

Package ‘pcutils’

January 9, 2024

Type Package

Title Some Useful Functions for Statistics and Visualization

Version 0.2.1

Description Offers a range of utilities and functions for everyday programming tasks.

- 1.Data Manipulation. Such as grouping and merging, column splitting, and character expansion.
- 2.File Handling. Read and convert files in popular formats, including ``blast'', ``diamond'', ``fasta'', ``gff'', ``gtf'', and various image formats like ``jpg'', ``png'', ``pdf'', and ``svg''.
- 3.Plotting Assistance. Helpful utilities for generating color palettes, validating color formats, and adding transparency.
- 4.Statistical Analysis. Includes functions for pairwise comparisons and multiple testing corrections, enabling perform statistical analyses with ease.
- 5.Graph Plotting, Provides efficient tools for creating doughnut plot and multi-layered doughnut plot; Venn diagrams, including traditional Venn diagrams, upset plots, and flower plots; Simplified functions for creating stacked bar plots, or a box plot with alphabets group for multiple comparison group.

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Depends R (>= 4.1.0)

Imports dplyr, magrittr, ggplot2, stats, utils, grDevices, reshape2, scales, agricolae, tools, graphics

Suggests clipr, remotes, BiocManager, ggpubr, kableExtra, animation, htmlwidgets, jpeg, pagedown, png, tidyr, RColorBrewer, ggsci, readr, grImport2, pdftools, rsvg, PMCMRplus, nortest, fitdistrplus, ggalluvial, gghalves, ggspatial, sf, ggimage, ggpmisc, ggvenn, UpSetR, plotrix, RBGL, vegan, ggtern, wordcloud2, circlize, RIdeogram, igraph, tibble, knitr, rmarkdown, prettydoc, ggnewscale, plotly, htmltools, leaflet, relaimpo, snow, doSNOW, foreach, stringr, ggraph, ggrepel, treemap, voronoiTreemap, devtools, multcompView, plot3D, fanyi, rio, bookdown, sysfonts, showtext, biomformat, ggtree, aplot, ape, ggforce

VignetteBuilder knitr

BugReports <https://github.com/Asa12138/pcutils/issues>

URL <https://github.com/Asa12138/pcutils>

Date/Publication 2024-01-09 10:40:02 UTC

NeedsCompilation no

Author Chen Peng [aut, cre] (<<https://orcid.org/0000-0002-9449-7606>>)

Maintainer Chen Peng <pengchen2001@zju.edu.cn>

Repository CRAN

R topics documented:

add_alpha	4
add_analysis	4
add_theme	5
change_fac_lev	6
china_map	6
chunlian	7
copy_df	8
copy_vector	8
count2	9
dabiao	9
del_ps	10
df2link	11
DNA_plot	11
download2	12
explode	12
fittest	13
generate_labels	13
get_cols	14
get_doi	15
ggheatmap	15
gghist	17
gghuan	17
gghuan2	18
ggplot_lim	19
give_you_a_rose	20
grepl.data.frame	20
group_box	21
group_test	22
gsub.data.frame	23
guolv	24
hebing	24
how_to_set_options	25
how_to_update_parameters	25
how_to_use_parallel	26

how_to_use_sbatch	26
is.ggplot.color	27
legend_size	27
lib_ps	28
little_guodong	28
lm_coefficients	29
make_gitbook	29
make_project	30
metadata	31
mmscale	31
multireg	32
multitest	32
my_cat	33
my_circle_packing	33
my_circo	34
my_lm	35
my_sunburst	36
my_syteny	37
my_treemap	37
my_voronoi_treemap	38
Olympic_rings	38
otutab	39
plot.coefficients	39
plotgif	40
plotpdf	40
pre_number_str	41
read.file	41
read_fasta	42
reinstall_my_packages	42
remove.outliers	43
rgb2code	43
rm_low	44
sample_map	44
sanxian	46
search_browse	47
set_pcutils_config	48
show_pcutils_config	48
squash	49
stackplot	49
strsplit2	51
t2	52
taxonomy	52
tax_pie	53
tax_wordcloud	53
tidai	54
trans	55
translator	55
trans_format	56

triangp	57
twotest	57
update_param	58
venn	58
write_fasta	59

Index	60
--------------	-----------

add_alpha	<i>Add alpha for a Rcolor</i>
-----------	-------------------------------

Description

Add alpha for a Rcolor

Usage

```
add_alpha(color, alpha = 0.3)
```

Arguments

color	Rcolor
alpha	alpha, default 0.3

Value

8 hex color

Examples

```
add_alpha("red", 0.3)
```

add_analysis	<i>Add an analysis for a project</i>
--------------	--------------------------------------

Description

Add an analysis for a project

Usage

```
add_analysis(analysis_n, title = analysis_n, author = "Asa12138", theme = 1)
```

Arguments

<code>analysis_n</code>	analysis name
<code>title</code>	file title
<code>author</code>	author
<code>theme</code>	1~10

Value

No return value

<code>add_theme</code>	<i>Add a global gg_theme and colors for plots</i>
------------------------	---

Description

Add a global gg_theme and colors for plots

Usage

```
add_theme(set_theme = NULL)
```

Arguments

<code>set_theme</code>	your theme
------------------------	------------

Value

No return value

Examples

```
add_theme()
```

change_fac_lev	<i>Change factor levels</i>
----------------	-----------------------------

Description

Change factor levels

Usage

```
change_fac_lev(x, levels = NULL, last = FALSE)
```

Arguments

x	vector
levels	custom levels
last	put the custom levels to the last

Value

factor

Examples

```
change_fac_lev(letters[1:5], levels = c("c", "a"))
```

china_map	<i>Plot china map</i>
-----------	-----------------------

Description

Plot china map

Usage

```
china_map(china_shp = NULL, download_dir = "pcutils_temp")
```

Arguments

china_shp	china.json file
download_dir	download_dir, "pcutils_temp"

