

Package ‘mtExtra’

April 2, 2025

Date 2023-08-17
Type Package
Title More functions for metabolomics data analysis
Version 0.1.3
Author Wanchang Lin
Maintainer Wanchang Lin <wanchanglin@hotmail.com>
Description Further functions for metabolomics data analysis.
Depends R (>= 3.0.0)
License GPL (>= 2)
URL <https://github.com/wanchanglin/mtExtra>
BugReports <https://github.com/wanchanglin/mtExtra/issues>
Encoding UTF-8
Roxygen list(markdown = TRUE)
RoxygenNote 7.3.2
LazyLoad yes
LazyData yes
ZipData No
Imports tidy, dplyr, purrr, tibble, ggplot2, plyr, reshape2, magrittr,
cowplot, ggdendro, ggraph, igraph, MASS, lattice, latticeExtra, ellipse,
ppcor, locfdr, rlang

Contents

arrange_row	2
batch_shift	3
bi_cor_net	4
blank_filter	5
chunk	6
cor_hcl	6
cor_net	7
cor_tab	8
dat2long	9
dat_summ	10

dat_symb	10
dat_trans	11
df_na_idx	12
df_t	12
dim_mat	13
feat_count	13
gg_heat_dend	14
graph_stats	15
heat_dend	15
is_zero	16
locfdr_filter	16
mat2df	18
mv_filter	18
mv_perc	19
net_graph	20
non_digit	20
non_zero	21
outl_det_m	21
outl_det_u	22
pca_plot	23
pcor_dat	23
plot_aam	24
plot_pval	25
qc_filter	26
range_scale	27
rbind_df	27
rsd	28
rsd_filter	29
samp_strat	30
samp_sub	30
sym2long	31
trim_str	32
upd_data	32
var_filter	33
vec2dat	34
vec_segment	34
vec_stats	35
vertex_stats	36

Index	37
--------------	-----------

arrange_row	<i>Sort a data frame</i>
-------------	--------------------------

Description

Sort a data frame and keep the row names.

Usage

```
arrange_row(df, ...)
```

Arguments

df a data frame.
... further parameters to order.

Details

This function is modified from arrange of R package `plyr`. `arrange` deliberately remove row-names, but sometimes it is worth to keep them.

Value

a sorted data frame.

batch_shift	<i>Batch shifting</i>
-------------	-----------------------

Description

Remove batch effect withing each block.

Usage

```
batch_shift(x, y, method = "mean", overall_average = TRUE)
```

Arguments

x a data matrix.
y a categorical data for batch/block information.
method method for shifting.
overall_average a logical value to indicate whether or not an overall average will be added after shifting.

Value

a shifted data matrix.

References

Silvia Wagner, et.al, Tools in Metabonomics: An Integrated Validation Approach for LC-MS Metabolic Profiling of Mercapturic Acids in Human Urine *Anal. Chem.*, 2007, 79 (7), pp 2918-2926, DOI: 10.1021/ac062153w

bi_cor_net	<i>Bipartite/two-mode correlation network</i>
------------	---

Description

Perform bipartite/two-mode correlation network analysis.

Usage

```
bi_cor_net(co_mat, thres = 0.6, dn = NULL)
```

Arguments

co_mat	a correlation coefficient matrix.
thres	correlation coefficient threshold for network analysis.
dn	a character string for correlation name.

Value

a list of ggplot2 plots.

Examples

```
data <- mtExtra:::data
meta <- mtExtra:::meta

## Not run:
## process meta
library(mt)
meta <- mv_filter(meta, thres = 0.3)$dat
meta <- mv.fill(meta, method = "mean")
meta <- preproc(meta, method = "auto")

## End(Not run)

co <- cor(meta, data, use = "pairwise.complete.obs")
heat_dend(co)

res <- bi_cor_net(co, thres = 0.2)
names(res)
res$net1
res$dend2
```

blank_filter*MS/NMR data filtering based on "blank" data*

Description

MS/NMR data filtering based on "blank" data

Usage

```
blank_filter(  
  x,  
  y,  
  method = c("mean", "median", "max"),  
  factor = 1,  
  f_mv = TRUE,  
  thres_mv = 0.3  
)
```

Arguments

x	a data matrix.
y	a character string with contents of "sample", "qc" and "blank".
method	method for stats. Support "mean", "median" and "max".
factor	multiplier for blank stats
f_mv	a flag indicating whether perform missing value filtering on "sample" data.
thres_mv	threshold of missing values on QC. Features less than this threshold will be kept.

Details

This function provides an option to perform missing value filtering on "sample" data.

Value

a list of with contents:

- dat the filtered data matrix
- idx a logical vector of index for keeping features.

