

Package ‘tinyarray’

August 2, 2026

Type Package

Title Expression Data Analysis and Visualization

Version 3.0.0

Maintainer Xiaojie Sun <18763899370@163.com>

Description A toolkit for microarray and RNA-seq data analysis, including annotation conversion, differential expression, enrichment, survival analysis, and visualization.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports Hmisc, dplyr, ggplot2, limma, patchwork, pheatmap, stringr, curl, httr2, survival, survminer, tibble

Suggests testthat, AnnoProbe, GEOquery, Biobase, VennDiagram, FactoMineR, factoextra, knitr, rmarkdown, cowplot, ggpubr, ggplotify, tidyr, labeling, Rtsne, scatterplot3d, ComplexHeatmap, circlize, AnnotationDbi, BiocManager, clusterProfiler, org.Rn.eg.db, org.Mm.eg.db, org.Hs.eg.db

Depends R (>= 4.3.0)

NeedsCompilation no

Config/roxygen2/version 8.0.0

Author Xiaojie Sun [aut, cre]

Repository CRAN

Date/Publication 2026-08-02 06:20:02 UTC

Contents

auto_geo_group	3
box_surv	4
cod	5
cor.full	6
cor.one	7
corheatmap	8

corscatterplot	8
deg	10
deseq_data	10
double_enrich	11
draw_boxplot	12
draw_heatmap	13
draw_heatmap2	15
draw_KM	16
draw_pca	17
draw_tsne	18
draw_venn	20
draw_volcano	21
draw_volcano2	23
dumd	24
edges_to_nodes	25
exists_anno_list	26
exprSet_hub1	26
exp_boxplot	27
exp_hub1	28
exp_surv	28
find_anno	29
genes	30
geo_download	30
geo_parser	32
get_cgs	32
get_count_txt	33
get_deg	34
get_deg_all	36
get_gpl_txt	38
get_ids	39
ggheat	40
hypertest	41
interaction_to_edges	42
intersect_all	43
lnc_anno	43
make_tcga_group	44
match_exp_cl	45
metal	46
mRNA_anno	46
multi_deg	47
multi_deg_all	48
pkg_all	50
plcortest	51
plot_deg	52
point_cut	53
print.geo_auto_group	54
quick_enrich	55
risk_plot	56

sam_filter	57
surv_cox	57
surv_KM	59
trans_array	60
trans_ensembl_exp	61
trans_entrezexp	62
trans_exp	63
trans_exp_new	64
t_choose	65
union_all	66

Index

67

auto_geo_group	auto_geo_group
----------------	----------------

Description

Infer sample groups from GEO phenotype metadata.

Usage

```
auto_geo_group(  
  x,  
  group_col = "title",  
  choice = NULL,  
  min_group_size = 2L,  
  max_groups = 20L,  
  min_coverage = 0.8,  
  search_other_cols = TRUE,  
  stopwords = .tinyarray_geo_stopwords(),  
  levels = NULL,  
  ref = NULL,  
  verbose = TRUE  
)
```

Arguments

x	A GEO result list with a ‘pd’ data frame, or a phenotype data frame.
group_col	Column name to try first. The default is ‘title’.
choice	Candidate index to select. Use ‘0’ to cancel. If ‘NULL’, the function returns the candidate table without selecting.
min_group_size	Minimum number of samples required for a split label.
max_groups	Maximum number of group labels allowed in one column.
min_coverage	Minimum fraction of samples that must be assigned a label.
search_other_cols	Whether to search other columns when ‘group_col’ and ‘title’ fail.

stopwords	Character vector of generic words to ignore during tokenization.
levels	Optional character vector giving the desired factor level order.
ref	Optional reference level to place first.
verbose	Whether to print messages and candidate tables.

Details

The function first tries the user-specified ‘group_col’, then ‘title’. If no stable grouping is found, it scans the remaining columns and returns a table of candidate groupings for manual selection. Both binary and multigroup factors are supported. Rare groups are left as ‘NA’ rather than being relabeled as another group.

Labels that differ only in capitalization are merged, retaining their most common spelling. Shared affixes and trailing sample identifiers are removed when they can be identified consistently.

Value

A list with ‘group_list’, ‘source_col’, ‘selected_index’, ‘candidate_table’, ‘candidates’, and ‘pdata’.

Examples

```
pd <- data.frame(
  title = c("Control sample", "Control sample", "Treat sample", "Treat sample"),
  stringsAsFactors = FALSE
)
rownames(pd) <- paste0("s", 1:4)
res <- auto_geo_group(
  pd,
  choice = 1,
  levels = c("Control", "Treat"),
  ref = "Control",
  verbose = FALSE
)
res$source_col
levels(res$group_list)
table(res$group_list)
```

box_surv	box_surv
----------	----------

Description

draw box plot for a hub gene expression matrix

Usage

```
box_surv(  
  exp_hub,  
  exprSet_hub,  
  meta,  
  xlab = "Group",  
  ylab = "Expression",  
  p.label = FALSE  
)
```

Arguments

- exp_hub an expression matrix for hubgenes
- exprSet_hub a tumor expression set for hubgenes
- meta meta data corresponds to expression set
- xlab x axis title for the boxplot
- ylab y axis title for the boxplot
- p.label whether to show the exact p-value, if FALSE show significance symbols

Value

patchwork result for hub genes boxplot and survival plot

Author(s)

Xiaojie Sun

See Also

[exp_boxplot](#); [exp_surv](#)

Examples

```
k = box_surv(log2(exp_hub1+1), exprSet_hub1, meta1); k[[1]]
```

cod	<i>cod</i>
-----	------------

Description

Example TCGA expression matrix used in grouping and plotting demos.

Usage

```
cod
```

Format

An object of class `matrix` (inherits from `array`) with 100 rows and 512 columns.

Examples

`cod`

<code>cor.full</code>	<i>cor.test for all variables</i>
-----------------------	-----------------------------------

Description

`cor.test` for all variables(each two columns)

Usage

`cor.full(x, drop = min(x, na.rm = TRUE) - 0.001, min.obs = 10)`

Arguments

- `x` A numeric matrix or `data.frame`
- `drop` drop values
- `min.obs` minimum number of observations after dropping

Value

a `data.frame` with `cor.test` p.value and estimate

Author(s)

Xiaojie Sun

See Also

[cor.one](#)

Examples

```
x = iris[,-5]
cor.full(x)
```

`cor.one`*cor.test for one variable with all variables*

Description

cor.test for all variables(each two columns)

Usage

```
cor.one(  
  x,  
  var,  
  drop.var = min(x[, var], na.rm = TRUE) - 0.001,  
  drop.other = min(x[, -which(colnames(x) == var)], na.rm = TRUE) - 0.001,  
  min.obs = 10  
)
```

Arguments

<code>x</code>	A numeric matrix or data.frame
<code>var</code>	your chosen variable,only one.
<code>drop.var</code>	drop values in var
<code>drop.other</code>	drop values in other columns
<code>min.obs</code>	minimum number of observations after dropping

Value

A data.frame with cor.test p.value and estimate

Author(s)

Xiaojie Sun

See Also

[cor.full](#)

Examples

```
x = iris[,-5]  
cor.one(x,"Sepal.Width")
```

corheatmap	<i>corheatmap</i>
------------	-------------------

Description

draw cor heatmap

Usage

```
corheatmap(exp, x, y, color = c("#2fa1dd", "white", "#f87669"))
```

Arguments

exp	A numeric matrix
x	genes or cells from exp
y	genes or cells from exp
color	color for heatmap

Value

a ggplot object

Author(s)

Xiaojie Sun

Examples

```
x = rownames(exprSet_hub1)[1:3]
y = rownames(exprSet_hub1)[4:7]
corheatmap(exprSet_hub1,x,y)
```

corscatterplot	<i>corscatterplot</i>
----------------	-----------------------

Description

draw cor scatter plot with density plot by ggplot2

Usage

```
corscatterplot(  
  dat,  
  x,  
  y,  
  color_cor = "blue",  
  fill_cor = "lightgray",  
  fill_x = "#ff820e",  
  fill_y = "#0000fe",  
  type = "density",  
  ...  
)
```

Arguments

dat	plot data
x	x
y	y
color_cor	color for cor reg.line
fill_cor	fill for cor reg.line
fill_x	fill for top density plot
fill_y	fill for right density plot
type	whether to use a density plot or a histogram plot for the side panel.
...	other parameters for ggscatter

Value

a ggplot object

Author(s)

Xiaojie Sun

Examples

```
corscatterplot(iris, "Sepal.Length", "Sepal.Width")
```

deg	<i>deg</i>
-----	------------

Description

Example limma differential analysis result for GSE42872.

Usage

deg

Format

An object of class `data.frame` with 18591 rows and 10 columns.

Examples

`head(deg)`

deseq_data	<i>deseq_data</i>
------------	-------------------

Description

Example DESeq2 differential analysis result.

Usage

deseq_data

Format

An object of class `data.frame` with 552 rows and 6 columns.

Examples

`head(deseq_data)`

double_enrich	<i>draw enrichment bar plots for both up and down genes</i>
---------------	---

Description

draw enrichment bar plots for both up and down genes,for human only.

Usage

```
double_enrich(deg, n = 10, color = c("#2874C5", "#f87669"), species = "human")
```

Arguments

deg	a data.frame contains at least two columns:"ENTREZID" and "change"
n	how many terms will you perform for up and down genes respectively
color	color for bar plot
species	choose human, mouse, or rat; default is human

Value

a list with kegg and go bar plot according to up and down genes enrichment result.