Value

a ggplot

`chunlian`*Draw a Chunlian (Spring Festival couplet) using ggplot2*

Description

Draw a Chunlian (Spring Festival couplet) using ggplot2

Usage

```
chunlian(  
  words = NULL,  
  bg_size = 20,  
  bg_shape = 22,  
  bg_fill = "red2",  
  text_size = 10,  
  text_params = list(),  
  font_file = NULL,  
  download_dir = "pcutils_temp"  
)
```

Arguments

<code>words</code>	A character vector containing three strings for the three lines of the couplet
<code>bg_size</code>	Size of the points in <code>geom_point</code> , 20
<code>bg_shape</code>	Shape of the points in <code>geom_point</code> (21~25), 22 or 23 are very good.
<code>bg_fill</code>	Fill color of the points in <code>geom_point</code>
<code>text_size</code>	Size of the text in <code>geom_text</code> , 10
<code>text_params</code>	parameters parse to <code>geom_text</code>
<code>font_file</code>	font file, e.g XX.ttf, XX.ttc
<code>download_dir</code>	download_dir for font_file

Value

A ggplot object representing the Chunlian

copy_df	<i>Copy a data.frame</i>
---------	--------------------------

Description

Copy a data.frame

Usage

```
copy_df(df)
```

Arguments

df a R data.frame object

Value

No return value

copy_vector	<i>Copy a vector</i>
-------------	----------------------

Description

Copy a vector

Usage

```
copy_vector(vec)
```

Arguments

vec a R vector object

Value

No return value

count2	<i>Like uniq -c in shell to count a vector</i>
--------	--

Description

Like `uniq -c` in shell to count a vector

Usage

```
count2(df)
```

Arguments

df two columns: first is type, second is number

Value

two columns: first is type, second is number

Examples

```
count2(data.frame(group = c("A", "A", "B", "C", "C", "A"), value = c(2, 2, 2, 1, 3, 1)))
```

dabiao	<i>Print some message with =</i>
--------	----------------------------------

Description

Print some message with =

Usage

```
dabiao(  
  str = "",  
  ...,  
  n = 80,  
  char = "=",  
  mode = c("middle", "left", "right"),  
  print = FALSE  
)
```

Arguments

str	output strings
...	strings will be paste together
n	the number of output length
char	side chars default:=
mode	"middle", "left" or "right"
print	print or message?

Value

No return value

Examples

```
dabiao("Start running!")
```

del_ps

Detach packages

Description

Detach packages

Usage

```
del_ps(p_list, ..., origin = NULL)
```

Arguments

p_list	a vector of packages list
...	packages
origin	keep the original Namespace

Value

No return value

df2link	<i>df 2 link</i>
---------	------------------

Description

df 2 link

Usage

```
df2link(test, fun = sum)
```

Arguments

test	df
fun	function to summary the elements number, defalut: sum, you can choose mean.

Value

data.frame

Examples

```
data(otutab)
cbind(taxonomy, num = rowSums(otutab))[1:10, ] -> test
df2link(test)
```

DNA_plot	<i>Plot a DNA double helix</i>
----------	--------------------------------

Description

Plot a DNA double helix

Usage

```
DNA_plot(
  col_DNA = "#377EB8",
  col_ATCG = c("#7FC97F", "#FB8072", "#FFFFB3", "#A6CEE3"),
  DNA_length = 2
)
```

Arguments

col_DNA	col_DNA, "#377EB8"
col_ATCG	col_ATCG, c("#7FC97F", "#FB8072", "#FFFFB3", "#A6CEE3")
DNA_length	DNA_length, 2

Value

ggplot

Referenceshttps://github.com/SherryDong/create_plot_by_R_base**Examples**

DNA_plot()

`download2`*Download File*

Description

This function downloads a file from the provided URL and saves it to the specified location.

Usage

```
download2(url, file_path, timeout = 300, force = FALSE)
```

Arguments

<code>url</code>	The URL from which to download the file.
<code>file_path</code>	The full path to the file.
<code>timeout</code>	timeout, 300s
<code>force</code>	FALSE, if TRUE, overwrite existed file

`explode`*Explode a data.frame if there are split charter in one column*

Description

Explode a data.frame if there are split charter in one column

Usage

```
explode(df, column, split = ",")
```

Arguments

<code>df</code>	data.frame
<code>column</code>	column
<code>split</code>	split string

Value

data.frame

Examples

```
df <- data.frame(a = 1:2, b = c("a,b", "c"), c = 3:4)
explode(df, "b", ",")
```

fittest	<i>Fit a distribution</i>
---------	---------------------------

Description

Fit a distribution

Usage

```
fittest(a)
```

Arguments

a a numeric vector

Value

distribution

generate_labels	<i>Generate labels position</i>
-----------------	---------------------------------

Description

Generate labels position

Usage

```
generate_labels(
  labels = NULL,
  input = c(0, 0),
  nrows = NULL,
  ncols = NULL,
  x_offset = 0.3,
  y_offset = 0.15,
  just = 1
)
```

Arguments

labels	labels
input	c(0,0)
nrows	default: NULL
ncols	default: NULL
x_offset	0.3
y_offset	0.15
just	0~5

Value

matrix

Examples

```
library(ggplot2)
labels <- vapply(1:8, \i)paste0(sample(LETTERS, 4), collapse = ""), character(1))
df <- data.frame(label = labels, generate_labels(labels))
ggplot(data = df) +
  geom_label(aes(x = X1, y = X2, label = label))
```

get_cols

Get n colors

Description

Get n colors

Usage

```
get_cols(n = 11, pal = "col1")
```

Arguments

n	how many colors you need
pal	col1~3; or a vector of colors, you can get from: RColorBrewer::brewer.pal(5, "Set2") or ggsci::pal_aaas()(5)

Value

a vector of n colors

Examples

```
get_cols(10, "col2") -> my_cols
scales::show_col(my_cols)

scales::show_col(get_cols(15, RColorBrewer::brewer.pal(5, "Set2")))
scales::show_col(get_cols(15, ggsci::pal_aaas()(5)))
```

`get_doi`*Download supplemental materials according to a doi*

Description

Download supplemental materials according to a doi

Usage

```
get_doi(
  doi,
  dir = "~/Downloads/",
  bget_path = "~/software/bget_0.3.2_Darwin_64-bit/bget"
)
```