See Also

Other variable filters: [locfdr_filter\(\)](#), [mv_filter\(\)](#), [qc_filter\(\)](#), [rsd_filter\(\)](#), [var_filter\(\)](#)

chunk	<i>Split a vector into chunks</i>
-------	-----------------------------------

Description

Split a vector into chunks.

Usage

```
chunk(x, n)
```

Arguments

x	an vector.
n	length of chunk.

Details

For details, see <https://bit.ly/2SM4m2G>

Value

return the split chunk.

Examples

```
x <- 1:10  
n <- 3  
chunk(x, n)
```

cor_hcl	<i>Correlation hierarchical cluster analysis</i>
---------	--

Description

Perform hierarchical cluster analysis based on correlation.

Usage

```
cor_hcl(  
  mat,  
  cutoff = 0.75,  
  use = "pairwise.complete.obs",  
  method = "pearson",  
  fig_title = "Cluster Dendrogram",  
  size = 3,  
  rotate = FALSE  
)
```

Arguments

mat	a data matrix
cutoff	a threshold for correlation analysis.
use	a string giving a method for computing covariances. For details, see <code>stats::cor()</code> .
method	method for correlation <code>stats::cor()</code> .
fig_title	title for plotting
size	text font size of ggplot2.
rotate	a logical indicating whether to rotate plot.

Value

a list of a ggplot2 object for clustering and a list of cluster centres.

Examples

```
cor_hcl(mtcars)
```

cor_net	<i>Correlation network analysis</i>
---------	-------------------------------------

Description

Perform network analysis for correlation coefficient.

Usage

```
cor_net(
  mat,
  use = "pairwise.complete.obs",
  method = "pearson",
  thres = 0.6,
  fig_title = "Correlation network"
)
```

Arguments

mat	a data matrix for correlation analysis
use, method	parameters of <code>stats::cor()</code> .
thres	correlation coefficient threshold for network analysis. Only keep those with coefficient larger than thres.
fig_title	a character string for figure title.

Value

a list of ggplot2 plots.

Examples

```
res <- cor_net(mtcars)
names(res)
res$p1      # or res$p2, res$p3, res$p4
## Not run:
## see https://ggplot2-book.org/arranging-plots.html
library(patchwork)
res$p1 + res$p2 + res$p3 + res$p4

## End(Not run)
```

cor_tab

Calculate correlation coefficient and p-values

Description

Calculate correlation coefficient and p-values.

Usage

```
cor_tab(
  x,
  cor.method = c("pearson", "kendall", "spearman"),
  adj.method = c("holm", "hochberg", "hommel", "bonferroni", "BH", "BY", "fdr", "none"),
  ...
)
```

Arguments

x	a data frame or matrix for correlation analysis column-wise.
cor.method	method for correlation
adj.method	p-value correction method
...	other parameter for correlation.

Details

This file is modified from `cor.table` of package `picante` and `corr.test` of package `psych`. The original implementation is from Bill Venables, the author of R great book MASS. For details, see <https://stat.ethz.ch/pipermail/r-help/2001-November/016201.html>

Value

a list with contents:

- r correlation coefficient
- p statistics matrix, in which the lower triangular is p-values and the upper triangular is adjusted p-values

Examples

```
co <- cor_tab(mtcars, cor.method = "spearman", adj.method = "BH")
names(co)

## Not run:
library(psy)
co <- corr.test(mtcars, method="spearman",adjust="BH")
# From psy: For symmetric matrices, p values adjusted for multiple tests
# are reported above the diagonal.

## End(Not run)
```

dat2long

Melt a numeric data matrix to long format

Description

Reshape a matrix or data frame to long format with row names and column names in two columns.

Usage

```
dat2long(x)
```

Arguments

x a matrix or data frame

Details

`reshape2::melt` keeps the rownames when melting a matrix, but not when melting a data frame. This function keeps rownames for both matrix and data frame.

Value

a tibble object

Examples

```
class(mtcars)        # data.frame
reshape2::melt(mtcars)
reshape2::melt(as.matrix(mtcars))
dat2long(mtcars)
```

dat_summ	<i>Data matrix summary</i>
----------	----------------------------

Description

Summarise a data matrix. This is a wrapper function for any summary function on an vector.

Usage

```
dat_summ(x, method = mean, ...)
```

Arguments

x	a matrix-like object.
method	summary method for an vector.
...	further parameters for method.

Value

a summarised table.