Author(s)

Xiaojie Sun

See Also

[quick_enrich](#)

Examples

```
## Not run:
deg = data.frame(
  ENTREZID = genes[1:20],
  change = rep(c("up", "down"), 10)
)
double_enrich(deg)

## End(Not run)
```

draw_boxplot	<i>draw boxplot for expression</i>
--------------	------------------------------------

Description

draw boxplot for expression

Usage

```
draw_boxplot(
  exp,
  group_list,
  method = "kruskal.test",
  sort = TRUE,
  drop = FALSE,
  width = 0.5,
  pvalue_cutoff = 0.05,
  xlab = "Gene",
  ylab = "Expression",
  grouplab = "Group",
  p.label = FALSE,
  add_error_bar = FALSE,
  color = c("#2874C5", "#f87669", "#e6b707", "#868686", "#66C2A5", "#FC8D62", "#8DA0CB",
    "#E78AC3", "#A6D854", "#FFD92F", "#E5C494", "#B3B3B3"),
  ...
)
```

Arguments

exp	A numeric matrix
group_list	A character or factor vector with one label per sample; character input will be converted to factor and unused levels will be dropped.
method	one of kruskal.test,aov,t.test and wilcox.test
sort	whether the boxplot will be sorted
drop	whether to discard insignificant values
width	width of boxplot and error bar
pvalue_cutoff	if drop = TRUE,genes with p-values below the threshold will be drawn
xlab	title of the x axis
ylab	title of the y axis
grouplab	title of group legend
p.label	whether to show the exact p value in the plot; if 'FALSE', show significance stars
add_error_bar	whether to add error bar
color	color vector
...	other parameters from stat_compare_means

Value

a boxplot according to exp and grouped by group.

Author(s)

Xiaojie Sun

See Also

[draw_heatmap](#); [draw_volcano](#); [draw_venn](#)

Examples

```
draw_boxplot(t(iris[,1:4]),iris$Species)
exp <- matrix(rnorm(60),nrow = 10)
colnames(exp) <- paste0("sample",1:6)
rownames(exp) <- paste0("gene",1:10)
exp[,4:6] = exp[,4:6] +10
exp[1:4,1:4]
group_list <- factor(rep(c("A","B"),each = 3))
draw_boxplot(exp,group_list)
draw_boxplot(exp,group_list,color = c("grey","red"))
```

draw_heatmap

draw a heatmap plot

Description

draw a heatmap plot for an expression matrix grouped by group_list; if scale = TRUE, the matrix is scaled before plotting.

Usage

```
draw_heatmap(
  n,
  group_list,
  scale_before = FALSE,
  n_cutoff = 3,
  legend = FALSE,
  show_rownames = FALSE,
  annotation_legend = FALSE,
  split_column = FALSE,
  show_column_title = FALSE,
  color = (grDevices::colorRampPalette(c("#2fa1dd", "white", "#f87669")))(100),
  color_an = c("#2fa1dd", "#f87669", "#e6b707", "#868686", "#92C5DE", "#F4A582",
    "#66C2A5", "#FC8D62", "#8DA0CB", "#E78AC3", "#A6D854", "#FFD92F", "#E5C494",
    "#B3B3B3"),
```

```

    scale = TRUE,
    main = NA,
    ...
  )

```

Arguments

<code>n</code>	A numeric matrix
<code>group_list</code>	A character or factor vector with one label per sample; character input will be converted to factor and unused levels will be dropped.
<code>scale_before</code>	deprecated parameter
<code>n_cutoff</code>	numeric cutoff used after scaling; values outside $[-n_cutoff, n_cutoff]$ are truncated before plotting.
<code>legend</code>	logical, show legend or not
<code>show_rownames</code>	logical, show rownames or not
<code>annotation_legend</code>	logical, show annotation legend or not
<code>split_column</code>	split column by <code>group_list</code>
<code>show_column_title</code>	show column title or not
<code>color</code>	color for heatmap
<code>color_an</code>	color for column annotation
<code>scale</code>	logical, scale the matrix or not
<code>main</code>	the title of the plot
<code>...</code>	other parameters from <code>pheatmap</code>

Value

a heatmap plot according to `n` and grouped by `group_list`.

Author(s)

Xiaojie Sun

See Also

[draw_pca](#); [draw_volcano](#); [draw_venn](#)

Examples

```

#example data
exp = matrix(abs(rnorm(60, sd = 16)), nrow = 10)
exp[, 4:6] <- exp[, 4:6] + 20
colnames(exp) <- paste0("sample", 1:6)
rownames(exp) <- paste0("gene", 1:10)
exp[1:4, 1:4]

```

```

group_list = factor(rep(c("A","B"),each = 3))
draw_heatmap(exp,group_list)
#use iris
n = t(iris[,1:4]);colnames(n) = 1:150
group_list = iris$Species
draw_heatmap(n,group_list)
draw_heatmap(n,group_list,color = colorRampPalette(c("green","black","red"))(100),
             color_an = c("red","blue","pink") )

```

draw_heatmap2

*draw heatmap plots***Description**

Print heatmap plots for an expression matrix grouped by group_list.

Usage

```
draw_heatmap2(exp, group_list, deg, my_genes = NULL, heat_union = TRUE, ...)
```

Arguments

exp	A numeric matrix
group_list	A character or factor vector with one label per sample; character input will be converted to factor and unused levels will be dropped.
deg	a data.frame created by Differential analysis
my_genes	genes for pheatmap
heat_union	logical ,use union or intersect DEGs for heatmap
...	other parameters from draw_heatmap

Value

a heatmap plot according to exp and grouped by group_list.

Author(s)

Xiaojie Sun

See Also

[draw_pca](#);[draw_volcano](#);[draw_venn](#)

Examples

```
## Not run:
if(requireNamespace("Biobase",quietly = TRUE)&
  requireNamespace("AnnoProbe",quietly = TRUE)){
  gse = "GSE474"
  geo = geo_download(gse,destdir=tempdir())
  geo$exp[1:4,1:4]
  geo$exp=log2(geo$exp+1)
  group_list=ifelse(stringr::str_detect(geo$pd$title,"MObese"),"MObese",
    ifelse(stringr::str_detect(geo$pd$title,"NonObese"),"NonObese","Obese"))
  group_list=factor(group_list,levels = c("NonObese","Obese","MObese"))
  find_anno(geo$gpl)
  ids <- AnnoProbe::idmap(geo$gpl,destdir = tempdir())
  deg = multi_deg(geo$exp,group_list,ids,adjust = FALSE,entriz = FALSE)
  draw_heatmap2(geo$exp,group_list,deg)
}else{
  if(!requireNamespace("AnnoProbe",quietly = TRUE)) {
    warning("Package 'AnnoProbe' needed for this function to work.
      Please install it by install.packages('AnnoProbe')",call. = FALSE)
  }
  if(!requireNamespace("Biobase",quietly = TRUE)) {
    warning("Package 'Biobase' needed for this function to work.
      Please install it by BiocManager::install('Biobase')",call. = FALSE)
  }
}

## End(Not run)
```

draw_KM

draw_KM

Description

draw KM-plot with two or more group

Usage

```
draw_KM(
  meta,
  group_list,
  time_col = "time",
  event_col = "event",
  legend.title = "Group",
  legend.labs = NULL,
  color = c("#2874C5", "#f87669", "#e6b707", "#868686", "#92C5DE", "#F4A582", "#66C2A5",
    "#FC8D62", "#8DA0CB", "#E78AC3", "#A6D854", "#FFD92F", "#E5C494", "#B3B3B3"),
  ...
)
```


Arguments

meta	survival data with time and event column
group_list	A character or factor vector with one label per sample; character input will be converted to factor and unused levels will be dropped.
time_col	colname of time
event_col	colname of event
legend.title	legend title
legend.labs	character vector specifying legend labels
color	color vector
...	other parameters from ggsurvplot

Value

a KM-plot

Author(s)

Xiaojie Sun

Examples

```
x = survival::lung
draw_KM(meta = x,
        group_list = x$sex,event_col = "status")
```

draw_pca	<i>draw PCA plots</i>
----------	-----------------------

Description

do PCA analysis and warning a PCA plot

Usage

```
draw_pca(
  exp,
  group_list,
  color = c("#2874C5", "#f87669", "#e6b707", "#868686", "#92C5DE", "#F4A582", "#66C2A5",
    "#FC8D62", "#8DA0CB", "#E78AC3", "#A6D854", "#FFD92F", "#E5C494", "#B3B3B3"),
  addEllipses = TRUE,
  style = "default",
  color.label = "Group",
  title = "",
  ...
)
```

Arguments

<code>exp</code>	A numeric matrix
<code>group_list</code>	A character or factor vector with one label per sample; character input will be converted to factor and unused levels will be dropped.
<code>color</code>	color vector
<code>addEllipses</code>	logical,add ellipses or not
<code>style</code>	plot style,"default","ggplot2"and "3D"
<code>color.label</code>	color legend label
<code>title</code>	plot title
<code>...</code>	other parameters from <code>fviz_pca_ind</code>

Value

a PCA plot according to `exp` and grouped by `group_list`.

