Package ‘tinyarray’

June 13, 2024

Type Package

Title Expression Data Analysis and Visualization

Version 2.4.2

Maintainer Xiaojie Sun <18763899370@163.com>

Description
The Gene Expression Omnibus (<https://www.ncbi.nlm.nih.gov/geo/>) and The Cancer Genome Atlas (<https://portal.gdc.cancer.gov/>) are widely used medical public databases. Our platform integrates routine analysis and visualization tools for expression data to provide concise and intuitive data analysis and presentation.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Imports BiocManager, clusterProfiler, dplyr, limma, stringr, tibble, pheatmap, ggplot2, survival, Hmisc, survminer, patchwork

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

URL https://github.com/xjsun1221/tinyarray

BugReports https://github.com/xjsun1221/tinyarray/issues

Depends R (>= 4.1.0)

RoxygenNote 7.3.1

NeedsCompilation no

Author Xiaojie Sun [aut, cre]

Repository CRAN

Date/Publication 2024-06-13 14:20:02 UTC

Contents

`box_surv` *box_surv*

Description

draw box plot for a hub gene expression matrix

Usage

```
box_surv(exp_hub, exprSet_hub, meta)
```

Arguments

`exp_hub`	an expression matrix for hubgenes
`exprSet_hub`	a tumor expression set for hubgenes
`meta`	meta data corresponds to expression set

Value

patchwork result for hub genes boxplot and survival plot

Author(s)

Xiaojie Sun

See Also

`exp_boxplot`;`exp_surv`

Examples

```
if(requireNamespace("ggpubr",quietly = TRUE)) {
k = box_surv(log2(exp_hub1+1),exprSet_hub1,meta1);k[[1]]
}else{
 warning("Package 'ggpubr' needed for this function to work.
        Please install it by install.packages('ggpubr')")
}
```

cod *cod*

Description

An expression matrix form TCGA

Usage

```
cod
```

Format

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

Examples

```
  cod
```

cor.full *cor.test for all variables*

Description

cor.test for all variables(each two columns)

Usage

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

Arguments

`x`	A numeric matrix or data.frame
`var`	your chosen variable,only one.
`drop.var`	drop values in var
`drop.other`	drop values in other columns
`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.full`

Examples

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

`corheatmap` *corheatmap*

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	*corscatterplot*

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 paramters for ggscatter

Value

a ggplot object

Author(s)

Xiaojie Sun

Examples

```
if(requireNamespace("ggpubr",quietly = TRUE)){
corscatterplot(iris,"Sepal.Length","Sepal.Width")
}else{
  warning("Package 'ggpubr' needed for this function to work.
           Please install it by install.packages('ggpubr')")
}
```

deg *deg*

Description

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 *deseq_data*

Description

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 *draw enrichment bar plots for both up and down genes*

Description

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

Usage

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

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

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:
if(requireNamespace("org.Hs.eg.db",quietly = TRUE)&
   requireNamespace("labeling",quietly = TRUE)){
   double_enrich(deg)
}else{
  if(!requireNamespace("org.Hs.eg.db",quietly = TRUE)) {
    warning("Package 'org.Hs.eg.db' needed for this function to work.
        Please install it by BiocManager::install('org.Hs.eg.db')",call. = FALSE)
  }
  if(!requireNamespace("labeling",quietly = TRUE)) {
    warning("Package 'labeling' needed for this function to work.
        Please install it by install.packages('labeling')",call. = FALSE)
  }
}

## End(Not run)
```

draw_boxplot *draw boxplot for expression*

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 factor with duplicated character or factor
`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 p value in the plot
`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