Examples

```
library(dplyr)
library(tidyr)
library(purrr)
iris %>% group_by(Species) %>% group_modify(~ dat_summ(., method = mean))
iris %>% group_by(Species) %>% group_modify(~ dat_summ(., method = vec_stats))
iris %>% group_by(Species) %>% do(dat_summ(., method = vec_segment))
iris %>% dat_summ(method = sd)
```

dat_symb	<i>Symbolise data set</i>
----------	---------------------------

Description

Symbolise data values as (-1, 0, 1) based on a range

Usage

```
dat_symb(x, thres)
```

Arguments

x	an vector or data matrix.
thres	an vector with lower and higher values.

Details

thres can be estimated by [locfdr_filter\(\)](#).

Value

a symbolic data set.

See Also

[locfdr_filter\(\)](#) for estimating threshold and [locfdr\(\)](#) for $z.2$.

dat_trans	<i>Transform data</i>
-----------	-----------------------

Description

Perform data set transformation

Usage

```
dat_trans(x, method = "auto", na.rm = TRUE, add = 1)
```

```
vec_trans(x, method = "auto", na.rm = TRUE, add = 1)
```

Arguments

x	a matrix, data frame or vector.
method	transformation method, including: "center", "auto", "range", "pareto", "vast", "level", "log", "log10", "sqrt" and "asinh".
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
add	a shift value for log transformation.

Value

transformed data

References

Berg, R., Hoefsloot, H., Westerhuis, J., Smilde, A. and Werf, M. (2006), Centering, scaling, and transformations: improving the biological information content of metabolomics data, *BMC Genomics*, 7:142

Examples

```
data(iris)

## transform an vector
vec <- iris[, 1]
dat_trans(vec, method = "auto")
vec_trans(vec, method = "auto")

## transform a data frame
mat <- iris[, 1:4]
dat_trans(mat, method = "log")
```

```
## transform data frame under different conditions
plyr::ddply(iris, ("Species"), function(x, method) {
  dat_trans(x[, 1:4], method = method)
}, method = "range")

## use 'tidyverse'
library(dplyr)
library(tidyr)
library(purrr)

## transform whole data set
iris %>% mutate(across(where(is.numeric), ~ vec_trans(., method = "range")))

## transform data set within groups
iris %>%
  group_by(Species) %>%
  mutate(across(where(is.numeric), ~ vec_trans(., method = "range")))
```

df_na_idx	<i>Get row index of missing values in data frame</i>
-----------	--

Description

Get row index of missing values in data frame.

Usage

```
df_na_idx(df, vars)
```

Arguments

df	data frame being used to check index of missing values.
vars	vector of variable names in df for missing values checking.

Value

an vector of missing value index.

df_t	<i>Transpose a numeric data frame (with/without NAs)</i>
------	--

Description

Transpose a numeric data frame (with/without NAs)

Usage

```
df_t(x)
```

Arguments

x a data frame

Details

It is used mostly for numeric data frame with NAs and return a

Value

a transposed data frame.

Examples

```
x <- data.frame(group = c(1,1,2,NA,2), val = c(6,4,6,3.1,NA))
x <- df_t(x)
```

dim_mat

Get dimension matrix of a list of data frame

Description

Get a matrix for dimension information of a list of data frame.

Usage

```
dim_mat(mat_list)
```

Arguments

mat_list a list of data frame.

Value

a data frame.

feat_count

Calculate the count number of features

Description

Calculate the features counts based on the frequency of multiple selectors.

Usage

```
feat_count(fs.ord, top.k = 30)
```

Arguments

fs.ord data matrix for multiple feature order lists.
top.k top feature number to be processed.

Value

a data matrix of feature counts.

Examples

```
fs.ord <- mtExtra::fs.ord
feat_count(fs.ord, top.k = 20)
```

gg_heat_dend

Heatmap with dendrograms with ggplot2

Description

Plot heatmap of a data matrix using ggplot2. This function is modified from <https://bit.ly/2UUnY2L>.

Usage

```
gg_heat_dend(
  mat,
  row.dend = T,
  col.dend = T,
  row.dend.right = TRUE,
  colors = c("red", "white", "blue"),
  font.size = 10,
  x.rot = 60,
  legend.title = "value",
  dist.method = "euclidean",
  clust.method = "complete",
  dend.line.size = 0.5
)
```

Arguments

mat	a data matrix to be plotted.
row.dend	plot row dendrogram or not.
col.dend	plot column dendrogram or not.
row.dend.right	a logical value to indicate the position of row dendrogram.
colors	a vector of colours for heatmap.
font.size	label font size.
x.rot	plot rotate degree.
legend.title	legend title.
dist.method	distance method.
clust.method	cluster method.
dend.line.size	dendrogram line size.

Value

an object of class ggplot2.

Examples

```
gg_heat_dend(mtcars)
```

graph_stats	<i>Graph level metrics</i>
-------------	----------------------------

Description

Return graph level metrics.