Arguments

<code>doi</code>	doi
<code>dir</code>	dir
<code>bget_path</code>	your bget_path

Value

file at work directory

`ggheatmap`*Heatmap by ggplot*

Description

Heatmap by ggplot

Usage

```
ggheatmap(  
  otutab,  
  pal = NULL,  
  scale = "none",  
  rowname = TRUE,  
  colname = TRUE,  
  row_cluster = TRUE,  
  col_cluster = TRUE,  
  row_annotation = NULL,  
  col_annotation = NULL,  
  annotation_pal = NULL  
)
```

Arguments

otutab	otutab
pal	the main color pal, a vector of colors
scale	"none", "row", "column"
rowname	show row names?
colname	show column names?
row_cluster	cluster the row?
col_cluster	cluster the column?
row_annotation	row annotation
col_annotation	column annotation
annotation_pal	the annotation color pal, a list. e.g. list(Group=c("red","blue"))

Value

a ggplot

Examples

```
data(otutab)  
ggheatmap(otutab[1:30, ],  
  scale = "row", row_annotation = otutab[1:30, 1:2],  
  col_annotation = metadata[, c(2, 4)]  
)
```

gghist	<i>gg Histogram</i>
--------	---------------------

Description

gg Histogram

Usage

```
gghist(x, ...)
```

Arguments

x	vector
...	parameters parse to gghistogram

Value

ggplot

Examples

```
gghist(rnorm(100))
```

gghuan	<i>Plot a doughnut chart</i>
--------	------------------------------

Description

Plot a doughnut chart

Usage

```
gghuan(  
  tab,  
  reorder = TRUE,  
  mode = "1",  
  topN = 5,  
  name = TRUE,  
  percentage = TRUE,  
  bar_params = NULL,  
  text_params = NULL,  
  text_params2 = NULL  
)
```

Arguments

tab	two columns: first is type, second is number
reorder	reorder by number?
mode	plot style, 1~3
topN	plot how many top items
name	label the name
percentage	label the percentage
bar_params	parameters parse to <code>geom_rect</code> , for mode=1,3 or <code>geom_col</code> for mode=2.
text_params	parameters parse to <code>geom_text</code>
text_params2	parameters parse to <code>geom_text</code> , for name=TRUE & mode=1,3

Value

a ggplot

Examples

```
a <- data.frame(type = letters[1:6], num = c(1, 3, 3, 4, 5, 10))
gghuan(a) + ggplot2::scale_fill_manual(values = get_cols(6, "col3"))
gghuan(a,
  bar_params = list(col = "black"),
  text_params = list(col = "#b15928", size = 3),
  text_params2 = list(col = "#006d2c", size = 5)
) +
  ggplot2::scale_fill_manual(values = get_cols(6, "col3"))
gghuan(a, mode = 2) + ggplot2::scale_fill_manual(values = get_cols(6, "col3"))
gghuan(a, mode = 3) + ggplot2::scale_fill_manual(values = get_cols(6, "col3"))
```

gghuan2

gghuan2 for multi-doughnut chart

Description

gghuan2 for multi-doughnut chart

Usage

```
gghuan2(
  tab = NULL,
  space_width = 0.2,
  name = TRUE,
  percentage = FALSE,
  text_params = NULL,
  bar_params = NULL
)
```

Arguments

tab	a dataframe with hierarchical structure
space_width	the space width between doughnuts (0~1).
name	label the name
percentage	label the percentage
text_params	parameters parse to geom_text
bar_params	parameters parse to geom_rect

Value

a ggplot

Examples

```
data.frame(  
  a = c("a", "a", "b", "b", "c"), b = c("a", LETTERS[2:5]), c = rep("a", 5),  
  number = 1:5  
) %>% gghuan2()
```

ggplot_lim

Get a ggplot xlim and ylim

Description

Get a ggplot xlim and ylim

Usage

```
ggplot_lim(p)
```

Arguments

p	ggplot
---	--------

Value

list

give_you_a_rose *Give you a rose*

Description

Give you a rose

Usage

```
give_you_a_rose(color = "red3")
```

Arguments

color "skyblue3"

Value

plot

References

<https://mp.weixin.qq.com/s/W-BYPR3UXL120XWpTmN3rA>

grepl.data.frame *Grepl applied on a data.frame*

Description

Grepl applied on a data.frame

Usage

```
grepl.data.frame(pattern, x, ...)
```

Arguments

pattern search pattern
x your data.frame
... additional arguments for gerpl()

Value

a logical data.frame

Examples

```
matrix(letters[1:6], 2, 3) |> as.data.frame() -> a
grepl.data.frame("c", a)
grepl.data.frame("\\w", a)
```

group_box

*Plot a boxplot***Description**

Plot a boxplot

Usage

```
group_box(
  tab,
  group = NULL,
  metadata = NULL,
  mode = 1,
  group_order = NULL,
  facet_order = NULL,
  alpha = FALSE,
  method = "wilcox",
  alpha_param = list(color = "red"),
  point_param = NULL,
  p_value1 = FALSE,
  p_value2 = FALSE,
  only_sig = TRUE,
  stat_compare_means_param = NULL,
  trend_line = FALSE,
  trend_line_param = list(color = "blue")
)
```

Arguments

tab	your dataframe
group	which colname choose for group or a vector
metadata	the dataframe contains the group
mode	1~3, plot style
group_order	the order of x group
facet_order	the order of the facet
alpha	whether plot a group alphabeta by test of method
method	test method:wilcox, tukeyHSD, LSD, (default: wilcox), see multitest
alpha_param	parameters parse to geom_text

point_param parameters parse to [geom_jitter](#)
 p_value1 multi-test of all group
 p_value2 two-test of each pair
 only_sig only_sig for p_value2
 stat_compare_means_param
 parameters parse to [stat_compare_means](#)
 trend_line add a trend line
 trend_line_param
 parameters parse to [geom_smooth](#)