```
if(requireNamespace("tidyr",quietly = TRUE)&
   requireNamespace("ggpubr",quietly = TRUE)){
  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"))
  }else{
if(!requireNamespace("ggpubr",quietly = TRUE)) {
 warning("Package 'ggpubr' needed for this function to work.
        Please install it by install.packages('ggpubr')")
}
if(!requireNamespace("tidyr",quietly = TRUE)) {
 warning("Package 'tidyr' needed for this function to work.
        Please install it by install.packages('tidyr')")
}
}
```

draw_heatmap *draw a heatmap plot*

Description

warning a heatmap plot for expression matrix and group by group_list praramter, exp will be scaled.

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

`n`	A numeric matrix
`group_list`	A factor with duplicated character or factor
`scale_before`	deprecated parameter
`n_cutoff`	3 by defalut , scale before plot and set a cutoff,usually 2 or 1.6
`legend`	logical,show legend or not
`show_rownames`	logical,show rownames or not
`annotation_legend`	logical,show annotation legend or not
`split_column`	split column by group_list
`show_column_title`	show column title or not
`color`	color for heatmap
`color_an`	color for column annotation
`scale`	logical,scale the matrix or not
`main`	the title of the plot
`...`	other parameters from pheatmap

Value

a heatmap plot according to `exp` and grouped by `group`.

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))
if(requireNamespace("ggplotify",quietly = TRUE)){
  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") )
}else{
  warning("Package 'ggplotify' needed for this function to work.
    Please install it by install.packages('ggplotify')")
}
```

---

draw_heatmap2 *draw heatmap plots*

Description

print heatmap plots for expression matrix and group by group_list paramter

Usage

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

Arguments

`exp`	A numeric matrix
`group_list`	A factor with duplicated character or factor
`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`.

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 = levels(group_list),
  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 factor with duplicated character or factor
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

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

draw_pca *draw PCA plots*

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

`exp`	A numeric matrix
`group_list`	A factor with duplicated character or factor
`color`	color vector
`addEllipses`	logical,add ellipses or not
`style`	plot style,"default","ggplot2"and "3D"
`color.label`	color legend label
`title`	plot title
`...`	other paramters from fviz_pca_ind

Value

a pca plot according to `exp` and grouped by `group`.

Author(s)

Xiaojie Sun

See Also

`draw_heatmap`;`draw_volcano`;`draw_venn`

Examples

```
if(requireNamespace("FactoMineR",quietly = TRUE)&
   requireNamespace("factoextra",quietly = TRUE)){
  draw_pca(t(iris[,1:4]),iris$Species)
  draw_pca(t(iris[,1:4]),iris$Species,style = "ggplot2")
  #change color
  draw_pca(t(iris[,1:4]),iris$Species,color = c("#E78AC3", "#A6D854", "#FFD92F"))
}else{
  if(!requireNamespace("FactoMineR",quietly = TRUE)){
    warning("Package 'FactoMineR' needed for this function to work.
         Please install it by install.packages('FactoMineR')")
  }
  if(!requireNamespace("factoextra",quietly = TRUE)){
    warning("Package 'factoextra' needed for this function to work.
         Please install it by install.packages('factoextra')")
  }
}

if(requireNamespace("scatterplot3d",quietly = TRUE)&
   requireNamespace("FactoMineR",quietly = TRUE)){
```

```
  draw_pca(t(iris[,1:4]),iris$Species,style = "3D")
}else{
  if(!requireNamespace("scatterplot3d",quietly = TRUE)){
    warning("Package 'scatterplot3d' needed for this function to work.
         Please install it by install.packages('scatterplot3d')")
  }
  if(!requireNamespace("FactoMineR",quietly = TRUE)){
    warning("Package 'FactoMineR' needed for this function to work.
         Please install it by install.packages('FactoMineR')")
  }
}
```

`draw_tsne` *draw_tsne*

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 factor with duplicated character or factor
`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))
if(requireNamespace("Rtsne",quietly = TRUE)){
draw_tsne(exp,group_list)
}else{
 warning("Package 'Rtsne' needed for this function to work.
        Please install it by install.packages('Rtsne')")
}
```

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` named "name" paramter

Author(s)

Xiaojie Sun

See Also

`draw_pca`;`draw_volcano`;`draw_heatmap`

Examples