Usage

```
graph_stats(g)
```

Arguments

`g` an igraph object.

Value

a vector of graph level metrics.

heat_dend	<i>Heatmap with dendrogram on both sides</i>
-----------	--

Description

Plot heatmap with dendrogram on both sides using lattice

Usage

```
heat_dend(
  mat,
  x.rot = 60,
  col = c("red", "white", "blue"),
  cex = 0.5,
  xlab = "",
  ylab = "",
  main = "",
  ...
)
```

Arguments

`mat` a data matrix for plotting.
`x.rot` the rotate degree.
`col` colours for heatmap.
`cex` font size.
`xlab, ylab, main` character strings for xlab, ylab and figure title.
`...` further parameters for lattice.

Value

an object of class `lattice`.

Examples

```
heat_dend(mtcars)
```

`is_zero`

Calculate the zero percentage

Description

Calculate the zero percentage

Usage

```
is_zero(mat)
```

Arguments

`mat` a data matrix.

Value

an vector.

`locfdr_filter`

Filtering variable based on local false discovery rate

Description

Filter data based on local false discovery rate. This function uses `z.2` of `locfdr`.

Usage

```
locfdr_filter(x, plot = 1, thres = NULL, ...)
```

Arguments

`x` a data matrix

`plot` an integer for plotting. 0 gives no plots. 1 gives single plot showing the histogram of `zz` and fitted densities `f` and `p0*f0`.

`thres` a user defined threshold for filtering. The default is `NULL`, which use local FDR as threshold for filtering.

`...` other parameters to be passed to `locfdr`.

Details

- Keep the variables which have at least one significant element. The significant element is defined as larger than the lower of threshold or less than the upper of threshold.
- Threshold can be given by user or be estimated by locfdr, i.e. the returned `z.2` as a threshold. It is not guaranteed that locfdr `z.2`. If not, user must provide this value.
- From R package locfdr vignette: `z.2` is the interval along the `zz`-axis outside of which $\text{fdr}(z) < 0.2$, the locations of the yellow triangles in the histogram plot. If no elements of `zz` on the left or right satisfy the criterion, the corresponding element of `z.2` is NA, and the corresponding triangle does not appear.

Value

a list of with contents:

- `dat` the filtered data matrix.
- `idx` a vector of filtering index.
- `thres` threshold used for filtering.

See Also

[locfdr\(\)](#)

Other variable filters: [blank_filter\(\)](#), [mv_filter\(\)](#), [qc_filter\(\)](#), [rsd_filter\(\)](#), [var_filter\(\)](#)

Examples

```
## Not run:
library(dplyr)
library(tidyr)
library(purrr)
library(readr)

## get ionomics data
dat <- read_csv("https://github.com/wanchanglin/ionflow/raw/master/extra/paper_ko.csv")
dim(dat)

## missing values filling with mean
dat <- dat %>%
  mutate(across(where(is.numeric), function(x) {
    m <- mean(x, na.rm = TRUE)
    x[is.na(x)] <- m
  }
  ))
dat

res <- locfdr_filter(t(dat[, -1]), plot = 1)
res$thres

## filter data
dat <- dat[res$idx, , drop = FALSE]

## symbolise data
dat_sym <- dat %>%
  mutate(across(where(is.numeric), ~ dat_symb(., thres = res$thres)))

## End(Not run)
```

mat2df	<i>Convert matrix to df</i>
--------	-----------------------------

Description

Convert matrix to df

Usage

```
mat2df(x)
```

Arguments

x a matrix

Details

from .matrix_to_df of package plyr.

Value

a data frame.

mv_filter	<i>Filtering variable based on the percentage of missing values</i>
-----------	---

Description

This function calculates the percentage of missing values and keeps those features with missing values percentage less than the designed threshold.

Usage

```
mv_filter(x, thres = 0.3)
```

Arguments

x a data matrix. The columns are features.
thres threshold of missing values. Features less than this threshold will be kept. Value has to be between 0 and 1.

Value

a list of with contents:

- dat the filtered data matrix
- idx a logical vector of index for keeping features.

See Also[mv_perc\(\)](#)Other variable filters: [blank_filter\(\)](#), [locfdr_filter\(\)](#), [qc_filter\(\)](#), [rsd_filter\(\)](#), [var_filter\(\)](#)**Examples**

```
meta <- mtExtra:::meta
mv_perc(meta)
mv_filter(meta, thres = 0.3)
```

mv_perc	<i>Missing value percentage</i>
---------	---------------------------------

Description

Calculate missing value percentage.

