

# Package ‘enrichplot’

May 7, 2026

**Title** Visualization of Functional Enrichment Result

**Version** 1.33.0

**Description** The 'enrichplot' package provides visualization methods for interpreting functional enrichment results from ORA or GSEA analyses.

It is designed to work with the 'clusterProfiler' ecosystem and builds on 'ggplot2' for flexible and extensible graphics.

**Depends** R (>= 4.2.0)

**Imports** aplot (>= 0.2.1), DOSE, dplyr, enrichit, ggfun (>= 0.1.7), ggnewscale, ggplot2 (>= 3.5.0), ggrepel (>= 0.9.0), ggtangle (>= 0.0.9), ggtree, GOSemSim (>= 2.37.2), graphics, grid, igraph, methods, purrr, RColorBrewer, reshape2, rlang, scatterpie, stats, tidydr, utils, yulab.utils (>= 0.2.2)

**Suggests** AnnotationDbi, clusterProfiler, europepmc, ggarchery, ggforce, ggHoriPlot, ggplotify, ggribbles, ggstar, ggtreeExtra, ggupset, glue, grDevices, gridExtra, gson, org.Hs.eg.db, quarto, scales, tibble, tidyr

**VignetteBuilder** quarto

**License** Artistic-2.0

**URL** <https://yulab-smu.top/contribution-knowledge-mining/>

**BugReports** <https://github.com/GuangchuangYu/enrichplot/issues>

**biocViews** Annotation, GeneSetEnrichment, GO, KEGG, Pathways, Software, Visualization

**Encoding** UTF-8

**RoxygenNote** 7.3.3

**Roxygen** list(markdown = TRUE)

**git\_url** <https://git.bioconductor.org/packages/enrichplot>

**git\_branch** devel

**git\_last\_commit** 61afb39

**git\_last\_commit\_date** 2026-04-28

**Repository** Bioconductor 3.24

**Date/Publication** 2026-05-07

**Author** Guangchuang Yu [aut, cre] (ORCID:

<https://orcid.org/0000-0002-6485-8781>),

Chun-Hui Gao [ctb] (ORCID: <https://orcid.org/0000-0002-1445-7939>)

**Maintainer** Guangchuang Yu <guangchuangyu@gmail.com>

## Contents

enrichplot-package . . . . .	2
as.data.frame.compareClusterResult . . . . .	3
autofacet . . . . .	3
barplot.enrichResult . . . . .	4
cnetplot.enrichResult . . . . .	5
color_palette . . . . .	7
dotplot . . . . .	8
dotplot2 . . . . .	11
emapplot . . . . .	12
enrichplot-common-params . . . . .	14
enrichplot-term-params . . . . .	14
enrichplot_point_shape . . . . .	15
fortify.compareClusterResult . . . . .	15
geom_gsea_gene . . . . .	16
get_enrichplot_color . . . . .	17
ggtable . . . . .	17
goplot . . . . .	18
gseadist . . . . .	19
gseaplot . . . . .	20
gseaplot2 . . . . .	21
gsearank . . . . .	22
gsInfo . . . . .	23
heatplot . . . . .	23
hplot . . . . .	24
manhattanplot . . . . .	25
pairwise_termsim . . . . .	27
plotting.clusterProfile . . . . .	28
pmcplot . . . . .	29
reexports . . . . .	30
ridgeplot . . . . .	30
set_enrichplot_color . . . . .	31
ssplot . . . . .	32
treeplot . . . . .	34
upsetplot . . . . .	36
volplot . . . . .	37
<b>Index</b>	<b>39</b>

---

enrichplot-package      *enrichplot: Visualization of Functional Enrichment Result*

---

## Description

The 'enrichplot' package provides visualization methods for interpreting functional enrichment results from ORA or GSEA analyses. It is designed to work with the 'clusterProfiler' ecosystem and builds on 'ggplot2' for flexible and extensible graphics.