```
if(requireNamespace(”VennDiagram”,quietly = TRUE)&
  requireNamespace(”ggplotify”,quietly = TRUE)&
  requireNamespace(”cowplot”,quietly = TRUE)){
 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”))
}else{
 if(!requireNamespace(”VennDiagram”,quietly = TRUE)) {
   warning(”Package 'VennDiagram' needed for this function to work.
   Please install it by install.packages('VennDiagram')”)
 }
 if(!requireNamespace(”ggplotify”,quietly = TRUE)) {
   warning(”Package 'ggplotify' needed for this function to work.
   Please install it by install.packages('ggplotify')”)
 }
 if(!requireNamespace(”cowplot”,quietly = TRUE)) {
   warning(”Package 'cowplot' needed for this function to work.
   Please install it by install.packages('cowplot')”)
 }
}
```

`draw_volcano` *draw a volcano plot*

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 = FALSE,
  symmetry = FALSE,
  color = c(”#2874C5”, ”grey”, ”#f87669”)
)
```

Arguments

`deg` a data.frame created by Differential analysis

`lab` label for x axis in volcano plot, if NA , x axis names by package

`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"

`adjust` a logical value, would you like to use adjusted pvalue to draw this plot,FAlSE by default.

`symmetry` a logical value ,would you like to get your plot symmetrical

`color` color vector

Value

a volcano plot according to logFC and P.value(or adjust P.value)

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` *draw_volcano2*

Description

print one or more volcano plot for Differential analysis result in data.frame fomat.

Usage

```
draw_volcano2(deg, pkg = 4, lab, ...)
```

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"
`lab`	label for x axis in volcano plot, if NA , x axis names by package
`...`	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` *edges_to_nodes*

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` *exists_anno_list*

Description

AnnoProbe supported GPLs

Usage

```
exists_anno_list
```

Format

An object of class `character` of length 175.

Examples

```
exists_anno_list
```

`exprSet_hub1` *exprSet_hub1*

Description

An cpm expression matrix from TCGA,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 *exp_boxplot*

Description

draw box plot for a hub gene expression matrix

Usage

```
exp_boxplot(exp_hub, color = c("#2fa1dd", "#f87669"))
```

Arguments

`exp_hub`	an expression matrix for hubgenes
`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

```
if(requireNamespace("ggpubr",quietly = TRUE)) {
k = exp_boxplot(log2(exp_hub1+1));k[[1]]
}else{
 warning("Package 'ggpubr' needed for this function to work.
        Please install it by install.packages('ggpubr')")
}
```

exp_hub1 *exp_hub1*

Description

An expression matrix from TCGA and Gtex

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 *exp_surv*

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 , use cut_point or not, if FALSE,use median by defult
`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 annotation package or files*

Description

find gpl annotation package or files

Usage

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

Arguments

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

Value

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

Author(s)

Xiaojie Sun

See Also

geo_download

Examples

```
find_anno("GPL570")
```

genes *genes*

Description

some gene entriz ids

Usage

```
genes
```

Format

An object of class `character` of length 511.

Examples

```
genes
```

geo_download *geo_download*

Description

download gse data and get informations

Usage

```
geo_download(
  gse,
  by_annopbrobe = TRUE,
  simpd = TRUE,
  colon_remove = FALSE,
  destdir = getwd(),
  n = 1
)
```

Arguments

`gse`	gse assession number
`by_annopbrobe`	download data by geoquery or annoprobe
`simpd`	get simplified pdata,drop out columns with all same values
`colon_remove`	whether to remove duplicated columns with colons
`destdir`	The destination directory for data downloads.
`n`	For data with more than one ExpressionSet, specify which one to analyze

Value

a list with exp,pd and gpl

Author(s)