Usage

```
mv_perc(x)
```

Arguments

x an vector, matrix or data frame.

Value

missing value percentage.

See Also[mv_filter\(\)](#)**Examples**

```
meta <- mtExtra:::meta
mv_perc(meta)
mv_filter(meta, thres = 0.3)
```

net_graph	Create an igraph object
-----------	-------------------------

Description

Create an igraph object

Usage

```
net_graph(edge_df, node_df = NULL)
```

Arguments

edge_df	graph edge matrix.
node_df	graph vertex matrix.

Value

an igraph object with some vertex statistics and community detection center.

non_digit	Calculate the percentage of non digits
-----------	--

Description

Calculate the percentage of non digits.

Usage

```
non_digit(mat)
```

Arguments

mat	a data matrix.
-----	----------------

Value

an vector.

non_zero	<i>Calculate the non-zero percentage</i>
----------	--

Description

Calculate the non-zero percentage.

Usage

```
non_zero(mat)
```

Arguments

mat	a data matrix.
-----	----------------

Value

an vector.

outl_det_m	<i>Multivariate outlier detection</i>
------------	---------------------------------------

Description

Perform multivariate outlier detection.

Usage

```
outl_det_m(x, method = "mcd", conf.level = 0.95)
```

Arguments

x	a data matrix.
method	methods for resistant estimation of multivariate location and scatter. Only mve, mcd and classical are supported.
conf.level	a confidential level.

Value

a logical vector.

See Also

[cov.rob\(\)](#) for "Resistant Estimation of Multivariate Location and Scatter"

Other outlier detectors: [outl_det_u\(\)](#)

Examples

```
set.seed(134)
x <- cbind(rnorm(80), rnorm(80), rnorm(80))
y <- cbind(rnorm(10, 5, 1), rnorm(10, 5, 1), rnorm(10, 5, 1))
x <- rbind(x, y)
outl <- outl_det_m(x, method = "mcd", conf.level = 0.95)
```

outl_det_u

*Univariate outlier detection***Description**

Perform outlier detection using univariate method.

Usage

```
outl_det_u(x, method = c("percentile", "median"))
```

Arguments

x	a numeric vector.
method	method for univariate outlier detection. Only percentile and median are supported.

Details

- median: the absolute difference between the observation and the sample median is larger than 2 times of the Median Absolute Deviation divided by 0.6745.
- percentile: either smaller than the 1st quartile minus 1.5 times of IQR, or larger than the 3rd quartile plus 1.5 times of IQR.

Value

a logical vector.

References

Wilcox R R, Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy, Springer 2010 (2nd edition), pages 31-35.

See Also

Other outlier detectors: [outl_det_m\(\)](#)

Examples

```
x <- c(2, 3, 4, 5, 6, 7, NA, 9, 50, 50)
outl_det_u(x, "percentile")
```

pca_plot

PCA plot

Description

Plot PCA of a matrix or data frame with base R function.

Usage

```
pca_plot(x, y = NULL, scale = TRUE, ep.plot = FALSE, ...)
```

Arguments

x	a matrix for plotting. Should have row names
y	a factor or character vector specifying the group information of the row.
scale	a logical flag to indicate whether or not scale x.
ep.plot	plot ellipse or not.
...	further parameters passed to plot

Details

The rownames of x will be shown in the plot. y is used to indicate the group of row of x.

Value

a base R graphics object.

Author(s)

Wanchang Lin

Examples

```
data(iris)
pca_plot(iris[, 1:4], iris[, 5], ep.plot = TRUE)
```

pcor_dat

Partial correlation analysis

Description

Perform partial correlation analysis of two data matrix

Usage

```
pcor_dat(x, y, method = c("pearson", "kendall", "spearman"))
```

Arguments

x, y	two data matrix for correlation analyses.
method	correlation method.

Value

a correlation matrix.

See Also

[ppcor::pcor\(\)](#)

Examples

```
library(dplyr)
library(tidyr)
library(purrr)

## 'data' has been processed.
data <- mtExtra::data
meta <- mtExtra::meta

## filtering
meta <- mv_filter(meta, thres = 0.3)$dat

## missing value filling
meta <- meta %>%
  mutate(across(where(is.numeric), function(x) {
    m <- mean(x, na.rm = TRUE)
    x[is.na(x)] <- m
    x
  })))

## normalisation
meta <- meta %>%
  mutate(across(where(is.numeric), ~ {
    .x <- (.x - mean(.x, na.rm = TRUE)) / sd(.x, na.rm = TRUE)
  })))

co <- pcor_dat(meta, data)
heat_dend(co)
```

plot_aam

Wrapper function for plotting classification results

Description

This function plots accuracy, AUC and margin (aam) of classification results from package mt.