Value

a ggplot

Examples

```
a <- data.frame(a = 1:18, b = runif(18, 0, 5))
group_box(a, group = rep(c("a", "b", "c"), each = 6))
```

group_test

Performs multiple mean comparisons for a data.frame

Description

Performs multiple mean comparisons for a data.frame

Usage

```
group_test(
  df,
  group,
  metadata = NULL,
  method = "wilcox.test",
  threads = 1,
  p.adjust.method = "BH",
  verbose = TRUE
)
```

Arguments

df a data.frame
 group The compare group (categories) in your data, one column name of metadata when metadata exist or a vector whose length equal to columns number of df.
 metadata sample information dataframe contains group
 method the type of test. Default is wilcox.test. Allowed values include:

- `t.test` (parametric) and `wilcox.test` (non-parametric). Perform comparison between two groups of samples. If the grouping variable contains more than two levels, then a pairwise comparison is performed.
- `anova` (parametric) and `kruskal.test` (non-parametric). Perform one-way ANOVA test comparing multiple groups.

threads default 1
 p.adjust.method p.adjust.method, see `p.adjust`, default BH.
 verbose logical

Value

data.frame

Examples

```
data(otutab)
group_test(otutab, metadata$Group, method = "kruskal.test")
group_test(otutab[, 1:12], metadata$Group[1:12], method = "wilcox.test")
```

`gsub.data.frame` *Gsub applied on a data.frame*

Description

Gsub applied on a data.frame

Usage

```
gsub.data.frame(pattern, replacement, x, ...)
```

Arguments

pattern search pattern
 replacement a replacement for matched pattern
 x your data.frame
 ... additional arguments for `gerpl()`

Value

a logical data.frame

Examples

```
matrix(letters[1:6], 2, 3) |> as.data.frame() -> a
gsub.data.frame("c", "a", a)
```

 guolv

Filter your data

Description

Filter your data

Usage

```
guolv(tab, sum = 10, exist = 1)
```

Arguments

tab	dataframe
sum	the rowsum should bigger than sum(default:10)
exist	the exist number bigger than exist(default:1)

Value

input object

Examples

```
data(otutab)
guolv(otutab)
```

hebing

Group your data

Description

Group your data

Usage

```
hebing(otutab, group, margin = 2, act = "mean")
```

Arguments

otutab	data.frame
group	group vector
margin	1 for row and 2 for column(default: 2)
act	do (default: mean)

Value

data.frame

Examples

```
data(otutab)
hebing(otutab, metadata$Group)
```

how_to_set_options *How to set options in a package*

Description

How to set options in a package

Usage

```
how_to_set_options(package = "My_package")
```

Arguments

package package name

Value

No return value

how_to_update_parameters
How to update parameters

Description

How to update parameters

Usage

```
how_to_update_parameters()
```

Value

No return value

how_to_use_parallel *How to use parallel*

Description

How to use parallel

Usage

```
how_to_use_parallel(  
  loop = function(i) {  
    return(mean(rnorm(100)))  
  }  
)
```

Arguments

loop the main function

Value

No return value

how_to_use_sbatch *How to use sbatch*

Description

How to use sbatch

Usage

```
how_to_use_sbatch(mode = 1)
```

Arguments

mode 1~3

Value

No return value

is.ggplot.color	<i>Judge if a characteristic is Rcolor</i>
-----------------	--

Description

Judge if a characteristic is Rcolor

Usage

```
is.ggplot.color(color)
```

Arguments

color	characteristic
-------	----------------

Value

TRUE or FALSE

Examples

```
is.ggplot.color("red")  
is.ggplot.color("notcolor")  
is.ggplot.color(NA)  
is.ggplot.color("#000")
```

legend_size	<i>Scale a legend size</i>
-------------	----------------------------

Description

Scale a legend size

Usage

```
legend_size(scale = 1)
```

Arguments

scale	default: 1.
-------	-------------

Value

"theme" "gg"

lib_ps	<i>Attach packages or install packages have not benn installed</i>
--------	--

Description

Attach packages or install packages have not benn installed

Usage

```
lib_ps(p_list, ..., all_yes = FALSE, library = TRUE)
```

Arguments

p_list	a vector of packages list
...	packages
all_yes	all install try set to yes?
library	should library the package or just get Namespace ?

Value

No return value

little_guodong	<i>My cat.</i>
----------------	----------------

Description

my little cat named Guo Dong which drawn by my girlfriend.

Format

rastergrob object.

lm_coefficients	<i>Get coefficients of linear regression model</i>
-----------------	--

Description

This function fits a linear regression model using the given data and formula, and returns the coefficients.

Usage

```
lm_coefficients(data, formula, each = TRUE)
```

Arguments

data	A data frame containing the response variable and predictors.
formula	A formula specifying the structure of the linear regression model.
each	each variable do a lm or whole multi-lm

Value

coefficients The coefficients of the linear regression model.

Examples

```
data <- data.frame(
  response = c(2, 4, 6, 7, 9),
  x1 = c(1, 2, 3, 4, 5),
  x2 = c(2, 3, 6, 8, 9),
  x3 = c(3, 6, 5, 12, 12)
)
coefficients_df <- lm_coefficients(data, response ~ x1 + x2 + x3)
print(coefficients_df)
plot(coefficients_df)
```

make_gitbook	<i>Make a Gitbook using bookdown</i>
--------------	--------------------------------------

Description

Make a Gitbook using bookdown

Usage

```
make_gitbook(
  book_n,
  root_dir = "~/Documents/R/",
  mode = c("gitbook", "bs4")[1],
  author = "Asa12138",
  bib = "~/Documents/R/pc_blog/content/bib/My Library.bib",
  cs1 = "~/Documents/R/pc_blog/content/bib/science.cs1"
)
```

Arguments

book_n	project name
root_dir	root directory
mode	"gitbook","bs4"
author	author
bib	cite papers bib, from Zotero
cs1	cite papers format, default science.cs1

Value

No return value

make_project	<i>Make a R-analysis project</i>
--------------	----------------------------------

Description

Make a R-analysis project

Usage

```
make_project(
  pro_n,
  root_dir = "~/Documents/R/",
  bib = "~/Documents/R/pc_blog/content/bib/My Library.bib",
  cs1 = "~/Documents/R/pc_blog/content/bib/science.cs1"
)
```

Arguments

pro_n	project name
root_dir	root directory
bib	cite papers bib, from Zotero
cs1	cite papers format, default science.cs1

Value

No return value

metadata	<i>test data for pcutils package.</i>
----------	---------------------------------------

Description

an otutab, metadata and a taxonomy table.

