")
nperm(para) <- 1000
```

outdir<- *outdir*

Description

outdir

Usage

```
outdir(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
outdir(para) <- "outdir"
```

pathwayAnalysis *Pathway analysis*

Description

Pathway analysis

Usage

```
pathwayAnalysis(id, id.type = "hmdb", outfile)
```

Arguments

id	A vector of metabolite IDs
id.type	The type of metabolite ID type, default is hmdb.
outfile	The output file name

Value

A data.frame object

Examples

```
## Not run:
res <- pathwayAnalysis(id=c("HMDB00060","HMDB00056","HMDB00064"),
    outfile="pathway.csv")
head(res)

## End(Not run)
```

---

peakFinder *Peak detection by using XCMS package*

Description

peakFinder takes a set of MS sample data and performs a peak detection, retention time correction and peak grouping steps using XCMS package.

Usage

```
peakFinder(para, ...)

## S4 method for signature 'metaXpara'
peakFinder(para, ...)
```

Arguments

para	A metaXpara object
...	Additional parameter

Value

A metaXpara object

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
## Not run:
library(faahKO)
para <- new("metaXpara")
dir.case(para) <- system.file("cdf/KO", package = "faahKO")
dir.ctrl(para) <- system.file("cdf/WT", package = "faahKO")
## set parameters for peak picking
xcmsSet.peakwidth(para) <- c(20,50)
xcmsSet.snthresh(para) <- 10
xcmsSet.prefilter(para) <- c(3,100)
xcmsSet.noise(para) <- 0
xcmsSet.nSlaves(para) <- 4
## run peak picking
p <- peakFinder(para)

## End(Not run)
```

---

peaksData<- *peaksData*

Description

peaksData

Usage

```
peaksData(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
peaksData(para) <- data.frame()
```

peakStat *Do the univariate and multivariate statistical analysis*

Description

Do the univariate and multivariate statistical analysis

Usage

```
peakStat(para, plsdaPara, doROC = TRUE, saveRds = FALSE, ...)

## S4 method for signature 'metaXpara,plsDAPara'
peakStat(para, plsdaPara, doROC = TRUE,
  saveRds = FALSE, ...)
```

Arguments

para	A metaXpara object
plsdaPara	A plsDAPara object
doROC	A logical indicates whether to calculate the ROC
saveRds	Boolean, setting the argument to TRUE to save some objects to disk for debug. Only useful for developer. Default is FALSE.
...	Additional parameter

Value

none

An object of metaXpara

Methods (by class)

- para = metaXpara,plsdaPara = plsDAPara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
## Not run:
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
```

```
ratioPairs(para) <- "S:C"
addValueNorm(para) <- para
plsdaPara <- new("plsDAPara")
res <- peakStat(para,plsdaPara)

## End(Not run)
```

---

permutePLSDA *permutePLSDA*

Description

Validation of the PLS-DA model by using permutation test statistics

Usage

```
permutePLSDA(x, y, n = 100, np = 2, outdir = "./", prefix = "metaX",
  tol = 0.001, cpu = 0, ...)
```

Arguments

x	a matrix of observations.
y	a vector or matrix of responses.
n	number of permutations to compute the PLD-DA p-value based on R2 magnitude. Default n=100
np	the number of components to be used in the modelling.
outdir	output dir
prefix	the prefix of output figure file
tol	tolerance value based on maximum change of cumulative R-squared coefficient for each additional PLS component. Default tol=0.001
cpu	0
...	additional arguments

Value

pvalue

plotCorHeatmap *Plot correlation heatmap*

Description

This function plots correlation heatmap.

Usage

```
plotCorHeatmap(para, valueID = "value", samples = NA, label = "order",
  width = 6, cor.method = "spearman", height = 6, anno = FALSE,
  cluster = FALSE, shownames = FALSE, ...)
```

Arguments

para	A metaXpara object
valueID	The name of the column that used for plot
samples	Samples used for plot
label	Label to show in figure
width	The width of the graphics region in inches. The default values are 6.
cor.method	Method used for correlation
height	The height of the graphics region in inches. The default values are 6.
anno	A logical value indicates whether to plot heatmap with annotating class information
cluster	A logical value indicates whether to do the cluster when anno is TRUE
shownames	A logical indicates whether show names when plot
...	Additional parameter

Value

The fig name

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
plotCorHeatmap(para,valueID="value",samples=NULL,width=6,anno=TRUE)
```

plotCV *Plot the CV distribution of peaks in each group*

Description

Plot the CV distribution of peaks in each group.