Author(s)

Xiaojie Sun

See Also

[draw_heatmap](#);[draw_volcano](#);[draw_venn](#)

Examples

```
draw_pca(t(iris[,1:4]),iris$Species)
#change color
draw_pca(t(iris[,1:4]),iris$Species,color = c("#E78AC3", "#A6D854", "#FFD92F"))

draw_pca(t(iris[,1:4]),iris$Species,style = "ggplot2")

draw_pca(t(iris[,1:4]),iris$Species,style = "3D")
```

<code>draw_tsne</code>	<i>draw_tsne</i>
------------------------	------------------

Description

draw tsne plot with annotation by `ggplot2`

Usage

```
draw_tsne(
  exp,
  group_list,
  perplexity = 30,
  color = c("#2874C5", "#f87669", "#e6b707", "#868686", "#92C5DE", "#F4A582", "#66C2A5",
    "#FC8D62", "#8DA0CB", "#E78AC3", "#A6D854", "#FFD92F", "#E5C494", "#B3B3B3"),
  color.label = "group",
  addEllipses = TRUE
)
```

Arguments

exp	A numeric matrix
group_list	A character or factor vector with one label per sample; character input will be converted to factor and unused levels will be dropped.
perplexity	numeric; perplexity parameter for Rtsne
color	color vector
color.label	color legend label
addEllipses	logical, add ellipses or not

Value

a ggplot object

Author(s)

Xiaojie Sun

Examples

```
exp <- matrix(rnorm(10000), nrow = 50)
colnames(exp) <- paste0("sample", 1:200)
rownames(exp) <- paste0("gene", 1:50)
exp[1:4, 1:4]
exp[, 1:100] = exp[, 1:100] + 10
group_list <- factor(rep(c("A", "B"), each = 100))
draw_tsne(exp, group_list)
```

draw_venn

*draw a venn plot***Description**

warning a venn plot for deg result created by three packages

Usage

```
draw_venn(
  x,
  main,
  color = c("#2874C5", "#f87669", "#e6b707", "#868686", "#66C2A5", "#FC8D62", "#8DA0CB",
    "#E78AC3", "#A6D854", "#FFD92F", "#E5C494", "#B3B3B3"),
  imagetype = "png",
  filename = NULL,
  lwd = 1,
  lty = 1,
  col = color[1:length(x)],
  fill = color[1:length(x)],
  cat.col = color[1:length(x)],
  cat.cex = 1,
  cat.dist = -0.15,
  rotation.degree = 0,
  main.cex = 1,
  cex = 1,
  alpha = 0.1,
  reverse = TRUE,
  ...
)
```

Arguments

x	a list for plot
main	Character giving the main title of the diagram
color	color vector
imagetype	Specification of the image format (e.g. tiff, png or svg)
filename	Filename for image output, or if NULL returns the grid object itself
lwd	width of the circle's circumference
lty	dash pattern of the circle's circumference
col	Colour of the circle's circumference
fill	Colour of the circle's area
cat.col	Colour of the category name
cat.cex	size of the category name

cat.dist	The distance (in npc units) of the category name from the edge of the circle (can be negative)
rotation.degree	Number of degrees to rotate the entire diagram
main.cex	Number giving the cex (font size) of the main title
cex	size of the area label
alpha	Alpha transparency of the circle's area
reverse	logical,reflect the three-set Venn diagram along its central vertical axis of symmetry. Use in combination with rotation to generate all possible set orders
...	other parameters from venn.diagram

Value

a venn plot according to x, y, and z using the list names as labels.

Author(s)

Xiaojie Sun

See Also

[draw_pca](#);[draw_volcano](#);[draw_heatmap](#)

Examples

```
x = list(Deseq2=sample(1:100,30),edgeR = sample(1:100,30),limma = sample(1:100,30))
draw_venn(x,"test")
draw_venn(x,"test",color = c("darkgreen", "darkblue", "#B2182B"))
```

draw_volcano	<i>draw a volcano plot</i>
--------------	----------------------------

Description

warning a volcano plot for Differential analysis result in data.frame format.

Usage

```
draw_volcano(
  deg,
  lab = NA,
  xlab.package = TRUE,
  pvalue_cutoff = 0.05,
  logFC_cutoff = 1,
  pkg = 1,
```

```

adjust = TRUE,
change_col = "change",
symmetry = FALSE,
color = c("#2874C5", "grey", "#f87669")
)

```

Arguments

deg	a data.frame created by Differential analysis
lab	label for x axis in volcano plot; when 'NA', pkg = 4 uses 'log2FoldChange', and the other package codes keep their original labels
xlab.package	whether to use the package name as the x axis name
pvalue_cutoff	Cutoff value of pvalue, 0.05 by default.
logFC_cutoff	Cutoff value of logFC, 1 by default.
pkg	a integer, means which Differential analysis packages you used, we support three packages by now, 1, 2, 3, 4 respectively means "DESeq2", "edgeR", "limma(voom)", "limma"; pkg = 4 uses 'log2FoldChange' as the default x-axis label.
adjust	a logical value; if TRUE, use padj and plot $-\log_{10}(\text{padj})$; otherwise use P.value and plot $-\log_{10}(\text{P.value})$.
change_col	column name for the up/down/status label. Default uses 'change' when it exists; otherwise the function falls back to the existing automatic classification logic.
symmetry	a logical value, would you like to get your plot symmetrical
color	color vector

Value

a volcano plot according to logFC and either $-\log_{10}(\text{P.value})$ or $-\log_{10}(\text{padj})$, depending on adjust.

Author(s)

Xiaojie Sun

See Also

[draw_heatmap](#); [draw_pca](#); [draw_venn](#)

Examples

```

head(deseq_data)
draw_volcano(deseq_data)
draw_volcano(deseq_data, pvalue_cutoff = 0.01, logFC_cutoff = 2)
draw_volcano(deseq_data, color = c("darkgreen", "darkgrey", "#B2182B"))

```

draw_volcano2	<i>draw_volcano2</i>
---------------	----------------------

Description

Print one or more volcano plots for differential analysis results in a data.frame.

Usage

```
draw_volcano2(deg, pkg = 4, lab, change_col = "change", ...)
```

Arguments

deg	a data.frame created by Differential analysis
pkg	a integer ,means which Differential analysis packages you used,we support three packages by now, 1,2,3,4 respectively means "DESeq2","edgeR","limma(voom)","limma"; pkg = 4 uses 'log2FoldChange' as the default x-axis label.
lab	label for x axis in volcano plot; when 'NA', pkg = 4 uses 'log2FoldChange', and the other package codes keep their original labels
change_col	column name for the up/down/status label. Default uses 'change' when it exists; otherwise the function falls back to the existing automatic classification logic.
...	other parameters from draw_volcano

Value

one or more volcano plot

Author(s)

Xiaojie Sun

See Also

[geo_download](#);[draw_volcano](#);[draw_venn](#)

Examples

```
## Not run:
if(requireNamespace("Biobase",quietly = TRUE)&
  requireNamespace("AnnoProbe",quietly = TRUE)){
  #two group
  gse = "GSE42872"
  geo = geo_download(gse,destdir=tempdir())
  group_list = rep(c("A","B"),each = 3)
  ids = AnnoProbe::idmap('GPL6244',destdir = tempdir())
  deg = get_deg(geo$exp,group_list,ids)
  draw_volcano2(deg)
```

```

#multigroup
gse = "GSE474"
geo = geo_download(gse,destdir=tempdir())
geo$exp[1:4,1:4]
geo$exp=log2(geo$exp+1)
group_list=ifelse(stringr::str_detect(geo$pd$title,"MObese"),"MObese",
ifelse(stringr::str_detect(geo$pd$title,"NonObese"),"NonObese","Obese"))
group_list=factor(group_list,levels = c("NonObese","Obese","MObese"))
find_anno(geo$gpl)
ids <- AnnoProbe::idmap(geo$gpl,destdir = tempdir())
deg = multi_deg(geo$exp,group_list,ids,adjust = FALSE,entriz = FALSE)
draw_volcano2(deg)
draw_volcano2(deg,color = c("darkgreen","grey","darkred"))
}else{
  if(!requireNamespace("AnnoProbe",quietly = TRUE)) {
    warning("Package 'AnnoProbe' needed for this function to work.
Please install it by install.packages('AnnoProbe')",call. = FALSE)
  }
  if(!requireNamespace("Biobase",quietly = TRUE)) {
    warning("Package 'Biobase' needed for this function to work.
Please install it by BiocManager::install('Biobase')",call. = FALSE)
  }
}
}

## End(Not run)

```

dumd

count unique values in every columns for data.frame

Description

in geo analysis,this function can help you simplify pdata, delete columns with unique values,which can't be used as group vector

Usage

```
dumd(x)
```

Arguments

x A data.frame.

Value

The simple data.frame of columns unique values count in x

Examples

```
dumd(iris)
data(ToothGrowth)
x = ToothGrowth
dumd(ToothGrowth)
```

edges_to_nodes	<i>edges_to_nodes</i>
----------------	-----------------------

Description

get nodes from edges

Usage

```
edges_to_nodes(edges)
```

Arguments

edges data.frame

Value

nodes data.frame

Author(s)

Xiaojie Sun

See Also

[interaction_to_edges](#)

Examples

```
df = data.frame(a = c("gene1", "gene2", "gene3"),
b = c("d,f,a,b",
"c,e,g",
"a,b,d"))
edges = interaction_to_edges(df)
nodes = edges_to_nodes(edges)
```

exists_anno_list	<i>exists_anno_list</i>
------------------	-------------------------

Description

GEO GPL platforms with direct AnnoProbe annotation support.

Usage

```
exists_anno_list
```

Format

An object of class character of length 175.