Usage

```
plot_aam(aam_list, fig_title = "Accuracy, AUC and Margin")
```


Arguments

aam_list a data matrix of classification results.
fig_title a string of figure title

Value

an object of class ggplot2

See Also

aam.mcl in R package mt for how to get accuracy, AUC and margin.

Examples

```
aam <- mtExtra:::aam  
plot_aam(aam)
```

plot_pval

Plot adjusted p-values

Description

Plot the adjusted p-values using ggplot2

Usage

```
plot_pval(pval_list)
```

Arguments

pval_list a data matrix or a list of data matrix of p-value correction

Value

an object of class ggplot2

Examples

```
pval <- mtExtra:::pval  
plot_pval(pval)
```

 qc_filter

MS/NMR data filtering based on RSD of "qc" data

Description

MS/NMR data filtering based on RSD of "qc" data

Usage

```
qc_filter(
  x,
  y,
  thres_rsd = 20,
  f_mv = TRUE,
  f_mv_qc_sam = FALSE,
  thres_mv = 0.3
)
```

Arguments

x	a data matrix.
y	a character string with contents of "sample", "qc" and "blank".
thres_rsd	threshold of RSD on QC. Features less than this threshold will be kept.
f_mv	a flag indicating whether or not to performance missing value filtering on either "sample" or "qc" data.
f_mv_qc_sam	a flag for filtering using percentage of missing values on "qc" or "sample". TRUE is for "qc".
thres_mv	threshold of missing values. Features less than this threshold will be kept.

Details

This filter process takes two steps. First, the missing values filtering is performed on either "qc" or "sample". Then RSD-based filtering is applied to "qc" data.

Value

a list of with contents:

- dat the filtered data matrix
- idx a logical vector of index for keeping features.

See Also

Other variable filters: [blank_filter\(\)](#), [locfdr_filter\(\)](#), [mv_filter\(\)](#), [rsd_filter\(\)](#), [var_filter\(\)](#)

range_scale	<i>Scale vector to a defined range</i>
-------------	--

Description

Scale vector to a defined range.

Usage

```
range_scale(x, range = c(0, 1))
```

Arguments

x	a numeric vector
range	a vector with two values: lower and higher.

Value

a scaled vector.

Examples

```
set.seed(100)
x <- rnorm(10)
range_scale(x, range = c(10, 20))
```

rbind_df	<i>Row binding of a list of data matrix</i>
----------	---

Description

Row bind of a list of matrix or data frame with the same dimension.

Usage

```
rbind_df(list)
```

Arguments

list	a list of data matrix.
------	------------------------

Value

a data matrix.

Examples

```
df1 <- data.frame(x = c(1, 3), y = c(2, 4))
df2 <- data.frame(x = c(5, 7), y = c(6, 8))
df <- list(df1 = df1, df2 = df2)
rbind_df(df)

tmp <- vector(mode = "list", length = 3)
for (i in 1:3) {
  tmp[[i]] <- data.frame(
    a = sample(letters, 5, rep = TRUE),
    b = rnorm(5), c = rnorm(5)
  )
}
names(tmp) <- c("abc", "def", "ghi")
tmp

do.call("rbind", tmp)
dplyr::bind_rows(tmp, .id = "column_label")
plyr::ldply(tmp, data.frame, .id = "column_label")
rbind_df(tmp)
```

rsd

Relative standard deviation

Description

Calculate Relative Standard Deviation(RSD). RSD is also known as the coefficient of variation (CV)

Usage

```
rsd(x, na.rm = TRUE)
```

Arguments

x	an vector, matrix or data frame.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.

Details

Some notes:

- The higher the CV, the greater the dispersion in the variable.
- The CV is often presented as the given ratio multiplied by 100
- Basically CV<10 is very good, 10-20 is good, 20-30 is acceptable, and CV>30 is not acceptable.

Value

RSD value multiplied by 100.

Examples

```
library(ggplot2)
library(dplyr)
library(tidyr)
library(purrr)

data(iris)
rsd(iris[, 1:4])

## group rsd
val <- iris %>%
  group_by(Species) %>%
  group_modify(~ dat_summ(., method = rsd)) %>%
  pivot_longer(cols = !Species) %>% filter(!is.na(value))

ggplot(val, aes(x = value)) +
  geom_histogram(colour = "black", fill = "white") +
  facet_grid(Species ~ .)
ggplot(val, aes(x = Species, y = value)) + geom_boxplot()

## The plotting of missing value is similar like this.
```

rsd_filter

*Filtering variable based on relative standard deviation (RSD)***Description**

Filtering variable based on relative standard deviation (RSD).