Format

contains an otutab, metadata and a taxonomy table.

otutab contains otutable rawdata

metadata contains metadata

taxonomy contains taxonomy table

mmscale	<i>Min_Max scale</i>
---------	----------------------

Description

Min_Max scale

Usage

```
mmscale(x, min_s = 0, max_s = 1, n = 1, plot = FALSE)
```

Arguments

x	a numeric vector
min_s	scale min
max_s	scale max
n	linear transfer for n=1; the slope will change if n>1 or n<1
plot	whether plot the transfer?

Value

a numeric vector

Examples

```
x <- runif(10)
mmscale(x, 5, 10)
```

multireg	<i>Multiple regression/ variance decomposition analysis</i>
----------	---

Description

Multiple regression/ variance decomposition analysis

Usage

```
multireg(formula, data, TopN = 3)
```

Arguments

formula	formula
data	dataframe
TopN	give top variable importance

Value

ggplot

Examples

```
data(otutab)
multireg(env1 ~ Group * ., data = metadata[, 2:7])
```

multitest	<i>Multi-groups test</i>
-----------	--------------------------

Description

anova (parametric) and kruskal.test (non-parametric). Perform one-way ANOVA test comparing multiple groups. LSD and TukeyHSD are post hoc test of anova. dunn and nemenyi are post hoc test of kruskal.test. t.test or wilcox is just perform t.test or wilcox.test in each two group (no p.adjust).

Usage

```
multitest(var, group, print = TRUE, return = FALSE)
```

Arguments

var	numeric vector
group	more than two-levels group vector
print	whether print the result
return	return which method result (tukeyHSD or LSD or wilcox?)

Value

No value or a dataframe.

Examples

```
multitest(runif(30), rep(c("A", "B", "C"), each = 10), print = FALSE, return = "wilcox") -> aa
```

my_cat	<i>Show my little cat named Guo Dong which drawn by my girlfriend.</i>
--------	--

Description

Show my little cat named Guo Dong which drawn by my girlfriend.

Usage

```
my_cat(mode = 1)
```

Arguments

mode	1~2
------	-----

Value

a ggplot

my_circle_packing	<i>My Circle packing plot</i>
-------------------	-------------------------------

Description

My Circle packing plot

Usage

```
my_circle_packing(  
  test,  
  anno = NULL,  
  mode = 1,  
  Group = "level",  
  Score = "weight",  
  label = "label",  
  show_level_name = "all",  
  show_tip_label = TRUE,  
  str_width = 10  
)
```

Arguments

test	a dataframe with hierarchical structure
anno	annotation table with rowname for color or fill.
mode	1~2
Group	fill for mode2
Score	color for mode1
label	the labels column
show_level_name	show which level name? a vector contains some column names.
show_tip_label	show_tip_label, logical
str_width	str_width

Value

ggplot

Examples

```
data(otutab)
cbind(taxonomy, weight = rowSums(otutab))[1:10, ] -> test
my_circle_packing(test)
```

my_circo

My circo plot

Description

My circo plot

Usage

```
my_circo(
  df,
  reorder = TRUE,
  pal = NULL,
  mode = c("circlize", "chorddiag")[1],
  ...
)
```

Arguments

df	dataframe with three column
reorder	reorder by number?
pal	a vector of colors, you can get from here too: <code>RColorBrewer::brewer.pal(5, "Set2")</code> or <code>ggsci::pal_aaas()(5)</code>
mode	"circlize", "chorddiag"
...	chordDiagram

Value

chordDiagram

Examples

```
data.frame(
  a = c("a", "a", "b", "b", "c"),
  b = c("a", LETTERS[2:5]), c = 1:5
) %>% my_circo(mode = "circlize")
data(otutab)
cbind(taxonomy, num = rowSums(otutab))[1:10, c(2, 6, 8)] -> test
my_circo(test)
```

my_lm

Fit a linear model and plot

Description

Fit a linear model and plot

Usage

```
my_lm(tab, var, metadata = NULL, lm_color = "red", ...)
```

Arguments

tab	your dataframe
var	which colname choose for var or a vector
metadata	the dataframe contains the var
lm_color	"red"
...	parameters parse to geom_point

Value

a ggplot

Examples

```
my_lm(runif(50), var = 1:50)
my_lm(c(1:50) + runif(50, 0, 5), var = 1:50)
```

my_sunburst

My Sunburst plot

Description

My Sunburst plot

Usage

```
my_sunburst(test, ...)
```

Arguments

test a dataframe with hierarchical structure
... look for parameters in [plot_ly](#)

Value

htmlwidget

Examples

```
data(otutab)
cbind(taxonomy, num = rowSums(otutab))[1:10, ] -> test
my_sunburst(test)
```

my_synteny	<i>My synteny plot</i>
------------	------------------------

Description

My synteny plot

Usage

```
my_synteny()
```

Value

plot

my_treemap	<i>My Treemap plot</i>
------------	------------------------

Description

My Treemap plot

Usage

```
my_treemap(test, ...)
```

Arguments

test	a three-columns dataframe with hierarchical structure
...	look for parameters in plot_ly

Value

htmlwidget

Examples

```
data(otutab)
cbind(taxonomy, num = rowSums(otutab))[1:10, c(4, 7, 8)] -> test
my_treemap(test)
```

my_voronoi_treemap *My Voronoi treemap plot*

Description

My Voronoi treemap plot

Usage

```
my_voronoi_treemap(test, ...)
```

Arguments

test a three-columns dataframe with hierarchical structure
... look for parameters in [vt_d3](#)

Value

htmlwidget

Examples

```
data(otutab)  
cbind(taxonomy, num = rowSums(otutab))[1:10, c(4, 7, 8)] -> test  
my_voronoi_treemap(test)
```

Olympic_rings *Plot the Olympic rings*

Description

Plot the Olympic rings

Usage

```
Olympic_rings()
```

Value

ggplot

Examples

```
Olympic_rings()
```

otutab	<i>test data for pcutils package.</i>
--------	---------------------------------------

Description

an otutab, metadata and a taxonomy table.