Usage

```
plotCV(x, ...)

## S4 method for signature 'metaXpara'
plotCV(x, ...)
```

Arguments

x	A metaXpara object
...	Additional parameter

Value

none

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
xset <- group(faahko)
xset <- retcor(xset)
xset <- group(xset)
xset <- fillPeaks(xset)
peaksData <- as.data.frame(groupval(xset,"medret",value="into"))
peaksData$name <- row.names(peaksData)
para <- new("metaXpara")
rawPeaks(para) <- peaksData
sampleListFile(para) <- system.file("extdata/faahKO_sampleList.txt",
    package = "metaX")
para <- reSetPeaksData(para)
plotCV(para)
```

plotHeatMap *Plot heatmap*

Description

This function plots heatmap.

Usage

```
plotHeatMap(para, valueID = "valueNorm", log = TRUE, rmQC = TRUE,
  zero2na = FALSE, colors = "none", width = 12, height = 8,
  saveRds = FALSE, ...)

## S4 method for signature 'metaXpara'
plotHeatMap(para, valueID = "valueNorm", log = TRUE,
  rmQC = TRUE, zero2na = FALSE, colors = "none", width = 12,
  height = 8, saveRds = FALSE, ...)
```

Arguments

para	A metaXpara object
valueID	The name of the column that used for plot
log	A logical indicating whether to log the data
rmQC	A logical indicating whether to remove the QC samples
zero2na	A logical indicating whether to convert the value <=0 to NA
colors	Color for heatmap
width	The width of the graphics region in inches. The default values are 12.
height	The height of the graphics region in inches. The default values are 8.
saveRds	Boolean, setting the argument to TRUE to save some objects to disk for debug. Only useful for developer. Default is FALSE.
...	Additional parameter

Value

none

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
xset <- group(faahko)
xset <- retcor(xset)
xset <- group(xset)
xset <- fillPeaks(xset)
peaksData <- as.data.frame(groupval(xset,"medret",value="into"))
peaksData$name <- row.names(peaksData)
para <- new("metaXpara")
rawPeaks(para) <- peaksData
sampleListFile(para) <- system.file("extdata/faahKO_sampleList.txt",
    package = "metaX")
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
plotHeatMap(para,valueID="value",width=6)
```

`plotIntDistr` *Plot the distribution of the peaks intensity*

Description

Plot the distribution of the peaks intensity for both raw intensity and normalized intensity.

Usage

```
plotIntDistr(x, width = 14, ...)

## S4 method for signature 'metaXpara'
plotIntDistr(x, width = 14, ...)
```

Arguments

`x`	A metaXpara object.
`width`	The width of pdf, default is 14.
`...`	Additional parameter

Value

The figure name

Methods (by class)

- `metaXpara`:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
xset <- group(faahko)
xset <- retcor(xset)
xset <- group(xset)
xset <- fillPeaks(xset)
peaksData <- as.data.frame(groupval(xset,"medret",value="into"))
peaksData$name <- row.names(peaksData)
para <- new("metaXpara")
rawPeaks(para) <- peaksData
sampleListFile(para) <- system.file("extdata/faahKO_sampleList.txt",
    package = "metaX")
para <- reSetPeaksData(para)
plotIntDistr(para)
## after normalization
para <- metaX::normalize(para)
plotIntDistr(para)
```

---

plotMissValue *Plot missing value distribution*

Description

Plot missing value distribution.

Usage

```
plotMissValue(para, width = 8, height = 5, ...)