Usage

```
rsd_filter(x, thres = 20)
```

Arguments

x a data frame where columns are features.

thres threshold of RSD. Features less than this threshold will be kept.

Value

a list of with contents:

- dat the filtered data matrix
- idx a logical vector of index for keeping features.

See Also

Other variable filters: [blank_filter\(\)](#), [locfdr_filter\(\)](#), [mv_filter\(\)](#), [qc_filter\(\)](#), [var_filter\(\)](#)

Examples

```
x <- matrix(rnorm(20 * 20), ncol = 20)
res <- rsd_filter(x, thres = 30)
dim(res$dat)
```

samp_strat	<i>Select random sample with stratification</i>
------------	---

Description

Select random sample with stratification from a binary group.

Usage

```
samp_strat(grp, len, strat = TRUE)
```

Arguments

grp	a character string indicating two group information.
len	length of random selection.
strat	a logical value indicated the sampling should be stratified.

Details

multiple group is not supported at the current stage.

Value

a selected index.

Examples

```
cls <- iris[, 5, drop = TRUE]
cls <- cls[cls == "setosa" | cls == "versicolor"]
cls <- droplevels(cls)
ind <- samp_strat(cls, 6, strat = TRUE)
cls[ind]
ind <- samp_strat(cls, 5, strat = TRUE)
cls[ind]
```

samp_sub	<i>Select random samples from each group Select random samples from each group.</i>
----------	---

Description

Select random samples from each group
 Select random samples from each group.

Usage

```
samp_sub(x, k, n = 10)
```

Arguments

x a character string or factor indicating class info.
 k number of samples selected from each class.
 n number of replicates.

Value

a selected index.

Examples

```
cls <- iris[, 5, drop = TRUE]
(tmp <- samp_sub(cls,k = 6, n = 2))
cls[tmp[[1]]]
table(cls[tmp[[1]])]
```

sym2long

Convert a symmetric table(short format) to long format

Description

Convert a symmetric table(short format) to long format

Usage

```
sym2long(x, tri = c("upper", "lower"))
```

Arguments

x a symmetric matrix-like data set.
 tri triangular being used.

Value

a data frame of pair-wise comparison.

Examples

```
co <- cor_tab(mtcars, cor.method = "spearman", adj.method = "BH")
names(co)
corr <- sym2long(co$r, tri = "upper")
pval <- sym2long(t(co$p), tri = "upper")
padj <- sym2long(co$p, tri = "upper")
tmp <- data.frame(corr, pval, padj)
```

trim_str	<i>Trim white spaces</i>
----------	--------------------------

Description

Trim head and tail white spaces of a string.

Usage

```
trim_str(string)
```

Arguments

string	a character string to be processed.
--------	-------------------------------------

Value

a trimmed string.

upd_data	<i>Update data set by a subset of features</i>
----------	--

Description

Update data set after feature selection.

Usage

```
upd_data(dat.all, ord)
```

Arguments

dat.all	a list of metabolomics data including intensity data, peak data and meta data
ord	a subset of selected feature orders

var_filter	<i>Filtering variable based on variability</i>
------------	--

Description

Perform variable filtering based on feature variability measurement such as standard deviation(SD) and inter-quantile range(IQR).

Usage

```
var_filter(x, method = "IQR", na.rm = FALSE, thres = 0.25)
```

Arguments

x	a matrix or data frame.
method	variability measurement method, such as IQR and sd.
na.rm	a logical value indicating whether NA values should be stripped before the computation proceeds.
thres	a numeric value between 0 and 1 for the threshold of quantile. Features whose variability value is large than this threshold will be kept.

Value

a list of with contents:

- dat the filtered data matrix
- idx a vector of filtering index.

See Also

Other variable filters: [blank_filter\(\)](#), [locfdr_filter\(\)](#), [mv_filter\(\)](#), [qc_filter\(\)](#), [rsd_filter\(\)](#)

Examples

```
set.seed(100)
x <- matrix(rnorm(20 * 10), ncol = 10)
res <- var_filter(x, method = "sd", thres = 0.25)
sum(res$idx)
```

vec2dat	<i>Convert a list of unequal vectors to a data frame</i>
---------	--

Description

Convert a list of unequal vectors to a data frame.

Usage

```
vec2dat(x)
```

Arguments

x a list of vector.

Value

a data matrix.

Examples

```
lst <- list(data.frame(a = 1, b = 2), data.frame(a = 2, c = 3, d = 5))
do.call(plyr::rbind.fill, lst)
vec <- list(var1 = c(a = 1, b = 2), var2 = c(a = 2, c = 3, d = 5))
vec2dat(vec)
```

vec_segment	<i>Vector statistics for error bar plotting</i>
-------------	---

Description

Calculate vector's standard derivation, standard error of mean and confidence interval.