Examples

```
exists_anno_list
```

exprSet_hub1	<i>exprSet_hub1</i>
--------------	---------------------

Description

Example TCGA CPM matrix with tumor samples only.

Usage

```
exprSet_hub1
```

Format

An object of class matrix (inherits from array) with 8 rows and 177 columns.

Examples

```
exprSet_hub1[1:4, 1:4]
```

exp_boxplot	<i>exp_boxplot</i>
-------------	--------------------

Description

draw box plot for a hub gene expression matrix

Usage

```
exp_boxplot(  
  exp_hub,  
  xlab = NULL,  
  ylab = NULL,  
  p.label = FALSE,  
  color = c("#2fa1dd", "#f87669")  
)
```

Arguments

exp_hub	an expression matrix for hubgenes
xlab	x axis title
ylab	y axis title
p.label	whether to show the exact p-value, if FALSE show significance symbols
color	color for boxplot

Value

box plots list for all genes in the matrix

Author(s)

Xiaojie Sun

See Also

[exp_surv](#); [box_surv](#)

Examples

```
k = exp_boxplot(log2(exp_hub+1));k[[1]]
```

exp_hub1	<i>exp_hub1</i>
----------	-----------------

Description

Example TCGA+GTEx expression matrix.

Usage

exp_hub1

Format

An object of class `matrix` (inherits from `array`) with 8 rows and 350 columns.

Examples

exp_hub1[1:4, 1:4]

exp_surv	<i>exp_surv</i>
----------	-----------------

Description

draw surv plot for a hub gene expression matrix for tumor samples

Usage

exp_surv(exprSet_hub, meta, cut.point = FALSE, color = c("#2874C5", "#f87669"))

Arguments

- | | |
|-------------|--|
| exprSet_hub | a tumor expression set for hubgenes |
| meta | meta data corresponds to expression set |
| cut.point | logical, whether to use cut_point; if FALSE, use the median as the cutoff. |
| color | color for boxplot |

Value

survival plots list for all genes

Author(s)

Xiaojie Sun

See Also[exp_boxplot](#); [box_surv](#); [draw_venn](#)**Examples**

```
tmp = exp_surv(exprSet_hub1,meta1)
patchwork::wrap_plots(tmp)+patchwork::plot_layout(guides = "collect")
tmp2 = exp_surv(exprSet_hub1,meta1,cut.point = TRUE)
patchwork::wrap_plots(tmp2)+patchwork::plot_layout(guides = "collect")
```

find_anno*Find an annotation table for a GEO platform accession.*

Description

The function prints a short explanation plus the code used to obtain annotations. It only tests AnnoProbe 'type' values when no Bioconductor annotation package source exists for the GPL accession.

Usage

```
find_anno(gpl, install = FALSE, update = FALSE)
```

Arguments

gpl	a GEO platform accession
install	whether to install and library the package
update	whether to update the package

Value

No return value. The function prints available annotation resources and optionally installs a Bioconductor annotation package.

Author(s)

Xiaojie Sun

See Also[geo_download](#)**Examples**

```
## Not run:
try(find_anno("GPL570"), silent = TRUE)

## End(Not run)
```

genes	<i>genes</i>
-------	--------------

Description

Small example vector of Entrez IDs.

Usage

```
genes
```

Format

An object of class character of length 511.

Examples

```
genes
```

geo_download	<i>geo_download</i>
--------------	---------------------

Description

Download GEO expression data and return the expression matrix, phenotype data, and platform information.

Usage

```
geo_download(  
  gse,  
  by_annoprobe = TRUE,  
  simpd = TRUE,  
  colon_remove = TRUE,  
  download_timeout = getOption("timeout"),  
  destdir = getwd(),  
  n = 1  
)
```

Arguments

<code>gse</code>	GEO accession number or a path to a cached <code>‘.rds’</code> or <code>‘.RData’</code> file containing an <code>‘ExpressionSet’</code> or a list of <code>‘ExpressionSet’</code> objects
<code>by_anno</code>	whether to try the AnnoProbe cache first and fall back to <code>GEOquery::getGEO()</code>
<code>simpld</code>	whether to simplify phenotype data by dropping columns with a single unique value
<code>colon_remove</code>	whether to drop raw <code>‘characteristics_ch1*’</code> columns while keeping the derived <code>‘:ch1’</code> phenotype column names intact
<code>download_timeout</code>	timeout in seconds used for GEOquery fallback and for mirror stall detection during the resumable download
<code>destdir</code>	The destination directory for data downloads
<code>n</code>	For data with more than one <code>ExpressionSet</code> , specify which one to analyze

Details

Cached `‘eSet’` objects are stored as `‘.rds’` files in `‘destdir’`; legacy `‘.Rdata’` cache files are still readable, and incomplete cache files will be kept so the download can resume automatically.

Value

a list with `‘exp’`, `‘pd’`, `‘gpl’`, and `‘group_candidates’`, or `‘NULL’` when the data cannot be read or downloaded

Author(s)

Xiaojie Sun

See Also

[find_anno](#)

Examples

```
## Not run:
if(requireNamespace("Biobase", quietly = TRUE)){
  gse = "GSE42872"
  a = geo_download(gse, destdir=tempdir())
}

## End(Not run)
```

`geo_parser`*geo_parser*

Description

Legacy helper for parsing GEO series matrix files. Prefer `[geo_download()]` for normal use.

Usage

```
geo_parser(gse, destdir = getwd())
```

Arguments

<code>gse</code>	GEO accession number
<code>destdir</code>	The destination directory for data downloads

Value

an ExpressionSet object list

Author(s)

Xiaojie Sun

See Also

[find_anno](#)

Examples

```
## Not run:
gse = "GSE42872"
a = try(geo_parser(gse, destdir=tempdir()), silent = TRUE)

## End(Not run)
```

`get_cgs`*get_cgs*

Description

extract DEGs from deg data.frame

Usage

```
get_cgs(deg)
```


Arguments

deg a data.frame created by Differential analysis

Value

a list with upgenes,downgenes,diffgenes.

Author(s)

Xiaojie Sun

See Also

[geo_download](#);[draw_volcano](#);[draw_venn](#)

Examples

```
## Not run:
#two group
gse = "GSE42872"
geo = geo_download(gse,destdir=tempdir())
group_list = rep(c("A","B"),each = 3)
ids = AnnoProbe::idmap('GPL6244',destdir=tempdir())
deg = get_deg(geo$exp,group_list,ids)
cgs = get_cgs(deg)
#mutigroup
gse = "GSE474"
geo = geo_download(gse,destdir=tempdir())
geo$exp[1:4,1:4]
geo$exp=log2(geo$exp+1)
group_list=ifelse(stringr::str_detect(geo$pd$title,"MObese"),"MObese",
ifelse(stringr::str_detect(geo$pd$title,"NonObese"),"NonObese","Obese"))
group_list=factor(group_list,levels = c("NonObese","Obese","MObese"))
find_anno(geo$gpl)
ids = AnnoProbe::idmap(geo$gpl,destdir = tempdir())
deg = multi_deg(geo$exp,group_list,ids,adjust = FALSE)
cgs = get_cgs(deg)

## End(Not run)
```

get_count_txt

get count from GEO

Description

get RNA-seq count file from GEO database

Usage

```
get_count_txt(gse, destdir = getwd(), download = FALSE)
```

Arguments

<code>gse</code>	GEO accession number or a path to a cached <code>‘.rds’</code> or <code>‘.RData’</code> file containing an <code>‘ExpressionSet’</code> or a list of <code>‘ExpressionSet’</code> objects
<code>destdir</code>	The destination directory for data downloads
<code>download</code>	download the txt file or not

Value

No return value. When `‘download = TRUE’`, the count file is saved in `‘destdir’`.

Author(s)

Xiaojie Sun

See Also

[geo_download](#)

Examples

```
get_count_txt("GSE162550", destdir = tempdir())
```

`get_deg`*get_deg*

Description

Do differential analysis according to an expression set and group information.

Usage

```
get_deg(  
  exp,  
  group_list,  
  ids = NULL,  
  logFC_cutoff = 1,  
  pvalue_cutoff = 0.05,  
  adjust = TRUE,  
  entriz = TRUE,  
  species = "human"  
)
```

Arguments

exp	A numeric matrix
group_list	A character or factor vector with one label per sample; character input will be converted to factor and unused levels will be dropped.
ids	optional probe-to-symbol annotation table. If 'NULL', 'exp' is treated as a gene-level matrix and no ID conversion is performed.
logFC_cutoff	Cutoff value of logFC, 1 by default.
pvalue_cutoff	Cutoff value of pvalue, 0.05 by default.
adjust	a logical value; if TRUE, use padj and plot $-\log_{10}(\text{padj})$; otherwise use P.value and plot $-\log_{10}(\text{P.value})$.
entriz	whether to convert symbols to Entrez IDs
species	choose human, mouse, or rat; default is human

Value

a deg data.frame

Author(s)

Xiaojie Sun

See Also

[multi_deg](#); [get_deg_all](#)

Examples

```
## Not run:
if(requireNamespace("Biobase", quietly = TRUE) &
  requireNamespace("AnnoProbe", quietly = TRUE)){
  gse = "GSE42872"
  a = geo_download(gse, destdir=tempdir())
  find_anno(geo$gpl)
  ids <- AnnoProbe::idmap(geo$gpl, destdir = tempdir())
  Group = rep(c("control", "treat"), each = 3)
  Group = factor(Group)
  deg = get_deg(geo$exp, Group, ids, entriz = FALSE)
  head(deg)
}else{
  if(!requireNamespace("AnnoProbe", quietly = TRUE)) {
    warning("Package 'AnnoProbe' needed for this function to work.
      Please install it by install.packages('AnnoProbe')", call. = FALSE)
  }
  if(!requireNamespace("Biobase", quietly = TRUE)) {
    warning("Package 'Biobase' needed for this function to work.
      Please install it by BiocManager::install('Biobase')", call. = FALSE)
  }
}
```

```
## End(Not run)
```

```
get_deg_all
```

```
get_deg_all
```

Description

Do differential analysis according to an expression set and group information.