## S4 method for signature 'metaXpara'
plotMissValue(para, width = 8, height = 5, ...)
```

Arguments

para	A metaXpara object
width	The width of the graphics region in inches. The default values are 8.
height	The height of the graphics region in inches. The default values are 5.
...	Additional parameter

Value

The figure name

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
xset <- group(faahko)
xset <- retcor(xset)
xset <- group(xset)
xset <- fillPeaks(xset)
peaksData <- as.data.frame(groupval(xset,"medret",value="into"))
peaksData$name <- row.names(peaksData)
para <- new("metaXpara")
rawPeaks(para) <- peaksData
sampleListFile(para) <- system.file("extdata/faahKO_sampleList.txt",
    package = "metaX")
para <- reSetPeaksData(para)
plotMissValue(para)
```

---

plotNetwork *Plot correlation network map*

Description

Plot correlation network map

Usage

```
plotNetwork(para, group, valueID = "value", cor.thr = 0.95,
  degree.thr = 10, size.factor = 0.5, layout = layout_in_circle, ...)
```

Arguments

para	A metaXpara object
group	Samples used for plot
valueID	The name of the column that used for plot
cor.thr	Threshold of correlation
degree.thr	Threshold of degree of node
size.factor	Node size factor for plot
layout	layout for plotting
...	Additional parameter

Value

An object of igraph

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
gg <- plotNetwork(para,group=c("S","C"),degree.thr = 10,cor.thr = 0.8)
```

---

plotPCA *Plot PCA figure*

Description

Plot PCA figure

Usage

```
plotPCA(para, pcaMethod = "svdImpute", valueID = "valueNorm",
  label = "order", rmQC = TRUE, batch = FALSE, scale = "none",
  center = FALSE, saveRds = FALSE, ...)

## S4 method for signature 'metaXpara'
plotPCA(para, pcaMethod = "svdImpute",
  valueID = "valueNorm", label = "order", rmQC = TRUE, batch = FALSE,
  scale = "none", center = FALSE, saveRds = FALSE, ...)
```

Arguments

para	A metaXpara object
pcaMethod	See pca in **pcaMethods**
valueID	The name of the column which will be used
label	The label used for plot PCA figure, default is "order"
rmQC	A logical indicates whether remove QC data
batch	A logical indicates whether output batch information
scale	Scaling, see pca in **pcaMethods**
center	Centering, see pca in **pcaMethods**
saveRds	Boolean, setting the argument to TRUE to save some objects to disk for debug. Only useful for developer. Default is FALSE.
...	Additional parameter

Value

none

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
para <- transformation(para,valueID = "value")
metaX::plotPCA(para,valueID="value",scale="uv",center=TRUE)
```

---

plotPeakBox *Plot boxplot for each feature*

Description

Plot boxplot for each feature

Usage

```
plotPeakBox(para, samples, log = FALSE, ...)

## S4 method for signature 'metaXpara'
plotPeakBox(para, samples, log = FALSE, ...)
```

Arguments

para	A metaXpara object
samples	Sample class used
log	Whether log transform or not
...	Additional parameters

Value

The output figure name.

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)[1:20,]
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
addValueNorm(para) <- para
plotPeakBox(para,samples=c("S","C"))
```

---

plotPeakNumber *Plot the distribution of the peaks number*

Description

Plot the distribution of the raw peaks number without post-processing. This function not only generates a figure, but also saves the information of peaks number into a file.

Usage

```
plotPeakNumber(x, ...)

## S4 method for signature 'metaXpara'
plotPeakNumber(x, ...)
```

Arguments

x	A metaXpara object
...	Additional parameter

Value

The figure name

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
xset <- group(faahko)
xset <- retcor(xset)
xset <- group(xset)
xset <- fillPeaks(xset)
peaksData <- as.data.frame(groupval(xset,"medret",value="into"))
peaksData$name <- row.names(peaksData)
para <- new("metaXpara")
rawPeaks(para) <- peaksData
sampleListFile(para) <- system.file("extdata/faahKO_sampleList.txt",
    package = "metaX")
plotPeakNumber(para)
```

---

plotPeakSN *Plot the distribution of the peaks S/N*

Description

Plot the distribution of the peaks S/N, only suitable for XCMS result. This function generates a figure.

Usage

```
plotPeakSN(x, ...)

## S4 method for signature 'metaXpara'
plotPeakSN(x, ...)
```

Arguments

x	A metaXpara object
...	Additional parameter

Value

none

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
xset <- group(faahko)
xset <- retcor(xset)
xset <- group(xset)
xset <- fillPeaks(xset)
para <- new("metaXpara")
xcmsSetObj(para) <- xset
plotPeakSN(para)
```

`plotPeakSumDist` *Plot the total peak intensity distribution*

Description

Plot the total peak intensity distribution

Usage

```
plotPeakSumDist(para, valueID = "value", width = 6, height = 4, ...)

