Package ‘MSbox’

May 25, 2021

Type Package
Title Mass Spectrometry Tools
Version 1.4.0
Author Yonghui Dong
Maintainer Yonghui Dong <yonghui.dong@gmail.com>
Description Common mass spectrometry tools described in John Roboz (2013) <doi:10.1201/b15436>. It allows checking element isotopes, calculating (isotope labelled) exact monoisotopic mass, m/z values and mass accuracy, and inspecting possible contaminant mass peaks, examining possible adducts in electrospray ionization (ESI) and matrix-assisted laser desorption ionization (MALDI) ion sources.
Depends R (>= 3.5.0)
Imports stringr, crayon, xml2, stats, reshape2, ggplot2, ggfortify, plotly
License GPL-2
URL https://github.com/YonghuiDong/MSbox
BugReports https://github.com/YonghuiDong/MSbox/issues/new
Encoding UTF-8
RoxygenNote 7.1.1
NeedsCompilation no
Repository CRAN
Date/Publication 2021-05-25 16:30:02 UTC

R topics documented:

adduct ................................................................. 2
contam ................................................................... 3
describe ................................................................. 3
doNormalization .................................................... 4
adduct

```r
adduct(F, mode = c("+", "-"))
```

**Description**

calculate common adduct ions in positive or negative ion mode

**Usage**

```r
adduct('C1H4', mode = 'W')
adduct('C1h4', mode = '+')
```

**Arguments**

- `F` chemical formula, case insensitive
- `mode` ionization mode, either positive `'+` or negative `'-`.

**Author(s)**

Yonghui Dong

**Examples**

```r
adduct('C1H4', mode = 'W')
adduct('C1h4', mode = '+')
```
**contam**

*Contaminants in MS*

---

**Description**

check the possible contaminants

**Usage**

```r
contam(mz, mode = NULL, ppm = 10)
```

**Arguments**

- `mz` suspected m/z value
- `mode` ionization mode, either positive '+' or negative '-'
- `ppm` mass tolerance, default value = 10

**Author(s)**

Yonghui Dong

**Examples**

```r
contam(33.0335, ppm = 10, mode = '+')
contam(44.998, ppm = 10, mode = '-')
```

---

**describe**

*Get the compound information*

---

**Description**

get compound formula and structure from https://cactus.nci.nih.gov/chemical/structure

**Usage**

```r
describe(chem, representation = "formula", info = FALSE)
```

**Arguments**

- `chem`, chemical name of the compound
- `representation`, representation methods, formula is default
- `info`, extra molecular information that users can query
doNormalization

Author(s)

Yonghui Dong

Examples

```r
## Not run:
describe('malic acid', "formula")
describe(c('malic acid', 'citric acid', 'tartaric acid'), "smiles")
## End(Not run)
```

doNormalization  

perform normalization

Description

perform normalization

Usage

doNormalization(x, method = NULL)

Arguments

- `x`  
  sample ion intensity matrix

- `method`  
  normalization method: (1) LBME: linear baseline normalization based on mean values; (2) LBMD: linear baseline normalization based on median values; (3) PQN: probabilistic quotient normalization; (4) QT: quantile normalization; (5) TIC: total ion current normalization.

Value

normalized data matrix

Examples

```r
dat <- matrix(runif(100*10), ncol = 100, nrow = 10)
out <- doNormalization(dat, method = "PQN")
```
doStat  

**Performing statistics**

**Description**

performing statistics, including calculating fold change, p-values and VIP values

**Usage**

```r
doStat(x, Group = NULL)
```

**Arguments**

- `x`: sample ion intensity matrix, row sample, column feature.
- `Group`: sample group information

**Value**

a dataframe with statistical information

**Examples**

```r
dat <- matrix(runif(2*300), ncol = 2, nrow = 300)
rownames(dat) <- 1:dim(dat)[1]
myGroup <- rep_len(LETTERS[1:3], 300)
ret <- doStat(dat, Group = myGroup)
```

---

**E_iso**  

**Element isotopes**

**Description**

check element isotope information

**Usage**

```r
E_iso(S)
```

**Arguments**

- `S`: element, can be element symbol (i.e. C) or full name (i.e. Carbon). Both Element symbol and full name are case insensitive.