Usage

```
get_deg_all(
  exp,
  group_list,
  ids = NULL,
  symmetry = TRUE,
  my_genes = NULL,
  show_rownames = FALSE,
  cluster_cols = TRUE,
  color_volcano = c("#2874C5", "grey", "#f87669"),
  logFC_cutoff = 1,
  pvalue_cutoff = 0.05,
  adjust = TRUE,
  entriz = TRUE,
  n_cutoff = 2,
  annotation_legend = FALSE,
  lab = NA,
  change_col = "change",
  species = "human"
)
```

Arguments

exp	A numeric matrix
group_list	A character or factor vector with one label per sample; character input will be converted to factor and unused levels will be dropped.
ids	optional probe-to-symbol annotation table. If 'NULL', 'exp' is treated as a gene-level matrix and no ID conversion is performed.
symmetry	a logical value ,would you like to get your plot symmetrical
my_genes	genes for pheatmap
show_rownames	logical,show rownames or not
cluster_cols	boolean values determining if columns should be clustered or hclust object.
color_volcano	color for volcano
logFC_cutoff	Cutoff value of logFC,1 by default.

pvalue_cutoff	Cutoff value of pvalue, 0.05 by default.
adjust	a logical value; if TRUE, use padj and plot $-\log_{10}(\text{padj})$; otherwise use P.value and plot $-\log_{10}(\text{P.value})$.
entriz	logical, if TRUE, convert symbols to Entrez IDs.
n_cutoff	numeric cutoff used after scaling; values outside $[-n_cutoff, n_cutoff]$ are truncated before plotting.
annotation_legend	logical, show annotation legend or not
lab	label for x axis in volcano plot; when 'NA', pkg = 4 uses 'log2FoldChange', and the other package codes keep their original labels
change_col	column name for the up/down/status label. Default uses 'change' when it exists; otherwise the function falls back to the existing automatic classification logic.
species	choose human, mouse, or rat; default is human

Value

a list with a deg data.frame, volcano plot, PCA plot, heatmap, and a list with DEGs.

Author(s)

Xiaojie Sun

See Also

[get_deg](#); [multi_deg_all](#)

Examples

```
## Not run:
if(requireNamespace("Biobase", quietly = TRUE) &
  requireNamespace("AnnoProbe", quietly = TRUE)){
  gse = "GSE42872"
  geo = geo_download(gse, destdir = tempdir())
  group_list = rep(c("A", "B"), each = 3)
  group_list = factor(group_list)
  find_anno(geo$gpl)
  ids <- AnnoProbe::idmap(geo$gpl, destdir = tempdir())
  dcp = get_deg_all(geo$exp, group_list, ids, entriz = FALSE)
  head(dcp$deg)
  dcp$plots
}else{
  if(!requireNamespace("AnnoProbe", quietly = TRUE)) {
    warning("Package 'AnnoProbe' needed for this function to work.
      Please install it by install.packages('AnnoProbe')", call. = FALSE)
  }
  if(!requireNamespace("Biobase", quietly = TRUE)) {
    warning("Package 'Biobase' needed for this function to work.
      Please install it by BiocManager::install('Biobase')", call. = FALSE)
  }
}
```

```
}  
## End(Not run)
```

get_gpl_txt	<i>get gpl txt from GEO</i>
-------------	-----------------------------

Description

get gpl annotation txt file from GEO database

Usage

```
get_gpl_txt(gpl, destdir = getwd(), download = FALSE)
```

Arguments

gpl	gpl accession from GEO database
destdir	The destination directory for data downloads
download	download the txt file or not

Value

No return value. When ‘download = TRUE’, the GPL annotation file is saved in ‘destdir’.

Author(s)

Xiaojie Sun

See Also

[geo_download](#)

Examples

```
get_gpl_txt("GPL23270", destdir = tempdir())
```

get_ids*Get probe-to-symbol ids for a GEO platform accession.*

Description

Prefer the available Bioconductor annotation package when possible, and fall back to ‘AnnoProbe::idmap()’ when no Bioconductor annotation package source exists. The function prints a short explanation plus the code it used.

Usage

```
get_ids(gpl, type = NULL, install = FALSE, update = FALSE)
```

Arguments

gpl	a GEO platform accession
type	optional AnnoProbe ‘type’ setting. If omitted, the function tries candidate types automatically.
install	whether to install and library the package
update	whether to update the package

Value

a cleaned two-column data.frame with ‘probe_id’ and ‘symbol’

Author(s)

Xiaojie Sun

See Also

[find_anno](#)

Examples

```
## Not run:  
get_ids("GPL570")  
  
## End(Not run)
```

ggheat

ggheat

Description

draw heatmap plot with annotation by ggplot2

Usage

```
ggheat(
  dat,
  group,
  cluster = FALSE,
  color = c("#2874C5", "white", "#f87669"),
  legend_color = c("#2874C5", "#f87669", "#e6b707", "#868686", "#66C2A5", "#FC8D62",
    "#8DA0CB", "#E78AC3", "#A6D854", "#FFD92F", "#E5C494", "#B3B3B3"),
  show_rownames = NULL,
  show_colnames = TRUE,
  cluster_rows = FALSE,
  cluster_cols = FALSE,
  groupname = "group",
  expname = "exp",
  fill_mid = TRUE,
  rotate_rownames = NULL
)
```

Arguments

<code>dat</code>	expression matrix for plot
<code>group</code>	group for expression colnames
<code>cluster</code>	logical, cluster in both rows and column or not, default F, now replaced by <code>cluster_rows</code> and <code>cluster_cols</code> .
<code>color</code>	color for heatmap
<code>legend_color</code>	color for legend
<code>show_rownames</code>	logical or NULL. When NULL, the function shows x-axis labels for up to 100 rows, rotates them 45 degrees for 9 to 100 rows, and hides them above 100 rows.
<code>show_colnames</code>	logical, show colnames in plot or not, default T
<code>cluster_rows</code>	logical, if rows (on the plot) should be clustered, default F
<code>cluster_cols</code>	logical, if column (on the plot) should be clustered, default F
<code>groupname</code>	name of group legend
<code>expname</code>	name of exp legend
<code>fill_mid</code>	use median value as <code>geom_tile</code> fill midpoint
<code>rotate_rownames</code>	logical or NULL. When NULL, the function uses the automatic rotation rule described in <code>show_rownames</code> .

Value

a ggplot object

Author(s)

Xiaojie Sun

Examples

```
exp_dat = matrix(sample(100:1000,40),ncol = 4)
exp_dat[1:(nrow(exp_dat)/2),] = exp_dat[1:(nrow(exp_dat)/2),]-1000
rownames(exp_dat) = paste0("sample",1:nrow(exp_dat))
colnames(exp_dat) = paste0("gene",1:ncol(exp_dat))
group = rep(c("A","B"),each = nrow(exp_dat)/2)
group = factor(group,levels = c("A","B"))
ggheat(exp_dat,group)
ggheat(exp_dat,group,cluster_rows = TRUE)
ggheat(exp_dat,group,cluster_rows = TRUE,show_rownames = FALSE,
       show_colnames = FALSE,groupname = "risk",expname = "expression")
```

hypertest	<i>hypertest</i>
-----------	------------------

Description

make hypertest for given lncRNA and mRNA common miRNAs

Usage

```
hypertest(lnc, pc, deMIR = NULL, lnctarget, pctarget)
```

Arguments

lnc	lncRNA names
pc	mRNA names
deMIR	miRNA names , default NULL
lnctarget	a data.frame with two column,lncRNA in the first column ,miRNA in the second column
pctarget	a data.frame with two column,mRNA in the first column ,miRNA in the second column

Value

a data.frame with hypertest result

Author(s)

Xiaojie Sun

See Also

[plcortest](#)

Examples

```
# to update
```

```
interaction_to_edges  interaction_to_edges
```

Description

Split interactions by ‘sep’ and return an edges data.frame.

Usage

```
interaction_to_edges(df, a = 1, b = 2, sep = ",")
```

Arguments

df	interactions data.frame
a	column to replicate
b	column to split
sep	a character string to separate b column

Value

a new data.frame with two columns, one interaction per row

Author(s)

Xiaojie Sun

See Also

[edges_to_nodes](#)

Examples

```
df = data.frame(a = c("gene1", "gene2", "gene3"),  
b = c("d,f,a,b",  
"c,e,g",  
"a,b,d"))  
interaction_to_edges(df)
```

intersect_all	<i>intersect_all</i>
---------------	----------------------

Description

calculate intersect set for two or more elements

Usage

intersect_all(...)

Arguments

... some vectors or a list with some vectors

Value

vector

Author(s)

Xiaojie Sun

See Also

[union_all](#)

Examples

```
x1 = letters[1:4]
x2 = letters[3:6]
x3 = letters[3:4]
re =intersect_all(x1,x2,x3)
re2 = intersect_all(list(x1,x2,x3))
re3 = union_all(x1,x2,x3)
```

lnc_anno	<i>lnc_anno</i>
----------	-----------------

Description

Compact lncRNA annotation table for TCGA expression matrices.