## S4 method for signature 'metaXpara'
plotPeakSumDist(para, valueID = "value", width = 6,
  height = 4, ...)
```

Arguments

`para`	A metaXpara object.
`valueID`	The name of the column used
`width`	Width of the figure
`height`	Height of the figure
`...`	Other argument

Value

The output figure name.

Methods (by class)

- `metaXpara`:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
plotPeakSumDist(para)
```

---

plotPLSDA *Plot PLS-DA figure*

Description

Plot PLS-DA figure

Usage

```
plotPLSDA(para, label = "order", valueID = "valueNorm", ncomp = 5, ...)

## S4 method for signature 'metaXpara'
plotPLSDA(para, label = "order",
  valueID = "valueNorm", ncomp = 5, ...)
```

Arguments

para	A metaXpara object
label	The label used for plot PLS-DA figure, default is "order"
valueID	The name of the column which will be used
ncomp	The number of components used for PLS-DA
...	Additional parameter

Value

none

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
para <- transformation(para,valueID = "value")
para <- preProcess(para=para,scale = "uv",center = TRUE,valueID = "value")
plotPLSDA(para,valueID="value")
```

---

plotQC *Plot the correlation change of the QC samples.*

Description

Plot the correlation change of the QC samples.

Usage

```
plotQC(para, valueID = "valueNorm", step = 4, log = TRUE, width = 8,
  height = 4, ...)

## S4 method for signature 'metaXpara'
plotQC(para, valueID = "valueNorm", step = 4,
  log = TRUE, width = 8, height = 4, ...)
```

Arguments

para	A metaXpara object
valueID	The name of the column that used for plot
step	The step value of calculate the cor of the samples. Default is 4.
log	A logical indicating whether to log the data
width	The width of the graphics region in inches. The default values are 8.
height	The height of the graphics region in inches. The default values are 4.
...	Additional parameter

Value

none

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
plotQC(para,valueID="value")
```

---

plotQCRLSC *Plot figures for QC-RLSC*

Description

Plot figures for QC-RLSC

Usage

```
plotQCRLSC(para)

## S4 method for signature 'metaXpara'
plotQCRLSC(para)
```

Arguments

para	A metaXpara object

Value

none

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

See Also

doQCRLSC

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)[1:20,]
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
res <- doQCRLSC(para,cpu=1)
plotQCRLSC(res$metaXpara)
```

---

plotTreeMap *Plot Phylogenies for samples*

Description

This function plots phylogenetic trees for samples.

Usage

```
plotTreeMap(para, valueID = "valueNorm", log = TRUE, rmQC = TRUE,
  nc = 8, treeType = "fan", width = 8, ...)

## S4 method for signature 'metaXpara'
plotTreeMap(para, valueID = "valueNorm", log = TRUE,
  rmQC = TRUE, nc = 8, treeType = "fan", width = 8, ...)
```

Arguments

para	A metaXpara object
valueID	The name of the column that used for plot
log	A logical indicating whether to log the data
rmQC	A logical indicating whether to remove the QC samples
nc	The number of clusters
treeType	A character string specifying the type of phylogeny to be drawn; it must be one of "phylogram" (the default), "cladogram", "fan", "unrooted", "radial" or any unambiguous abbreviation of these.
width	The width and height of the graphics region in inches. The default values are 8.
...	Additional parameter

Value

none

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
xset <- group(faahko)
xset <- retcor(xset)
xset <- group(xset)
xset <- fillPeaks(xset)
peaksData <- as.data.frame(groupval(xset,"medret",value="into"))
peaksData$name <- row.names(peaksData)
para <- new("metaXpara")
rawPeaks(para) <- peaksData
sampleListFile(para) <- system.file("extdata/faahKO_sampleList.txt",
    package = "metaX")
para <- reSetPeaksData(para)
plotTreeMap(para,valueID="value")
```