Xiaojie Sun

See Also

`find_anno`

Examples

```
## Not run:
if(requireNamespace("Biobase",quietly = TRUE)&
   requireNamespace("AnnoProbe",quietly = TRUE)){
  gse = "GSE42872"
  a = geo_download(gse,destdir=tempdir())
}else{
  if(!requireNamespace("AnnoProbe",quietly = TRUE)) {
    print("Package 'AnnoProbe' needed for this function to work.
         Please install it by install.packages('AnnoProbe')"print)
  }
  if(!requireNamespace("Biobase",quietly = TRUE)) {
    print("Package 'Biobase' needed for this function to work.
         Please install it by BiocManager::install('Biobase')"print)
  }
}

## 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

gse	gse assession number
destdir	The destination directory for data downloads.
download	download the txt file or not

Value

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

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 expression set and group information

Usage

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

Arguments

`exp`	A numeric matrix
`group_list`	A factor with duplicated character or factor
`ids`	a data.frame with 2 columns,including probe_id and symbol
`logFC_cutoff`	Cutoff value of logFC,1 by default.
`pvalue_cutoff`	Cutoff value of pvalue,0.05 by default.
`adjust`	a logical value, would you like to use adjusted pvalue to draw this plot,FAlSE by default.
`entriz`	whether convert symbols to entriz ids
`species`	choose human or mouse, or rat, default: human

Value

a deg data.frame with 10 columns

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 diffiencial analysis according to exprission set and group information

Usage

```
get_deg_all(
  exp,
  group_list,
  ids,
  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 = FALSE,
  entriz = TRUE,
  n_cutoff = 2,
  annotation_legend = FALSE,
  lab = NA,
  species = ”human”
)
```

Arguments

`exp`	A numeric matrix
`group_list`	A factor with duplicated character or factor
`ids`	a data.frame with 2 columns,including probe_id and symbol
`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, would you like to use adjusted pvalue to draw this plot,FAlSE by default.
`entriz`	logical , if TRUE ,convert symbol to entriz id.
`n_cutoff`	3 by defalut , scale before plot and set a cutoff,usually 2 or 1.6
`annotation_legend`	logical,show annotation legend or not
`lab`	label for x axis in volcano plot, if NA , x axis names by package
`species`	choose human or mouse, or rat, default: human

Value

a list with 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 *get gpl txt from GEO*

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

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

Author(s)

Xiaojie Sun

See Also

`geo_download`

Examples

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

---

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 = TRUE,
  show_colnames = TRUE,
  cluster_rows = FALSE,
  cluster_cols = FALSE,
  groupname = "group",
  expname = "exp",
  fill_mid = TRUE
)
```

Arguments

`dat` expression matrix for plot

`group` group for expression colnames

`cluster` logical,cluster in both rows and column or not, default F,now replaced by cluster_rows and cluster_cols.

`color`	color for heatmap
`legend_color`	color for legend
`show_rownames`	logical,show rownames in plot or not, default T
`show_colnames`	logical,show colnames in plot or not, default T
`cluster_rows`	logical, if rows (on the plot) should be clustered, default F
`cluster_cols`	logical, if column (on the plot) should be clustered, default F
`groupname`	name of group legend
`expname`	name of exp legend
`fill_mid`	use median value as geom_tile fill midpoint

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` *hypertest*

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 paramter,return 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 column ,one interaction by one rows

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 *intersect_all*

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` *lnc_anno*

Description

annotation for TCGA expression matrix(lncRNA),form genecode v22 gtf file.

Usage

```
lnc_anno
```

Format

An object of class `data.frame` with 14826 rows and 3 columns.

Examples

```
head(lnc_anno)
```

`lnc_annov23` *lnc_annov23*

Description

annotation for TCGA and gtex expression matrix(lncRNA),form genecode v23 gtf file.

Usage

```
lnc_annov23
```

Format

An object of class `data.frame` with 14852 rows and 3 columns.

Examples

```
head(lnc_annov23)
```

make_tcga_group *make_tcga_group*

Description

make tcga group for given tcga expression matrix

Usage

```
make_tcga_group(exp)
```

Arguments

`exp`	TCGA or TCGA_Gtex expression set from gdc or xena

Value

a group factor with normal and tumor ,correspond to colnames for expression matrix

Author(s)

Xiaojie Sun

See Also

`sam_filter`;`match_exp_cl`

Examples

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

match_exp_cl *match_exp_cl*

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, column number or colnmn name.