Usage

lnc_anno

Format

An object of class `data.frame` with 14852 rows and 2 columns.

Details

Only the columns needed by the package are kept, so the bundled object stays small and can be used offline without extra annotation files.

Examples

```
head(lnc_anno)
```

<code>make_tcga_group</code>	<code>make_tcga_group</code>
------------------------------	------------------------------

Description

make tcga group for given tcga sample names or expression matrix

Usage

```
make_tcga_group(exp)
```

Arguments

<code>exp</code>	sample names or an expression matrix/data.frame with sample names in its columns
------------------	--

Value

a group factor with normal and tumor, corresponding to the sample names

Author(s)

Xiaojie Sun

See Also

[sam_filter](#); [match_exp_cl](#)

Examples

```
k = make_tcga_group(exp_hub1); table(k)
k2 = make_tcga_group(colnames(exp_hub1)); table(k2)
```

match_exp_cl	<i>match_exp_cl</i>
--------------	---------------------

Description

match exp and clinical data from TCGA

Usage

```
match_exp_cl(exp, cl, id_column = "id", sample_centric = TRUE)
```

Arguments

- exp TCGA expression set
- cl TCGA clinical data.frame
- id_column which column contains patient IDs, either a column number or column name
- sample_centric logical, default TRUE; keep all samples from the same patients. If FALSE, keep only one tumor sample for each patient.

Value

a transformed clinical data.frame with sample ids.

Author(s)

Xiaojie Sun

See Also

[make_tcga_group](#); [sam_filter](#)

Examples

```
a = match_exp_cl(exp_hub1,meta1[,2:4],"X_PATIENT")
exp_matched = a[[1]]
cl_matched = a[[2]]
b = match_exp_cl(exp_hub1,meta1[,2:4],"X_PATIENT",sample_centric = FALSE)
exp_matched = b[[1]]
cl_matched = b[[2]]
```

meta1	<i>meta1</i>
-------	--------------

Description

Example clinical table matched to exprSet_hub1.

Usage

```
meta1
```

Format

An object of class `data.frame` with 177 rows and 4 columns.

Examples

```
head(meta1)
```

mRNA_anno	<i>mRNA_anno</i>
-----------	------------------

Description

Compact mRNA annotation table for TCGA expression matrices.

Usage

```
mRNA_anno
```

Format

An object of class `data.frame` with 19797 rows and 2 columns.

Details

Only the columns needed by the package are kept, so the bundled object stays small and can be used offline without extra annotation files.

Examples

```
head(mRNA_anno)
```

`multi_deg`

multi_deg

Description

Do differential analysis according to an expression set and group information.

Usage

```
multi_deg(  
  exp,  
  group_list,  
  ids = NULL,  
  logFC_cutoff = 1,  
  pvalue_cutoff = 0.05,  
  adjust = TRUE,  
  species = "human",  
  entriz = TRUE  
)
```

Arguments

<code>exp</code>	A numeric matrix
<code>group_list</code>	A character or factor vector with one label per sample; character input will be converted to factor and unused levels will be dropped.
<code>ids</code>	optional probe-to-symbol annotation table. If 'NULL', 'exp' is treated as a gene-level matrix and no ID conversion is performed.
<code>logFC_cutoff</code>	Cutoff value of logFC, 1 by default.
<code>pvalue_cutoff</code>	Cutoff value of pvalue, 0.05 by default.
<code>adjust</code>	a logical value; if TRUE, use padj and plot $-\log_{10}(\text{padj})$; otherwise use P.value and plot $-\log_{10}(\text{P.value})$.
<code>species</code>	choose human, mouse, or rat; default is human
<code>entriz</code>	whether to convert symbols to Entrez IDs

Value

a deg data.frame with one element per contrast

Author(s)

Xiaojie Sun

See Also

[get_deg](#); [multi_deg_all](#)

Examples

```
## Not run:
if(requireNamespace("Biobase",quietly = TRUE)&
  requireNamespace("AnnoProbe",quietly = TRUE)){
  gse = "GSE474"
  geo = geo_download(gse,destdir=tempdir())
  geo$exp[1:4,1:4]
  geo$exp=log2(geo$exp+1)
  group_list=ifelse(stringr::str_detect(geo$pd$title,"MObese"),
    "MObese",ifelse(stringr::str_detect(geo$pd$title,"NonObese"),
    "NonObese","Obese"))
  group_list=factor(group_list,levels = c("NonObese","Obese","MObese"))
  find_anno(geo$gpl)
  ids <- AnnoProbe::idmap(geo$gpl,destdir = tempdir())
  deg = multi_deg(geo$exp,group_list,ids,adjust = FALSE,entriz = FALSE)
  names(deg)
  head(deg[[1]])
  head(deg[[2]])
  head(deg[[3]])
}else{
  if(!requireNamespace("AnnoProbe",quietly = TRUE)) {
    warning("Package 'AnnoProbe' needed for this function to work.
      Please install it by install.packages('AnnoProbe')",call. = FALSE)
  }
  if(!requireNamespace("Biobase",quietly = TRUE)) {
    warning("Package 'Biobase' needed for this function to work.
      Please install it by BiocManager::install('Biobase')",call. = FALSE)
  }
}

## End(Not run)
```

multi_deg_all

multi_deg_all

Description

Do differential analysis according to an expression set and group information.

Usage

```
multi_deg_all(
  exp,
  group_list,
  ids = NULL,
  symmetry = TRUE,
  my_genes = NULL,
  show_rownames = FALSE,
  cluster_cols = TRUE,
```



```

color_volcano = c("#2874C5", "grey", "#f87669"),
pvalue_cutoff = 0.05,
logFC_cutoff = 1,
adjust = TRUE,
entriz = TRUE,
change_col = "change",
annotation_legend = FALSE,
lab = NA,
species = "human"
)

```

Arguments

exp	A numeric matrix
group_list	A character or factor vector with one label per sample; character input will be converted to factor and unused levels will be dropped.
ids	optional probe-to-symbol annotation table. If 'NULL', 'exp' is treated as a gene-level matrix and no ID conversion is performed.
symmetry	a logical value ,would you like to get your plot symmetrical
my_genes	genes for pheatmap
show_rownames	boolean specifying if column names are be shown.
cluster_cols	boolean values determining if columns should be clustered or hclust object.
color_volcano	color for volcano
pvalue_cutoff	Cutoff value of pvalue,0.05 by default.
logFC_cutoff	Cutoff value of logFC,1 by default.
adjust	a logical value; if TRUE, use padj and plot $-\log_{10}(\text{padj})$; otherwise use P.value and plot $-\log_{10}(\text{P.value})$.
entriz	whether to convert symbols to Entrez IDs
change_col	column name for the up/down/status label. Default uses 'change' when it exists; otherwise the function falls back to the existing automatic classification logic.
annotation_legend	boolean value showing if the legend for annotation tracks should be drawn.
lab	label for x axis in volcano plot; when 'NA', pkg = 4 uses 'log2FoldChange', and the other package codes keep their original labels
species	choose human, mouse, or rat; default is human

Value

a list with deg data.frame, volcano plot and a list with DEGs.

Author(s)

Xiaojie Sun

See Also

[geo_download](#); [draw_volcano](#); [draw_venn](#)

Examples

```
## Not run:
if(requireNamespace("Biobase",quietly = TRUE)&
  requireNamespace("AnnoProbe",quietly = TRUE)){
  gse = "GSE474"
  geo = geo_download(gse,destdir=tempdir())
  geo$exp[1:4,1:4]
  geo$exp=log2(geo$exp+1)
  group_list=ifelse(stringr::str_detect(geo$pd$title,"MObese"),"MObese",
    ifelse(stringr::str_detect(geo$pd$title,"NonObese"),"NonObese","Obese"))
  group_list=factor(group_list,levels = c("NonObese","Obese","MObese"))
  find_anno(geo$gpl)
  ids = AnnoProbe::idmap(geo$gpl,destdir = tempdir())
  dcp = multi_deg_all(geo$exp,
    group_list,ids,adjust = FALSE,entriz = FALSE)
  dcp[[3]]
}else{
  if(!requireNamespace("AnnoProbe",quietly = TRUE)) {
    warning("Package 'AnnoProbe' needed for this function to work.
      Please install it by install.packages('AnnoProbe')",call. = FALSE)
  }
  if(!requireNamespace("Biobase",quietly = TRUE)) {
    warning("Package 'Biobase' needed for this function to work.
      Please install it by BiocManager::install('Biobase')",call. = FALSE)
  }
}

## End(Not run)
```

pkg_all

pkg_all

Description

Lookup table for GEO platform annotation packages.

Usage

pkg_all

Format

An object of class data.frame with 82 rows and 3 columns.

Examples

```
head(pkg_all)
```

plcortest

plcortest

Description

make cor.test for given lncRNA and mRNA

Usage

```
plcortest(lnc_exp, mRNA_exp, cor_cutoff = 0)
```

Arguments

lnc_exp	lncRNA expression set
mRNA_exp	mRNA expression set which nrow equal to lncRNA_exp
cor_cutoff	cor estimate cut_off, default 0

Value

a list with cor.test result, names are lncRNAs, element are mRNAs

Author(s)

Xiaojie Sun

See Also

[hypertest](#)

Examples

```
# to update
```

plot_deg	<i>plot_deg</i>
----------	-----------------

Description

Plot PCA, volcano, heatmap, and Venn plots for differential analysis results.