Format

contains an otutab, metadata and a taxonomy table.

otutab contains otutable rawdata

metadata contains metadata

taxonomy contains taxonomy table

plot.coefficients	<i>Plot coefficients as a bar chart or lollipop chart</i>
-------------------	---

Description

This function takes the coefficients and generates a plot to visualize their magnitudes.

Usage

```
## S3 method for class 'coefficients'
plot(x, mode = 1, number = FALSE, x_order = NULL, ...)
```

Arguments

x	The coefficients to be plotted.
mode	The mode of the plot: 1 for bar chart, 2 for lollipop chart.
number	show number
x_order	order of variables
...	add

Value

ggplot

plotgif *Plot a gif*

Description

Plot a gif

Usage

```
plotgif(plist, file, mode = "gif")
```

Arguments

plist	plot list
file	prefix of your .gif file
mode	"gif" or "html"

Value

No return value

plotpdf *Plot a multi-pages pdf*

Description

Plot a multi-pages pdf

Usage

```
plotpdf(
  plist,
  file,
  width = 8,
  height = 7,
  brower = "/Applications/Microsoft Edge.app/Contents/MacOS/Microsoft Edge",
  ...
)
```

Arguments

plist	plot list
file	prefix of your .pdf file
width	width
height	height
brower	the path of Google Chrome, Microsoft Edge or Chromium in your computer.
...	additional arguments

Value

No return value

pre_number_str	<i>Prepare a numeric string</i>
----------------	---------------------------------

Description

Prepare a numeric string

Usage

```
pre_number_str(str, split_str = ",", continuous_str = "-")
```

Arguments

str	a string contain ',' and '-'
split_str	split_str ","
continuous_str	continuous_str "-"

Value

vector

Examples

```
pre_number_str("a1,a3,a5,a6-a10")
```

read.file	<i>Read some special format file</i>
-----------	--------------------------------------

Description

Read some special format file

Usage

```
read.file(file, format = NULL, just_print = FALSE, all_yes = FALSE)
```

Arguments

file	file path
format	"blast", "diamond", "fa", "fasta", "fna", "gff", "gtf", "jpg", "png", "pdf", "svg"...
just_print	just print the file
all_yes	all_yes?

Value

data.frame

read_fasta	<i>Read fasta file</i>
------------	------------------------

Description

Read fasta file

Usage

```
read_fasta(fasta_file)
```

Arguments

fasta_file file path

Value

data.frame

reinstall_my_packages	<i>Re-install my packages</i>
-----------------------	-------------------------------

Description

Re-install my packages

Usage

```
reinstall_my_packages(pkgs = c("pcutils", "pctax", "MetaNet", "ReporterScore"))
```

Arguments

pkgs pkgs

Value

No return value

remove.outliers	<i>Remove outliers</i>
-----------------	------------------------

Description

Remove outliers

Usage

```
remove.outliers(x, factor = 1.5)
```

Arguments

x	a numeric vector
factor	default 1.5

Value

a numeric vector

Examples

```
remove.outliers(c(1, 10:15))
```

rgb2code	<i>Transform a rgb vector to a Rcolor code</i>
----------	--

Description

Transform a rgb vector to a Rcolor code

Usage

```
rgb2code(x, rev = FALSE)
```

Arguments

x	vector or three columns data.frame
rev	reverse, transform a Rcolor code to a rgb vector

Value

Rcolor code like "#69C404"

Examples

```
rgb2code(c(12, 23, 34))  
rgb2code("#69C404", rev = TRUE)
```

rm_low	<i>Remove the low relative items in each column</i>
--------	---

Description

Remove the low relative items in each column

Usage

```
rm_low(otutab, relative_threshold = 0.0001)
```

Arguments

otutab	otutab
relative_threshold	threshold, default: 1e-4

Value

data.frame

Examples

```
data(otutab)
rm_low(otutab)
```

sample_map	<i>Plot the sampling map</i>
------------	------------------------------

Description

Plot the sampling map

Usage

```
sample_map(
  metadata,
  mode = 1,
  map_params = list(),
  group = NULL,
  point_params = list(),
  label = NULL,
  label_params = list(),
  shp_file = NULL,
  crs = NULL,
  xlim = NULL,
```

```

ylim = NULL,
add_scale = TRUE,
scale_params = list(),
add_north_arrow = TRUE,
north_arrow_params = list()
)

```

Arguments

metadata	metadata must contains "Longitude", "Latitude"
mode	1~3. 1 use basic data from ggplot2. 2 use a shp_file. 3 use the leaflet.
map_params	parameters parse to geom_polygon (mode=1) or geom_sf (mode=2)
group	one column name of metadata which mapping to point color
point_params	parameters parse to geom_point
label	one column name of metadata which mapping to point label
label_params	parameters parse to geom_sf_text
shp_file	a geojson file parse to sf::read_sf
crs	crs coordinate: https://asa-blog.netlify.app/p/r-map/#crs
xlim	xlim
ylim	ylim
add_scale	add annotation_scale
scale_params	parameters parse to ggspatial::annotation_scale
add_north_arrow	add annotation_north_arrow
north_arrow_params	parameters parse to ggspatial::annotation_north_arrow

Value

map

Examples

```

data(otutab)
anno_df <- metadata[, c("Id", "long", "lat", "Group")]
colnames(anno_df) <- c("Id", "Longitude", "Latitude", "Group")
sample_map(anno_df, mode = 1, group = "Group", xlim = c(90, 135), ylim = c(20, 50))

```

`sanxian`*Three-line table*

Description

Three-line table

Usage

```
sanxian(  
  df,  
  digits = 3,  
  nrow = 10,  
  ncol = 10,  
  fig = FALSE,  
  mode = 1,  
  background = "#D7261E",  
  ...  
)
```

Arguments

<code>df</code>	a data.frame
<code>digits</code>	how many digits should remain
<code>nrow</code>	show how many rows
<code>ncol</code>	show how many columns
<code>fig</code>	output as a figure
<code>mode</code>	1~2
<code>background</code>	background color
<code>...</code>	additional arguments e.g.(rows=NULL)

Value

a ggplot

Examples

```
data(otutab)  
sanxian(otutab)
```

search_browse	<i>Search and browse the web for specified terms</i>
---------------	--

Description

This function takes a vector of search terms, an optional search engine (default is Google), and an optional base URL to perform web searches. It opens the default web browser with search results for each term.