**Author(s)**

Yonghui Dong
Examples

- `E_iso('Na')` # element symbol
- `E_iso('nA')` # element symbol, case insensitive
- `E_iso('Carbon')` # element full name
- `E_iso('carBon')` # element full name, case insensitive

getCV  \hspace{1cm} \textit{Calculate coefficient of variation (CV)}

Description

Calculate coefficient of variation (CV), also known as relative standard deviation (RSD) among different sample groups

Usage

getCV(x, Group = NULL)

Arguments

- \texttt{x} \hspace{1cm} sample ion intensity matrix, row sample, column feature.
- \texttt{Group} \hspace{1cm} sample group information

Value

a dataframe with mean values and cv

Examples

```
dat <- matrix(runif(2*300), ncol = 2, nrow = 300)
myGroup <- rep_len(LETTERS[1:2], 300)
ret <- getCV(dat, Group = myGroup)
```

getFC  \hspace{1cm} \textit{calculate fold change}

Description

calculate fold change among different samples.

Usage

getFC(x, Group = NULL)
getMax

Arguments

x sample ion intensity matrix, row sample, column feature.

Group sample group information

Value

a dataframe with mean values and fold changes

Examples

dat <- matrix(runif(2*300), ncol = 2, nrow = 300)
myGroup <- rep_len(LETTERS[1:2], 300)
ret <- getFC(dat, Group = myGroup)

dat <- cbind.data.frame(mz = c(100, 101, 300), mz2 = c(0, 0 , 1), mz3 = c(1, 9, 1))
rownames(dat) <- c("A", "B", "C")
out <- getMax(dat)
**getP**

get *p*-values

**Description**

get *p*-values from Post Hoc analysis

**Usage**

```r
getP(x, Group = NULL)
```

**Arguments**

- **x**: sample ion intensity matrix, row sample, column feature.
- **Group**: sample group information

**Value**

a data frame

**Examples**

```r
dat <- matrix(runif(2*300), ncol = 2, nrow = 300)
myGroup <- rep_len(LETTERS[1:3], 300)
out <- getP(dat, Group = myGroup)
```

---

**Iso_mass**

*Isotope labelled molecular mass*

**Description**

Calculate isotope labelled molecular mass

**Usage**

```r
Iso_mass(F, iso)
```

**Arguments**

- **F**: chemical formula, case insensitive
- **iso**: labelled elements, case insensitive

**Author(s)**

Yonghui Dong
Iso_mz

Examples
Iso_mz(F = 'C7H6O4', iso = '[13]C2[2]H3') # Two 13C and three 2H are labeled

---

Iso_mz Isotope labelled molecular mass

Description
Calculate isotope labelled m/z

Usage
Iso_mz(F, iso, z)

Arguments
F, chemical formula, case insensitive
iso, labelled elements, case insensitive
z charge

Author(s)
Yonghui Dong

Examples
Iso_mz(F = 'C7H6O4', iso = '[13]C2[2]H3', z = -1) # Two 13C and three 2H are labeled

---

mass molecular mass

Description
calculate accurate molecular mass

Usage
mass(F, caseSensitive = FALSE)
Arguments

- `m` chemical formula of an ion, case insensitive
- `z` charge
- `caseSensitive` if case sensitive is ‘FALSE’ (default), the elements are separated by numbers. For instance, Carbon dioxide can be written as ‘c1o2’ or any combination of the two elements in lower or upper cases. However, the number of elements should be clearly stated in the chemical formula. If case sensitive is ‘TRUE’, the elements are separated by upper case letters. For instance, Carbon dioxide must be written as ‘C1O2’ or ‘CO2’. You don’t need to write the number of the element if it is 1.

Author(s)

- Yonghui Dong

Examples

```r
mass('C7h7o1')
mass('C7H7O', caseSensitive = TRUE)
mass(c('C7H7O4', 'C'), caseSensitive = TRUE) # vector input
mass(c('c7h7o4', 'c1'))
```

**Description**

Calculate accurate mass-to-charge ratio (m/z)

**Usage**

```r
mz(m, z, caseSensitive = FALSE)
```

**Arguments**

- `m` chemical formula of an ion, case insensitive
- `z` charge
- `caseSensitive` if case sensitive is ‘FALSE’ (default), the elements are separated by numbers. For instance, Carbon dioxide can be written as ‘c1o2’ or any combination of the two elements in lower or upper cases. However, the number of elements should be clearly stated in the chemical formula. If case sensitive is ‘TRUE’, the elements are separated by upper case letters. For instance, Carbon dioxide must be written as ‘C1O2’ or ‘CO2’. You don’t need to write the number of the element if it is 1.