---

plsDAPara-class *An S4 class to represent the parameters for PLS-DA analysis*

Description

An S4 class to represent the parameters for PLS-DA analysis

Usage

```
## S4 replacement method for signature 'plsDAPara'
scale(para) <- value

## S4 replacement method for signature 'plsDAPara'
center(para) <- value

## S4 replacement method for signature 'plsDAPara'
t(para) <- value

## S4 replacement method for signature 'plsDAPara'
validation(para) <- value

## S4 replacement method for signature 'plsDAPara'
ncomp(para) <- value

## S4 replacement method for signature 'plsDAPara'
```

```
nperm(para) <- value

## S4 replacement method for signature 'plsDAPara'
kfold(para) <- value

## S4 replacement method for signature 'plsDAPara'
method(para) <- value
```

Arguments

para	A metaXpara object
value	New value

Value

A object of plsDAPara

Methods (by generic)

- scale<-:
- center<-:
- t<-:
- validation<-:
- ncomp<-:
- nperm<-:
- kfold<-:
- method<-:

Slots

scale The method used to scale the data, see preProcess in **metaX**

center A logical which indicates if the matrix should be mean centred or not

t The method used to transform the data, see transformation in **metaX**

validation The method for validation, default is "CV"

ncomp The number of components used for PLS-DA, default is 2

nperm The number of permutations, default is 200

kfold The number of folds for cross-validation, default is 7

do A logical which indicates whether to do the plsDA analysis, default is TRUE

method The method used in PLS-DA. See plsr in **pls**

Author(s)

Bo Wen <wenbo@genomics.cn>

powerAnalyst *Power Analysis*

Description

Power Analysis

Usage

```
powerAnalyst(para, group, valueID = "value", log = TRUE, maxInd = 1000,
  fdr = 0.1)
```

Arguments

para	An metaXpara object
group	A vector of sample names
valueID	The column name used
log	A logical indicating whether transform the data with log2
maxInd	max sample number
fdr	The FDR threshold

Value

An value

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
## Not run:
library(reshape2)
library(dplyr)
a <- read.csv("http://www.metaboanalyst.ca/MetaboAnalyst/resources/data/power_example.csv")
peaksData <- melt(a,id.vars = c("Diet","Sample"),
    value.name = "value",variable.name = "ID")
peaksData <- dplyr::rename(peaksData,class=Diet,sample=Sample)
para <- new("metaXpara")
peaksData(para) <- peaksData
para <- missingValueImpute(para)
para <- metaX::normalize(para)
para <- transformation(para,valueID = "value")
para <- preProcess(para,scale = "pareto",valueID="value")
powerAnalyst(para,group=c("case","control"),log=FALSE,maxInd=200)

## End(Not run)
```

prefix<- *prefix*

Description

prefix

Usage

```
prefix(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
prefix(para) <- "test"
```

preProcess *Pre-Processing*

Description

Pre-Processing

Usage

```
preProcess(para, log = FALSE, scale = c("none", "pareto", "vector", "uv"),
  center = TRUE, valueID = "valueNorm", ...)
```

Arguments

para	An metaX object
log	A logical indicates whether do the log transformation
scale	The method of scaling
center	Centering
valueID	The name of column used for transformation
...	Additional parameter

Value

An new metaX object

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
para <- preProcess(para,valueID = "value",scale="uv")
```

---

qcRlscSpan<- *qcRlscSpan*

Description

qcRlscSpan

Usage

```
qcRlscSpan(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
qcRlscSpan(para) <- 0.4
```

`ratioPairs<-` *ratioPairs*

Description

ratioPairs

Usage

```
ratioPairs(para) <- value
```

Arguments

`para`	An object of metaXpara
`value`	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
ratioPairs(para) <- "1:2"
```

rawPeaks<- *rawPeaks*

Description

rawPeaks

Usage

```
rawPeaks(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
rawPeaks(para) <- data.frame()
```

removeSample *Remove samples from the metaXpara object*

Description

Remove samples from the metaXpara object

Usage

```
removeSample(para, rsamples, ...)