Usage

```
search_browse(search_terms, engine = "google", base_url = NULL)
```

Arguments

search_terms	A character vector of search terms to be searched.
engine	A character string specifying the search engine to use (default is "google"). Supported engines: "google", "bing".
base_url	A character string specifying the base URL for web searches. If not provided, the function will use a default URL based on the chosen search engine.

Value

No return value

Examples

```
## Not run:
search_terms <- c(
  "s__Pandoraea_pnomenus",
  "s__Alicycliphilus_sp._B1"
)

# Using Google search engine
search_browse(search_terms, engine = "google")

# Using Bing search engine
search_browse(search_terms, engine = "bing")

## End(Not run)
```

set_pcutils_config *Set config*

Description

Set config

Usage

```
set_pcutils_config(item, value)
```

Arguments

item	item
value	value

Value

No value

show_pcutils_config *Show config*

Description

Show config

Usage

```
show_pcutils_config()
```

Value

config

squash	<i>Squash one column in a data.frame using other columns as id.</i>
--------	---

Description

Squash one column in a data.frame using other columns as id.

Usage

```
squash(df, column, split = ",")
```

Arguments

df	data.frame
column	column name, not numeric position
split	split string

Value

data.frame

Examples

```
df <- data.frame(a = c(1:2, 1:2), b = letters[1:4])  
squash(df, "b", ",")
```

stackplot	<i>Plot a stack plot</i>
-----------	--------------------------

Description

Plot a stack plot

Plot a area plot

Usage

```
stackplot(  
  otutab,  
  metadata = NULL,  
  group = "Group",  
  get_data = FALSE,  
  bar_params = list(width = 0.7, position = "stack"),  
  topN = 8,  
  others = TRUE,  
  relative = TRUE,
```

```

    legend_title = "",
    stack_order = TRUE,
    group_order = FALSE,
    facet_order = FALSE,
    style = c("group", "sample")[1],
    flow = FALSE,
    flow_params = list(lode.guidance = "frontback", color = "darkgray"),
    number = FALSE,
    repel = FALSE,
    format_params = list(digits = 2),
    text_params = list(position = position_stack())
)

areaplot(
  otutab,
  metadata = NULL,
  group = "Group",
  get_data = FALSE,
  bar_params = list(position = "stack"),
  topN = 8,
  others = TRUE,
  relative = TRUE,
  legend_title = "",
  stack_order = TRUE,
  group_order = FALSE,
  facet_order = FALSE,
  style = c("group", "sample")[1],
  number = FALSE,
  format_params = list(digits = 2),
  text_params = list(position = position_stack())
)

```

Arguments

otutab	otutab
metadata	metadata
group	one group name of columns of metadata
get_data	just get the formatted data?
bar_params	parameters parse to geom_bar
topN	plot how many top species
others	should plot others?
relative	transfer to relative or absolute
legend_title	fill legend_title
stack_order	the order of stack fill
group_order	the order of x group
facet_order	the order of the facet

style	"group" or "sample"
flow	should plot a flow plot?
flow_params	parameters parse to geom_flow
number	show the number?
repel	use the <code>ggrepel::geom_text_repel</code> instead of <code>geom_text</code>
format_params	parameters parse to format
text_params	parameters parse to geom_text

Value

a ggplot
a ggplot

Examples

```
data(otutab)
stackplot(otutab, metadata, group = "Group")

stackplot(otutab, metadata,
  group = "Group", style = "sample",
  group_order = TRUE, flow = TRUE, relative = FALSE
)

data(otutab)
areaplot(otutab, metadata, group = "Id")

areaplot(otutab, metadata,
  group = "Group", style = "sample",
  group_order = TRUE, relative = FALSE
)
```

strsplit2

Split Composite Names

Description

Split Composite Names

Usage

```
strsplit2(x, split, colnames = NULL, ...)
```

Arguments

x character vector
split character to split each element of vector on, see [strsplit](#)
colnames colnames for the result
... other arguments are passed to [strsplit](#)

Value

data.frame

Examples

```
strsplit2(c("a;b", "c;d"), ";")
```

t2	<i>Transpose data.frame</i>
----	-----------------------------

Description

Transpose data.frame

Usage

```
t2(data)
```

Arguments

data data.frame

Value

data.frame

taxonomy	<i>test data for pcutils package.</i>
----------	---------------------------------------

Description

an otutab, metadata and a taxonomy table.

Format

contains an otutab, metadata and a taxonomy table.

otutab contains otutable rawdata

metadata contains metadata

taxonomy contains taxonomy table

tax_pie	<i>Pie plot</i>
---------	-----------------

Description

Pie plot

Usage

```
tax_pie(otutab, topN = 6, ...)
```

Arguments

otutab	otutab
topN	topN
...	add

Value

a ggplot

Examples

```
data(otutab)
tax_pie(otutab, topN = 7)
```

tax_wordcloud	<i>Word cloud plot</i>
---------------	------------------------

Description

Word cloud plot

Usage

```
tax_wordcloud(  
  str_vector,  
  ignore_words = "Unclassified|uncultured|Ambiguous|Unknown|unknown|metagenome|Unassig",  
  topN = 50  
)
```

Arguments

str_vector	string vector
ignore_words	ignore_words
topN	topN, 50

Value

a htmlwidget

Examples

```
data(otutab)
tax_wordcloud(taxonomy$Genus)
```

tidai	<i>Replace a vector by named vector</i>
-------	---

Description

Replace a vector by named vector

Usage

```
tidai(x, y, fac = FALSE)
```

Arguments

x	a vector need to be replaced
y	named vector
fac	consider the factor?