`sample_centric` logical,deault T,keep all samples from the same patients.if FALSE,keep only one tumor sample for one 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` *meta1*

Description

clinical messages for some TCGA patients,correspond to exprSet_hub1

Usage

```
meta1
```

Format

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

Examples

```
  head(meta1)
```

mRNA_anno *mRNA_anno*

Description

annotation for TCGA and gtex expression matrix(mRNA),form genecode v22 gtf file.

Usage

```
mRNA_anno
```

Format

An object of class `data.frame` with 19814 rows and 3 columns.

Examples

```
head(mRNA_anno)
```

mRNA_annov23 *mRNA_annov23*

Description

annotation for TCGA and gtex expression matrix(mRNA),form genecode v23 gtf file.

Usage

```
mRNA_annov23
```

Format

An object of class `data.frame` with 19797 rows and 3 columns.

Examples

```
head(mRNA_annov23)
```

multi_deg *multi_deg*

Description

do diffiential analysis according to expression set and group information

Usage

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

Arguments

`exp` A numeric matrix

`group_list` A factor with duplicated character or factor

`ids` a data.frame with 2 columns,including probe_id and symbol

`logFC_cutoff` Cutoff value of logFC,1 by default.

`pvalue_cutoff` Cutoff value of pvalue,0.05 by default.

`adjust` a logical value, would you like to use adjusted pvalue to draw this plot,FAlSE by default.

`species` choose human or mouse, or rat, default: human

`entriz` whether convert symbols to entriz ids

Value

a deg data.frame with 10 columns

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 diffiencial analysis according to exprission set and group information

Usage

```
multi_deg_all(
  exp,
  group_list,
  ids,
  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 = FALSE,
  entriz = TRUE,
  annotation_legend = FALSE,
  lab = NA,
  species = "human"
)
```

Arguments

`exp` A numeric matrix

`group_list` A factor with duplicated character or factor

`ids` a data.frame with 2 columns,including probe_id and symbol

`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, would you like to use adjusted pvalue to draw this plot,FAlSE by default.

`entriz` whether convert symbols to entriz ids

`annotation_legend` boolean value showing if the legend for annotation tracks should be drawn.

`lab` label for x axis in volcano plot, if NA , x axis names by package

`species` choose human or mouse, or rat, default: 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

bioconductor annotation packages for GPLs

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` *plot_deg*

Description

plot pca plot,volcano plot,heatmap,and venn plot for Differential analysis result

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 = FALSE,
  annotation_legend = FALSE,
  lab = NA,
  species = ”human”
)
```

Arguments

`exp` A numeric matrix

`group_list` A factor with duplicated character or factor

`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, would you like to use adjusted pvalue to draw this plot,FAlSE by default.

`annotation_legend`
boolean value showing if the legend for annotation tracks should be drawn.

`lab` label for x axis in volcano plot, if NA , x axis names by package

`species` choose human or mouse, or rat, default: 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,entriz = 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

exprSet_hub	a tumor expression set for hubgenes
meta	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)
```

quick_enrich *quick_enrich*

Description

do diffiencial analysis according to exprission set and group information,for human only

Usage

```
quick_enrich(
  genes,
  kkgo_file = "kkgo_file.Rdata",
  destdir = getwd(),
  species = "human"
)
```

Arguments

genes	a gene symbol or entrizid vector
kkgo_file	Rdata filename for kegg and go result
destdir	destdir to save kkgofile
species	choose human or mouse, or rat, default: human

Value

enrichment results and dotplots

Author(s)