## S4 method for signature 'metaXpara'
removeSample(para, rsamples, ...)
```

Arguments

para	A metaXpara object.
rsamples	The samples needed to be removed
...	Other argument

Value

A metaXpara object.

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
new_para <- removeSample(para,rsamples=c("batch01_QC01"))
```

---

reSetPeaksData *reSetPeaksData*

Description

reSetPeaksData

Usage

```
reSetPeaksData(para)

## S4 method for signature 'metaXpara'
reSetPeaksData(para)
```

Arguments

para	An object of metaXpara

Value

none

Methods (by class)

- metaXpara:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
```

---

retcor.method<- *retcor.method*

Description

retcor.method

Usage

```
retcor.method(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
retcor.method(para) <- "obiwarp"
```

retcor.plottype<- *retcor.plottype*

Description

retcor.plottype

Usage

```
retcor.plottype(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
retcor.plottype(para) <- "deviation"
```

retcor.profStep<- *retcor.profStep*

Description

retcor.profStep

Usage

```
retcor.profStep(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
retcor.profStep(para) <- 0.005
```

---

runPLSDA *runPLSDA*

Description

Validation of the PLS-DA model by using permutation test statistics

Usage

```
runPLSDA(para, plsdaPara, auc = TRUE, sample = NULL,
  valueID = "valueNorm", cpu = 0, label = "order", ...)
```

Arguments

para	An object of metaXpara
plsdaPara	An object of plsDAPara
auc	A logical indicates whether calculate the AUC
sample	Sample class
valueID	The name of column used
cpu	The number of cpu used
label	The label used for plot
...	additional arguments

Value

pvalue

sampleListFile<- *sampleListFile*

Description

sampleListFile

Usage

```
sampleListFile(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
sampleListFile(para) <- "sample.txt"
```

scale<- *scale*

Description

scale

Usage

```
scale(para) <- value
```

Arguments

para	An object of plsDAPara
value	value

Value

An object of plsDAPara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("plsDAPara")
scale(para) <- "uv"
```

`selectBestComponent` *Select the best component for PLS-DA*

Description

Select the best component for PLS-DA

Usage

```
selectBestComponent(para, np = 10, sample = NULL, t = 1,
  method = "oscorespls", scale = NULL, center = TRUE,
  valueID = "valueNorm", validation = "CV", k = 7, ...)
```

Arguments

`para`	A metaXpara object
`np`	The number of max component
`sample`	The sample class used
`t`	Method used to transform the data
`method`	See `plsr`
`scale`	Method used to scale the data
`center`	Centering
`valueID`	The name of column contained the data
`validation`	See `plsr`
`k`	k-fold
`...`	Additional parameter

Value

A list

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
selectBestComponent(para,np=10,sample=c("S","C"),scale="uv",valueID="value")
```

---

t<- *t*

Description

t

Usage

```
t(para) <- value
```

Arguments

para	An object of plsDAPara
value	value

Value

An object of plsDAPara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("plsDAPara")
t(para) <- 1
```

`transformation` *Data transformation*

Description

Data transformation

Usage

```
transformation(para, method = 1, valueID = "valueNorm", ...)
```

Arguments

`para`	An metaX object
`method`	The method for transformation, 1=log, 2=Cube root
`valueID`	The name of column used for transformation
`...`	Additional parameter

Value

An new metaX object

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- missingValueImpute(para)
para <- transformation(para,valueID = "value")
```

validation<- *validation*

Description

validation

Usage

```
validation(para) <- value
```

Arguments

para	An object of plsDAPara
value	value

Value

An object of plsDAPara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("plsDAPara")
validation(para) <- "CV"
```

xcmsSet.fitgauss<- *xcmsSet.fitgauss*

Description

xcmsSet.fitgauss

Usage

```
xcmsSet.fitgauss(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.fitgauss(para) <- FALSE
```

`xcmsSet.fwhm<-` *xcmsSet.fwhm*

Description

xcmsSet.fwhm

Usage

```
xcmsSet.fwhm(para) <- value
```

Arguments

`para`	An object of metaXpara
`value`	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.fwhm(para) <- 30
```

xcmsSet.integrate<- *xcmsSet.integrate*

Description

xcmsSet.integrate

Usage

```
xcmsSet.integrate(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.integrate(para) <- 1
```

xcmsSet.max<- *xcmsSet.max*

Description

xcmsSet.max

Usage

```
xcmsSet.max(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.max(para) <- 5
```

`xcmsSet.method<-` *xcmsSet.method*

Description

xcmsSet.method

Usage