Usage

```
plot_deg(
  exp,
  group_list,
  deg,
  symmetry = TRUE,
  my_genes = NULL,
  show_rownames = FALSE,
  cluster_cols = TRUE,
  color_volcano = c("#2874C5", "grey", "#f87669"),
  pvalue_cutoff = 0.05,
  logFC_cutoff = 1,
  adjust = TRUE,
  change_col = "change",
  annotation_legend = FALSE,
  lab = NA,
  species = "human"
)
```

Arguments

exp	A numeric matrix
group_list	A character or factor vector with one label per sample; character input will be converted to factor and unused levels will be dropped.
deg	result of multi_deg or get_deg function
symmetry	a logical value ,would you like to get your plot symmetrical
my_genes	genes for pheatmap
show_rownames	boolean specifying if column names are be shown.
cluster_cols	boolean values determining if columns should be clustered or hclust object.
color_volcano	color for volcano
pvalue_cutoff	Cutoff value of pvalue,0.05 by default.
logFC_cutoff	Cutoff value of logFC,1 by default.
adjust	a logical value; if TRUE, use padj and plot $-\log_{10}(\text{padj})$; otherwise use P.value and plot $-\log_{10}(\text{P.value})$.
change_col	column name for the up/down/status label. Default uses 'change' when it exists; otherwise the function falls back to the existing automatic classification logic.

annotation_legend	boolean value showing if the legend for annotation tracks should be drawn.
lab	label for x axis in volcano plot; when 'NA', pkg = 4 uses 'log2FoldChange', and the other package codes keep their original labels
species	choose human, mouse, or rat; default is human

Value

plots

Author(s)

Xiaojie Sun

Examples

```
## Not run:
if(requireNamespace("Biobase",quietly = TRUE)&
  requireNamespace("AnnoProbe",quietly = TRUE)){
  gse = "GSE474"
  geo = geo_download(gse,destdir=tempdir())
  geo$exp[1:4,1:4]
  geo$exp=log2(geo$exp+1)
  group_list=ifelse(stringr::str_detect(geo$pd$title,"MObese"),"MObese",
    ifelse(stringr::str_detect(geo$pd$title,"NonObese"),"NonObese","Obese"))
  group_list=factor(group_list,levels = c("NonObese","Obese","MObese"))
  find_anno(geo$gpl)
  ids = AnnoProbe::idmap(geo$gpl,destdir = tempdir())
  deg = get_deg(geo$exp,group_list,ids,adjust = FALSE,entrez = FALSE)
  plot_deg(geo$exp,group_list,deg)
}else{
  if(!requireNamespace("AnnoProbe",quietly = TRUE)) {
    warning("Package 'AnnoProbe' needed for this function to work.
      Please install it by install.packages('AnnoProbe')",call. = FALSE)
  }
  if(!requireNamespace("Biobase",quietly = TRUE)) {
    warning("Package 'Biobase' needed for this function to work.
      Please install it by BiocManager::install('Biobase')",call. = FALSE)
  }
}
## End(Not run)
```

point_cut

*point_cut***Description**

calculate cut point for multiple genes

Usage

```
point_cut(exprSet_hub, meta)
```

Arguments

<code>exprSet_hub</code>	a tumor expression set for hubgenes
<code>meta</code>	meta data corresponds to expression set

Value

a vector with cutpoint for genes

Author(s)

Xiaojie Sun

See Also

[surv_KM](#); [surv_cox](#)

Examples

```
point_cut(exprSet_hub1, meta1)
```

`print.geo_auto_group` *Print a GEO auto-group result*

Description

Print a GEO auto-group result

Usage

```
## S3 method for class 'geo_auto_group'  
print(x, ...)
```

Arguments

<code>x</code>	An object returned by ‘ <code>auto_geo_group()</code> ’.
<code>...</code>	Additional arguments passed to ‘ <code>print()</code> ’.

quick_enrich	<i>quick_enrich</i>
--------------	---------------------

Description

Perform enrichment analysis for a gene list; human is the default species.

Usage

```
quick_enrich(genes, kkggo_file = NULL, destdir = getwd(), species = "human")
```

Arguments

genes	a gene symbol or Entrez ID vector
kkggo_file	optional RDS cache filename for KEGG and GO results; when 'NULL', the function recomputes on every call and does not write a cache
destdir	destination directory used to save the cache file when 'kkggo_file' is provided
species	choose human, mouse, or rat; default is human

Value

enrichment results and dotplots

Author(s)

Xiaojie Sun

See Also

[double_enrich](#)

Examples

```
## Not run:
head(genes)
g = quick_enrich(genes, destdir = tempdir())
names(g)
g$kk
g$go

## End(Not run)
```

`risk_plot`

*risk_plot***Description**

draw risk plot

Usage

```
risk_plot(  
  exprSet_hub,  
  meta,  
  riskscore,  
  cut.point = FALSE,  
  color = c("#2fa1dd", "#f87669"),  
  n_cutoff = 3  
)
```

Arguments

<code>exprSet_hub</code>	a tumor expression set for hubgenes
<code>meta</code>	meta data corresponds to expression set
<code>riskscore</code>	a numeric vector of riskscore
<code>cut.point</code>	logical, whether to use <code>cut_point</code> ; if FALSE, use the median as the cutoff.
<code>color</code>	color for boxplot
<code>n_cutoff</code>	numeric cutoff used after scaling; values outside <code>[-n_cutoff, n_cutoff]</code> are truncated before plotting.

Value

risk plot

Author(s)

Xiaojie Sun

See Also

[exp_boxplot](#); [box_surv](#); [draw_venn](#)

Examples

```
risk_plot(exprSet_hub1, meta1, riskscore = rnorm(nrow(meta1)))
```

sam_filter	<i>sam_filter</i>
------------	-------------------

Description

drop duplicated samples from the same patients

Usage

sam_filter(exp)

Arguments

exp TCGA or TCGA+GTEx expression matrix from GDC or Xena

Value

a transformed expression set without duplicated samples

Author(s)

Xiaojie Sun

See Also

[make_tcga_group;match_exp_cl](#)

Examples

```
exp_hub1[1:4,1:4]
dim(exp_hub1)
exp2 = sam_filter(exp_hub1)
dim(exp2)
g = make_tcga_group(exp_hub1);table(g)
table(!duplicated(substr(colnames(exp_hub1[,g=="tumor"]),1,12)))
```

surv_cox	<i>surv_cox</i>
----------	-----------------

Description

calculate cox p values and HR for genes

Usage

```
surv_cox(  
  exprSet_hub,  
  meta,  
  cut.point = FALSE,  
  pvalue_cutoff = 0.05,  
  HRkeep = "all",  
  continuous = FALSE,  
  min_gn = 0.1  
)
```

Arguments

exprSet_hub	a tumor expression set for hubgenes
meta	meta data corresponds to expression set
cut.point	logical, whether to use cut_point; if FALSE, use the median as the cutoff.
pvalue_cutoff	p value cut off ,0.05 by default
HRkeep	one of "all", "protect" or "risk"
continuous	logical, gene expression or gene expression group
min_gn	Depending on the expression of a gene, there may be a large difference in sample counts between the two groups. If the smaller group contains less than 10 percent of all samples, the gene will be discarded.

Value

a matrix with gene names ,cox p value and HR

Author(s)

Xiaojie Sun

See Also

[point_cut](#); [surv_KM](#)

Examples

```
surv_cox(exprSet_hub1, meta1)  
surv_cox(exprSet_hub1, meta1, cut.point = TRUE, continuous = TRUE)  
surv_cox(exprSet_hub1, meta1, cut.point = TRUE, continuous = TRUE, pvalue_cutoff = 1)
```

surv_KM	<i>surv_KM</i>
---------	----------------

Description

calculate log_rank test p values for genes

Usage

```
surv_KM(
  exprSet_hub,
  meta,
  cut.point = FALSE,
  pvalue_cutoff = 0.05,
  min_gn = 0.1
)
```

Arguments

exprSet_hub	a tumor expression set for hubgenes
meta	meta data corresponds to expression set
cut.point	logical, whether to use cut_point; if FALSE, use the median as the cutoff.
pvalue_cutoff	p value cut off ,0.05 by default
min_gn	Depending on the expression of a gene, there may be a large difference in sample counts between the two groups. If the smaller group contains less than 10 percent of all samples, the gene will be discarded.

Value

a vector with gene names and log_rank p value

Author(s)

Xiaojie Sun

See Also

[point_cut](#); [surv_cox](#)

Examples

```
surv_KM(exprSet_hub1,meta1)
surv_KM(exprSet_hub1,meta1,pvalue_cutoff = 1)
surv_KM(exprSet_hub1,meta1,cut.point = TRUE)
```

trans_array	<i>trans_array</i>
-------------	--------------------

Description

transform rownames for microarray or rnaseq expression matrix

Usage

```
trans_array(exp, ids, from = "probe_id", to = "symbol", average = FALSE)
```

Arguments

exp	microarray or RNA-seq expression matrix
ids	data.frame with original rownames and new rownames
from	colname for original rownames
to	colname for new rownames
average	logical, if TRUE, average rows that map to the same target symbol; if FALSE, keep the first matched row for each symbol.