Xiaojie Sun

See Also

double_enrich

Examples

```
## Not run:
if(requireNamespace("org.Hs.eg.db",quietly = TRUE)){
  head(genes)
  g = quick_enrich(genes,destdir = tempdir())
  names(g)
  g[[1]][1:4,1:4]
  g[[3]]
  g[[4]]
}else{
  warning("Package 'org.Hs.eg.db' needed for this function to work.
         Please install it by BiocManager::install('org.Hs.eg.db')",call. = FALSE)
  }

## 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

`exprSet_hub`	a tumor expression set for hubgenes
`meta`	meta data corresponds to expression set
`riskscore`	a numeric vector of riskscore
`cut.point`	logical , use cut_point or not, if FALSE,use median by defult
`color`	color for boxplot
`n_cutoff`	3 by defalut , scale before plot and set a cutoff,usually 2 or 1.6

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 *sam_filter*

Description

drop duplicated samples from the same patients

Usage

```
sam_filter(exp)
```

Arguments

exp	TCGA or TCGA_Gtex expression set 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

```
cod[1:4,1:4]
dim(cod)
cod2 = sam_filter(cod)
dim(cod2)
g = make_tcga_group(cod);table(g)
library(stringr)
table(!duplicated(str_sub(colnames(cod[,g=="tumor"]),1,12)))
```

surv_cox	*surv_cox*

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 , use cut_point or not, if FALSE,use median by defult
`pvalue_cutoff`	p value cut off ,0.05 by defult
`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 the number of samples between the two groups, and if a smaller group of samples is less than 10 percent (default) of all, 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 *surv_KM*

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 , use cut_point or not, if FALSE,use median by defult
`pvalue_cutoff`	p value cut off ,0.05 by defult
`min_gn`	Depending on the expression of a gene, there may be a large difference in the number of samples between the two groups, and if a smaller group of samples is less than 10 percent (default) of all, 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 *trans_array*

Description

transform rownames for microarray or rnaseq expression matrix

Usage

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

Arguments

exp	microarray expression matrix with probe_id as rownames
ids	data.frame with original rownames and new rownames
from	colname for original rownames
to	colname for new rownames

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_entrezexp *trans_entrezexp*

Description

transform rownames of expression set from "entrez" to"symbol",according to the bitr function.

Usage

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

Arguments

entrezexp	expression set with entrezid as rownames
species	choose human or mouse, or rat, default: human

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)
if(requireNamespace("org.Hs.eg.db",quietly = TRUE)){
exp2 = trans_entrezexp(exp)
}else{
    warning("Package \"org.Hs.eg.db\" needed for this function to work.
        Please install it by BiocManager::install('org.Hs.eg.db')",call. = FALSE)
}
```

trans_exp *trans_exp*

Description

transform rownames of TCGA or TCGA_Gtex expression set from gdc or xena,from ensembl id to gene symbol

Usage

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

Arguments

exp	TCGA or TCGA_Gtex expression set from gdc or xena
mrna_only	only keep mrna rows in result
lncrna_only	only keep lncrna rows in result
gtex	logical,whether including 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_annov23$gene_id,100)
colnames(exp) = c(paste0("TCGA",1:5),paste0("GTEX",1:5))
k  = trans_exp(exp)
```

trans_exp_new *trans_exp_new*

Description

transform rownames of expression set from "ensembl" to"symbol",according to the new information from ensembl database.

Usage

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

Arguments

exp	expression set with ensembl as rownames
mrna_only	only keep mrna rows in result
lncrna_only	only keep lncrna rows in result
species	choose human or mouse, or rat, default: human

Value

a transformed expression set with symbol

Author(s)

Xiaojie Sun

See Also

trans_exp

Examples

```
exp = matrix(rnorm(1000),ncol = 10)
rownames(exp) = sample(mRNA_annov23$gene_id,100)
colnames(exp) = c(paste0("TCGA",1:5),paste0("GTEX",1:5))
if(requireNamespace("AnnoProbe")){
k  = trans_exp_new(exp)
}else{
  warning("Package \"AnnoProbe\" needed for this function to work.
         Please install it by install.packages('AnnoProbe')")
}
```

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 factor with duplicated character or factor
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 defult

Value

a vector with differential expressed genes

Author(s)

Xiaojie Sun

Examples

```
exp = matrix(rnorm(1000),ncol = 10)
rownames(exp) = sample(mRNA_annov23$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 *union_all*

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