Author(s)

- Yonghui Dong
**Examples**

```r
mz('C7H7O1', z = 1)
mz('C7H7O', z = 1, caseSensitive = TRUE)
mz(c('C7H7O4', 'C'), z = -1, caseSensitive = TRUE) # vector input
mz(c('c7h7o4', 'c1'), z = -1)
```

**Description**

Calculate the mass accuracy of measured m/z. Lazy input allowed.

**Usage**

```r
ppm(m, t, lazy = TRUE)
```

**Arguments**

- `m`: Measured m/z
- `t`: Theoretical m/z
- `lazy`: If lazy input is allowed

**Author(s)**

Yonghui Dong

**Examples**

```r
ppm(155.03383, 155.03388) # with m/z value
ppm(155.03383, .03388) # lazy input when the integer parts of m and t are the same
ppm(155.03384, mz('C7H7O4', z = 1)) # with ion formula
```

---

**searchDB**

*Search in customized database*

**Description**

Search in customized database based on accurate m/z and RT.

**Usage**

```r
searchDB(DF, DB, ppm = 5, RT = 0.2, useRT = FALSE)
```
viewPCA

Arguments

- **DF**
  - input file, should contain at least a column named mz
- **DB**
  - database, should contain at least a column named mz
- **ppm**
  - mass tolerance, default 5ppm
- **RT**
  - retention time tolerance, default 0.2min
- **useRT**
  - should RT be considered during database search?

Author(s)

Yonghui Dong

Examples

```r
## Not run:
searchDB(DF, DB)
## End(Not run)
```

## Description

perform PCA from xcms object

Usage

```r
viewPCA(
  dat,
  Group = NULL,
  centering = T,
  scaling = "none",
  x = 1,
  y = 2,
  size = 1.5,
  exclude = NULL,
  scale_group = NULL,
  scale_factor = 1,
  interactive = T,
  ...
)
```
Arguments

- `dat`: sample ion intensity matrix, row sample, column feature.
- `Group`: sample group information
- `centering`: centering, default = TRUE
- `scaling`: scaling method, default is scaling = "none". You can choose "auto" or "pareto"
- `x`: PCA X axis, default is PC1
- `y`: PCA Y axis, default is PC2
- `size`: dot size
- `exclude`: exclude some classes of samples
- `scale_group`: select groups needs to be scaled.
- `scale_factor`: the scale factor, default = 1.
- `interactive`: should interactive figure be plotted? default = TRUE. If you want to save the result in high resolution, use non interactive plot.
- `...`: other parameters

Author(s)

Yonghui Dong

Examples

```r
dat <- matrix(runif(2*300), ncol = 2, nrow = 300)
Group <- rep_len(LETTERS[1:3], 300)
out <- viewPCA(dat, Group = Group)
```

Description

View variations of TIC among samples

Usage

```r
viewTIC(
  x,
  Seq = NULL,
  Batch = NULL,
  Group = NULL,
  Trans = "none",
  resultBy = "Group"
)
```
viewVolcano

View volcano plot

Description

View volcano plot.

Usage

viewVolcano(result, compare_group, FC = 2, p = 0.05)

Arguments

result result from doStat() function
compare_group which groups you want to compare, i.e. c("WT", "Treat1"), only two groups are allowed
FC select fold change values, default = 2
p select p value, default = 0.05

Author(s)

Yonghui Dong
what search for m/z in from the idiom metabolomics database

Description

tentative metabolite identification based on m/z value search

Usage

what(myMZ, mode = NULL, ppm = 5, useDB = "HMDB")

Arguments

myMZ m/z values
mode ionization mode, either positive ‘+’ or negative ‘-’
ppm mass tolerance, default value = 10
useDB which database to use, HMDB or KEGG? default is HMDB

Author(s)

Yonghui Dong

Examples

a = what(133.014, mode = '-', ppm = 10)
Index

adduct, 2
contam, 3
describe, 3
doNormalization, 4
doStat, 5
E_iso, 5
getCV, 6
getFC, 6
getMax, 7
getP, 8
Iso_mass, 8
Iso_mz, 9
mass, 9
mz, 10
ppm, 11
searchDB, 11
viewPCA, 12
viewTIC, 13
viewVolcano, 14
what, 15