Value

vector

Examples

```
tidai(c("a", "a", "b"), c("a" = "red", b = "blue"))
tidai(c("a", "a", "b", "c"), c("red", "blue"))
```

trans	<i>Transfer your data</i>
-------	---------------------------

Description

Transfer your data

Usage

```
trans(df, method = "normalize", margin = 2, ...)
```

Arguments

df	dataframe
method	"cpm", "minmax", "acpm", "total", "log", "max", "frequency", "normalize", "range", "rank", "rrank", "standardize", "pa", "chi.square", "hellinger", "log", "clr", "rclr", "alr"
margin	1 for row and 2 for column (default: 2)
...	additional

Value

data.frame

See Also

[decostand](#)

Examples

```
data(otutab)
trans(otutab, method = "cpm")
```

translator	<i>translator</i>
------------	-------------------

Description

translator

Usage

```
translator(words, mode = "e2z")
```

Arguments

words	words
mode	"e2z","z2e"

Value

vector

Examples

```
## Not run:
translator(c("love", "if"), mode = "e2z")

## End(Not run)
```

trans_format	<i>Transfer the format of file</i>
--------------	------------------------------------

Description

Transfer the format of file

Usage

```
trans_format(
  file,
  to_format,
  format = NULL,
  ...,
  brower = "/Applications/Microsoft Edge.app/Contents/MacOS/Microsoft Edge"
)
```

Arguments

file	input file
to_format	transfer to
format	input file format
...	additional argument
brower	the path of Google Chrome, Microsoft Edge or Chromium in your computer.

Value

file at work directory

triangp	<i>Triangle plot</i>
---------	----------------------

Description

Triangle plot

Usage

```
triangp(group_df, class = NULL)
```

Arguments

group_df	group_df
class	point color

Value

ggplot

Examples

```
data(otutab)
hebing(otutab, metadata$Group, act = "mean") -> tmp
triangp(tmp, class = taxonomy$Phylum)
```

twotest	<i>Two-group test</i>
---------	-----------------------

Description

Two-group test

Usage

```
twotest(var, group)
```

Arguments

var	numeric vector
group	two-levels group vector

Value

No return value

Examples

```
twotest(runif(20), rep(c("a", "b"), each = 10))
```

update_param	<i>Update the parameters</i>
--------------	------------------------------

Description

Keep the different parameters while use the same name in update first.

Usage

```
update_param(default, update)
```

Arguments

default	default (data.frame, list, vector)
update	update (data.frame, list, vector)

Value

same class of your input (data.frame, list or vector)

Examples

```
update_param(list(a = 1, b = 2), list(b = 5, c = 5))
```

venn	<i>Plot a general venn (upset, flower)</i>
------	--

Description

Plot a general venn (upset, flower)

Usage

```
venn(...)

## S3 method for class 'list'
venn(aa, mode = "venn", elements_label = TRUE, ...)

## S3 method for class 'data.frame'
venn(otutab, mode = "venn", elements_label = TRUE, ...)
```

Arguments

...	add
aa	list
mode	"venn","venn2","upset","flower"
elements_label	logical, show elements label in network?
otutab	table

Value

a plot
a plot
a plot

Examples

```
aa <- list(a = 1:3, b = 3:7, c = 2:4)
venn(aa, mode = "venn")
venn(aa, mode = "network")
venn(aa, mode = "upset")
data(otutab)
venn(otutab, mode = "flower")
```

write_fasta

Write a data.frame to fasta

Description

Write a data.frame to fasta

Usage

```
write_fasta(df, file_path, str_per_line = 70)
```

Arguments

df	data.frame
file_path	output file path
str_per_line	how many base or animo acid in one line, if NULL, one sequence in one line.

Value

No return value

Index

add_alpha, 4
add_analysis, 4
add_theme, 5
anova, 23
areaplot (stackplot), 49

change_fac_lev, 6
china_map, 6
chordDiagram, 35
chunlian, 7
copy_df, 8
copy_vector, 8
count2, 9

dabiao, 9
decostand, 55
del_ps, 10
df2link, 11
DNA_plot, 11
download2, 12

explode, 12

fittest, 13
format, 51

generate_labels, 13
geom_bar, 50
geom_col, 18
geom_flow, 51
geom_jitter, 22
geom_point, 35
geom_rect, 18, 19
geom_smooth, 22
geom_text, 18, 19, 21, 51
get_cols, 14
get_doi, 15
ggheatmap, 15
gghist, 17
gghistogram, 17
gghuan, 17
gghuan2, 18
ggplot_lim, 19
give_you_a_rose, 20
grepl.data.frame, 20
group_box, 21
group_test, 22
gsub.data.frame, 23
guolv, 24

hebing, 24
how_to_set_options, 25
how_to_update_parameters, 25
how_to_use_parallel, 26
how_to_use_sbatch, 26

is.ggplot.color, 27

kruskal.test, 23

legend_size, 27
lib_ps, 28
little_guodong, 28
lm_coefficients, 29

make_gitbook, 29
make_project, 30
metadata, 31
mmscale, 31
multireg, 32
multitest, 21, 32
my_cat, 33
my_circle_packing, 33
my_circo, 34
my_lm, 35
my_sunburst, 36
my_syteny, 37
my_treemap, 37
my_voronoi_treemap, 38

Olympic_rings, 38
otutab, 39

p.adjust, 23
plot.coefficients, 39
plot_ly, 36, 37
plotgif, 40
plotpdf, 40
pre_number_str, 41

read.file, 41
read_fasta, 42
reinstall_my_packages, 42
remove.outliers, 43
rgb2code, 43
rm_low, 44

sample_map, 44
sanxian, 46
search_browse, 47
set_pcutils_config, 48
show_pcutils_config, 48
squash, 49
stackplot, 49
stat_compare_means, 22
strsplit, 52
strsplit2, 51

t.test, 23
t2, 52
tax_pie, 53
tax_wordcloud, 53
taxonomy, 52
tidai, 54
trans, 55
trans_format, 56
translator, 55
triangp, 57
twotest, 57

update_param, 58

venn, 58
vt_d3, 38

wilcox.test, 23
write_fasta, 59