Usage

```
vec_segment(x, bar = c("SD", "SE", "CI"))
```

Arguments

x an vector.
bar a character string, supporting "SD", "SE" and "CI".

Value

an vector including lower, center and upper values.

See Also

Other vector stats functions: [vec_stats\(\)](#)

Examples

```
library(plyr)
library(reshape2)
library(dplyr)
library(tidyr)
library(purrr)

vec_segment(iris[,1])

## Use 'plyr' and 'reshape2' for group
mat <- melt(iris)
ddply(mat, .(Species,variable), function(x,bar) {
  vec_segment(x$value, bar = bar)
}, bar = "SD")

## Use 'tidyverse' for group
iris %>%
  pivot_longer(cols = !Species, names_to = "variable") %>%
  group_by(Species, variable) %>%
  nest() %>%
  mutate(map_dfr(.x = data, .f = ~ vec_segment(.x$value))) %>%
  select(!data)

iris %>%
  pivot_longer(cols = !Species, names_to = "variable") %>%
  group_nest(Species, variable) %>%
  mutate(map_dfr(data, ~ vec_segment(.x$value))) %>%
  select(!data)
```

vec_stats

*Vector summary***Description**

Calculate the statistical summary of a vector.

Usage

```
vec_stats(x, na.rm = FALSE, conf.interval = 0.95)
```

Arguments

x a numeric vector.

na.rm remove NA or not.

conf.interval a numeric value for confidence interval.

Details

Can be used for error bar plotting. Modify from <https://bit.ly/3onsqot>

Value

a vector of summary consisting:

- number of vector length
- vector mean
- vector standard derivation
- standard error of mean
- confidence interval

See Also

Other vector stats functions: [vec_segment\(\)](#)

Examples

```
library(dplyr)
library(tidyr)
library(purrr)

iris %>% dat_summ(method = vec_stats)
iris %>% group_by(Species) %>% group_modify(~ dat_summ(., method = vec_stats))
iris %>%
  pivot_longer(cols = !Species, names_to = "var") %>%
  group_nest(Species, var) %>%
  mutate(map_dfr(data, ~ vec_stats(.x$value))) %>%
  select(!data)
```

vertex_stats	<i>Node descriptive metrics</i>
--------------	---------------------------------

Description

Return node descriptive metrics.

Usage

```
vertex_stats(g)
```

Arguments

g an igraph object.

Index

- * **multivariate**
 - pca_plot, 23
- * **outlier detectors**
 - outl_det_m, 21
 - outl_det_u, 22
- * **variable filters**
 - blank_filter, 5
 - locfdr_filter, 16
 - mv_filter, 18
 - qc_filter, 26
 - rsd_filter, 29
 - var_filter, 33
- * **vector stats functions**
 - vec_segment, 34
 - vec_stats, 35
- arrange_row, 2
- batch_shift, 3
- bi_cor_net, 4
- blank_filter, 5, 17, 19, 26, 29, 33
- chunk, 6
- cor_hcl, 6
- cor_net, 7
- cor_tab, 8
- cov.rob(), 21
- dat2long, 9
- dat_summ, 10
- dat_symb, 10
- dat_trans, 11
- df_na_idx, 12
- df_t, 12
- dim_mat, 13
- feat_count, 13
- gg_heat_dend, 14
- graph_stats, 15
- heat_dend, 15
- is_zero, 16
- locfdr(), 11, 17
- locfdr_filter, 5, 16, 19, 26, 29, 33
- locfdr_filter(), 10, 11
- mat2df, 18
- mv_filter, 5, 17, 18, 26, 29, 33
- mv_filter(), 19
- mv_perc, 19
- mv_perc(), 19
- net_graph, 20
- non_digit, 20
- non_zero, 21
- outl_det_m, 21, 22
- outl_det_u, 21, 22
- pca_plot, 23
- pcor_dat, 23
- plot_aam, 24
- plot_pval, 25
- ppcor::pcor(), 24
- qc_filter, 5, 17, 19, 26, 29, 33
- range_scale, 27
- rbind_df, 27
- rsd, 28
- rsd_filter, 5, 17, 19, 26, 29, 33
- samp_strat, 30
- samp_sub, 30
- stats::cor(), 7
- sym2long, 31
- trans(dat_trans), 11
- trim_str, 32
- upd_data, 32
- var_filter, 5, 17, 19, 26, 29, 33
- vec2dat, 34
- vec_segment, 34, 36
- vec_stats, 34, 35
- vec_trans(dat_trans), 11
- vertex_stats, 36