Value

a transformed expression set with new rownames

Author(s)

Xiaojie Sun

See Also

[trans_exp](#)

Examples

```
exp = matrix(1:50,nrow = 10)
rownames(exp) = paste0("g",1:10)
ids = data.frame(probe_id = paste0("g",1:10),
                  symbol = paste0("G",c(1:9,9)))
trans_array(exp,ids)
trans_array(exp,ids,average = TRUE)
```

trans_ensembl_exp	<i>trans_ensembl_exp</i>
-------------------	--------------------------

Description

Transform rownames of an expression set from Ensembl IDs to gene symbols using the latest gene annotation.

Usage

```
trans_ensembl_exp(  
  exp,  
  mrna_only = FALSE,  
  lncrna_only = FALSE,  
  species = "human",  
  average = FALSE  
)
```

Arguments

exp	expression matrix with Ensembl IDs as rownames
mrna_only	only keep mrna rows in result
lncrna_only	only keep lncrna rows in result
species	choose human, mouse, or rat; default is human
average	logical, if TRUE, average rows that map to the same target symbol; if FALSE, keep the first matched row for each symbol.

Value

a transformed expression set with symbol

Author(s)

Xiaojie Sun

See Also

[trans_exp_new](#)

Examples

```
exp = matrix(rnorm(1000),ncol = 10)  
rownames(exp) = sample(mRNA_anno$gene_id,100)  
colnames(exp) = c(paste0("TCGA",1:5),paste0("GTEx",1:5))  
k = trans_ensembl_exp(exp)
```

trans_entrezexp	<i>trans_entrezexp</i>
-----------------	------------------------

Description

Transform rownames of an expression set from Entrez IDs to gene symbols using `clusterProfiler::bitr()`.

Usage

```
trans_entrezexp(entrezexp, species = "human", average = FALSE)

trans_entrez_exp(entrezexp, species = "human", average = FALSE)
```

Arguments

entrezexp	expression matrix with Entrez IDs as rownames
species	choose human, mouse, or rat; default is human
average	logical, if TRUE, average rows that map to the same target symbol; if FALSE, keep the first matched row for each symbol.

Value

a transformed expression set with symbol

Author(s)

Xiaojie Sun

See Also

[trans_exp](#)

Examples

```
exp = matrix(rnorm(200),ncol = 10)
rownames(exp) = c("79691", "56271", "8662", "10394", "55630", "159162", "23541",
                  "79723", "54413", "22927", "92342", "23787", "5550", "8924",
                  "55274", "866", "8844", "353299", "587", "1473")
colnames(exp) = paste0("s",1:10)
exp2 = trans_entrezexp(exp)
```

trans_exp	<i>trans_exp</i>
-----------	------------------

Description

Transform rownames of an expression matrix from Ensembl IDs to gene symbols.

Usage

```
trans_exp(exp, mrna_only = FALSE, lncrna_only = FALSE, gtex = FALSE)
```

Arguments

exp	expression matrix with Ensembl IDs as rownames
mrna_only	only keep mrna rows in result
lncrna_only	only keep lncrna rows in result
gtex	logical, whether to include GTEx data

Value

a transformed expression set with symbol

Author(s)

Xiaojie Sun

See Also

[trans_array](#)

Examples

```
exp = matrix(rnorm(1000),ncol = 10)
rownames(exp) = sample(mRNA_anno$gene_id,100)
colnames(exp) = paste0("sample",1:10)
k = trans_exp(exp)
```

trans_exp_new	<i>trans_exp_new</i>
---------------	----------------------

Description

Route expression-matrix ID conversion by type.

Usage

```
trans_exp_new(  
  exp,  
  mrna_only = FALSE,  
  lncrna_only = FALSE,  
  species = "human",  
  average = FALSE,  
  type = c("ensembl", "array", "entrez"),  
  ids = NULL,  
  id = NULL  
)
```

Arguments

exp	expression matrix
mrna_only	only keep mrna rows in result when ‘type = "ensembl"‘
lncrna_only	only keep lncrna rows in result when ‘type = "ensembl"‘
species	choose human, mouse, or rat; default is human
average	logical, if TRUE, average rows that map to the same target symbol; if FALSE, keep the first matched row for each symbol.
type	conversion type, one of ‘ensembl‘, ‘array‘, or ‘entrez‘
ids	annotation table used when ‘type = "array"‘
id	alias of ‘ids‘; kept for compatibility

Value

a transformed expression set with symbol

Author(s)

Xiaojie Sun

See Also

[trans_ensembl_exp](#); [trans_array](#); [trans_entrezexp](#)

Examples

```
exp = matrix(rnorm(1000),ncol = 10)
rownames(exp) = sample(mRNA_anno$gene_id,100)
colnames(exp) = c(paste0("TCGA",1:5),paste0("GTEx",1:5))
k = trans_exp_new(exp)
```

t_choose

t_choose

Description

choose differential expressed genes by simple t.test

Usage

```
t_choose(
  genes,
  exp,
  group_list,
  up_only = FALSE,
  down_only = FALSE,
  pvalue_cutoff = 0.05
)
```

Arguments

genes	a vector with some genes
exp	A numeric matrix
group_list	A character or factor vector with one label per sample; character input will be converted to factor and unused levels will be dropped.
up_only	keep up genes in the result only
down_only	keep down genes in the result only
pvalue_cutoff	p value cut off ,0.05 by default

Value

a vector with differential expressed genes

Author(s)

Xiaojie Sun

Examples

```
exp = matrix(rnorm(1000),ncol = 10)
rownames(exp) = sample(mRNA_anno$gene_id,100)
colnames(exp) = c(paste0("TCGA",1:5),paste0("GTEx",1:5))
exp2 = trans_exp(exp)
exp2[,1:5] = exp2[,1:5]+10
group_list = rep(c("A","B"),each = 5)
genes = sample(rownames(exp2),3)
t_choose(genes,exp2,group_list)
```

union_all	<i>union_all</i>
-----------	------------------

Description

calculate union set for two or more elements

Usage

```
union_all(...)
```

Arguments

... some vectors or a list with some vectors

Value

vector

Author(s)

Xiaojie Sun

See Also

[intersect_all](#)

Examples

```
x1 = letters[1:4]
x2 = letters[3:6]
x3 = letters[3:4]
re =intersect_all(x1,x2,x3)
re2 = intersect_all(list(x1,x2,x3))
re3 = union_all(x1,x2,x3)
```

Index

* datasets

- cod, [5](#)
- deg, [10](#)
- deseq_data, [10](#)
- exists_anno_list, [26](#)
- exp_hub1, [28](#)
- exprSet_hub1, [26](#)
- genes, [30](#)
- lnc_anno, [43](#)
- meta1, [46](#)
- mRNA_anno, [46](#)
- pkg_all, [50](#)

auto_geo_group, [3](#)

box_surv, [4](#), [27](#), [29](#), [56](#)

cod, [5](#)
cor.full, [6](#), [7](#)
cor.one, [6](#), [7](#)
corheatmap, [8](#)
corscatterplot, [8](#)

deg, [10](#)
deseq_data, [10](#)
double_enrich, [11](#), [55](#)
draw_boxplot, [12](#)
draw_heatmap, [13](#), [13](#), [18](#), [21](#), [22](#)
draw_heatmap2, [15](#)
draw_KM, [16](#)
draw_pca, [14](#), [15](#), [17](#), [21](#), [22](#)
draw_tsne, [18](#)
draw_venn, [13–15](#), [18](#), [20](#), [22](#), [23](#), [29](#), [33](#), [50](#),
[56](#)
draw_volcano, [13–15](#), [18](#), [21](#), [21](#), [23](#), [33](#), [50](#)
draw_volcano2, [23](#)
dumd, [24](#)

edges_to_nodes, [25](#), [42](#)
exists_anno_list, [26](#)
exp_boxplot, [5](#), [27](#), [29](#), [56](#)

exp_hub1, [28](#)
exp_surv, [5](#), [27](#), [28](#)
exprSet_hub1, [26](#)

find_anno, [29](#), [31](#), [32](#), [39](#)

genes, [30](#)
geo_download, [23](#), [29](#), [30](#), [33](#), [34](#), [38](#), [50](#)
geo_parser, [32](#)
get_cgs, [32](#)
get_count_txt, [33](#)
get_deg, [34](#), [37](#), [47](#)
get_deg_all, [35](#), [36](#)
get_gpl_txt, [38](#)
get_ids, [39](#)
ggheatmap, [40](#)

hypertest, [41](#), [51](#)

interaction_to_edges, [25](#), [42](#)
intersect_all, [43](#), [66](#)

lnc_anno, [43](#)

make_tcga_group, [44](#), [45](#), [57](#)
match_exp_cl, [44](#), [45](#), [57](#)
meta1, [46](#)
mRNA_anno, [46](#)
multi_deg, [35](#), [47](#)
multi_deg_all, [37](#), [47](#), [48](#)

pkg_all, [50](#)
plcortest, [42](#), [51](#)
plot_deg, [52](#)
point_cut, [53](#), [58](#), [59](#)
print_geo_auto_group, [54](#)

quick_enrich, [11](#), [55](#)

risk_plot, [56](#)

sam_filter, [44](#), [45](#), [57](#)

surv_cox, [54](#), [57](#), [59](#)

surv_KM, [54](#), [58](#), [59](#)

t_choose, [65](#)

trans_array, [60](#), [63](#), [64](#)

trans_ensembl_exp, [61](#), [64](#)

trans_entrez_exp (trans_entrezexp), [62](#)

trans_entrezexp, [62](#), [64](#)

trans_exp, [60](#), [62](#), [63](#)

trans_exp_new, [61](#), [64](#)

union_all, [43](#), [66](#)