```
xcmsSet.method(para) <- value
```

Arguments

`para`	An object of metaXpara
`value`	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.method(para) <- "centWave"
```

xcmsSet.mzCenterFun<- *xcmsSet.mzCenterFun*

Description

xcmsSet.mzCenterFun

Usage

```
xcmsSet.mzCenterFun(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.mzCenterFun(para) <- "wMean"
```

xcmsSet.mzdiff<- *xcmsSet.mzdiff*

Description

xcmsSet.mzdiff

Usage

```
xcmsSet.mzdiff(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.mzdiff(para) <- -0.001
```

`xcmsSet.noise<-` *xcmsSet.noise*

Description

xcmsSet.noise

Usage

```
xcmsSet.noise(para) <- value
```

Arguments

`para`	An object of metaXpara
`value`	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.noise(para) <- 1000
```

xcmsSet.nSlaves<- *xcmsSet.nSlaves*

Description

xcmsSet.nSlaves

Usage

```
xcmsSet.nSlaves(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.nSlaves(para) <- 8
```

xcmsSet.peakwidth<- *xcmsSet.peakwidth*

Description

xcmsSet.peakwidth

Usage

```
xcmsSet.peakwidth(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.peakwidth(para) <- 12
```

`xcmsSet.polarity<-` *xcmsSet.polarity*

Description

xcmsSet.polarity

Usage

```
xcmsSet.polarity(para) <- value
```

Arguments

`para`	An object of metaXpara
`value`	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.polarity(para) <- "positive"
```

xcmsSet.ppm<- *xcmsSet.ppm*

Description

xcmsSet.ppm

Usage

```
xcmsSet.ppm(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.ppm(para) <- 10
```

xcmsSet.prefilter<- *xcmsSet.prefilter*

Description

xcmsSet.prefilter

Usage

```
xcmsSet.prefilter(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.prefilter(para) <- c(1,5000)
```

---

xcmsSet.profparam<- *xcmsSet.profparam*

Description

xcmsSet.profparam

Usage

```
xcmsSet.profparam(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.profparam(para) <- list(step=0.005)
```

xcmsSet.sleep<- *xcmsSet.sleep*

Description

xcmsSet.sleep

Usage

```
xcmsSet.sleep(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.sleep(para) <- 0
```

xcmsSet.snthresh<- *xcmsSet.snthresh*

Description

xcmsSet.snthresh

Usage

```
xcmsSet.snthresh(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.snthresh(para) <- 5
```

`xcmsSet.step<-` *xcmsSet.step*

Description

xcmsSet.step

Usage

```
xcmsSet.step(para) <- value
```

Arguments

`para`	An object of metaXpara
`value`	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.step(para) <- 0.1
```

xcmsSet.verbose.columns<- *xcmsSet.verbose.columns*

Description

xcmsSet.verbose.columns

Usage

```
xcmsSet.verbose.columns(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSet.verbose.columns(para) <- FALSE
```

xcmsSetObj<- *xcmsSetObj*

Description

xcmsSetObj

Usage

```
xcmsSetObj(para) <- value
```

Arguments

para	An object of metaXpara
value	value

Value

An object of metaXpara

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
library(faahKO)
para <- new("metaXpara")
xcmsSetObj(para) <- faahko
```

`zero2NA` *Convert the value <=0 to NA*

Description

Convert the value <=0 to NA

Usage

```
zero2NA(x, valueID = "value", ...)

## S4 method for signature 'metaXpara'
zero2NA(x, valueID = "value", ...)
```

Arguments

`x`	An object of data
`valueID`	The name of the column which will be used
`...`	Additional parameters

Value

An object of `metaXpara`

Methods (by class)

- `metaXpara`:

Author(s)

Bo Wen <wenbo@genomics.cn>

Examples

```
para <- new("metaXpara")
pfile <- system.file("extdata/MTBLS79.txt",package = "metaX")
sfile <- system.file("extdata/MTBLS79_sampleList.txt",package = "metaX")
rawPeaks(para) <- read.delim(pfile,check.names = FALSE)
sampleListFile(para) <- sfile
para <- reSetPeaksData(para)
para <- zero2NA(para)
```

Index