**Author(s)**

**Maintainer:** Guangchuang Yu <guangchuangyu@gmail.com> ([ORCID](#))

Other contributors:

- Chun-Hui Gao <gaospecial@gmail.com> ([ORCID](#)) [contributor]

**See Also**

Useful links:

- <https://yulab-smu.top/contribution-knowledge-mining/>
- Report bugs at <https://github.com/GuangchuangYu/enrichplot/issues>

---

as.data.frame.compareClusterResult

*Convert compareClusterResult to data frame*

---

**Description**

Convert compareClusterResult to data frame

**Usage**

```
## S3 method for class 'compareClusterResult'  
as.data.frame(x, ...)
```

**Arguments**

x	compareClusterResult object
...	additional parameters

**Value**

data frame

---

autofacet

*Plotting utility functions for enrichplot package*

---

**Description**

This file contains plotting and visualization helper functions for enrichplot. Automatically split barplot or dotplot into several facets.

**Usage**

```
autofacet(by = "row", scales = "free", levels = NULL)
```

**Arguments**

by	one of 'row' or 'column'
scales	whether 'fixed' or 'free'
levels	set facet levels

**Value**

a ggplot object

---

barplot.enrichResult *barplot*

---

**Description**

Barplot of enrichResult

**Usage**

```
## S3 method for class 'enrichResult'
barplot(
  height,
  x = "Count",
  color = "p.adjust",
  showCategory = 8,
  font.size = 12,
  title = "",
  label_format = 30,
  ...
)
```

**Arguments**

height	enrichResult object
x	one of 'Count' and 'GeneRatio'
color	one of 'pvalue', 'p.adjust' and 'qvalue'
showCategory	number of categories to display or a vector of terms.
font.size	font size
title	plot title
label_format	a numeric value sets wrap length, alternatively a custom function to format axis labels. by default wraps names longer than 30 characters
...	additional parameters

**Details**

Barplot of enrichResult

**Value**

ggplot object

**Examples**

```

library(DOSE)
data(geneList)
de <- names(geneList)[1:100]
x <- enrichDO(de)
barplot(x)
# use `showCategory` to select the displayed terms. It can be a number or a vector of terms.
barplot(x, showCategory = 10)
categories <- c("urinary bladder cancer", "bronchiolitis obliterans",
               "aortic aneurysm", "esophageal cancer")
barplot(x, showCategory = categories)

```

---

cnetplot.enrichResult *Category-Gene-Network Plot*

---

**Description**

Category-gene-network plot

**Usage**

```

## S3 method for class 'enrichResult'
cnetplot(
  x,
  layout = igraph::layout_with_kk,
  showCategory = 5,
  color_category = "#E5C494",
  size_category = 1,
  color_item = "#B3B3B3",
  size_item = 1,
  color_edge = "grey",
  size_edge = 0.5,
  categorySizeBy = ~itemNum,
  node_label = "all",
  foldChange = NULL,
  fc_threshold = NULL,
  highlight = "none",
  highlight_alpha = 0.3,
  ...
)

## S3 method for class 'gseaResult'
cnetplot(
  x,
  layout = igraph::layout_with_kk,
  showCategory = 5,
  color_category = "#E5C494",
  size_category = 1,
  color_item = "#B3B3B3",
  size_item = 1,
  color_edge = "grey",

```

```

    size_edge = 0.5,
    categorySizeBy = ~itemNum,
    node_label = "all",
    foldChange = NULL,
    fc_threshold = NULL,
    hilight = "none",
    hilight_alpha = 0.3,
    ...
)

## S3 method for class 'compareClusterResult'
cnetplot(
  x,
  layout = igraph::layout_with_kk,
  showCategory = 5,
  color_category = "#E5C494",
  size_category = 1,
  color_item = "#B3B3B3",
  size_item = 1,
  color_edge = "grey",
  size_edge = 0.5,
  categorySizeBy = ~itemNum,
  node_label = "all",
  foldChange = NULL,
  fc_threshold = NULL,
  hilight = "none",
  hilight_alpha = 0.3,
  pie = "equal",
  split = NULL,
  includeAll = TRUE,
  ...
)

```

### Arguments

x	input object
layout	network layout
showCategory	number of categories to display or a vector of terms.
color_category	color of category nodes
size_category	relative size of the category nodes
color_item	color of item nodes
size_item	relative size of the item nodes (e.g., genes)
color_edge	color of edge
size_edge	relative size of edge
categorySizeBy	An expression (e.g., <code>itemNum</code> , <code>p.adjust</code> ) or a formula (e.g., <code>~ -log10(p.adjust)</code> ) to set the category node size. For <code>compareClusterResult</code> , this controls the category pie size.
node_label	one of 'all', 'none', 'category', 'item', 'exclusive' or 'share'. 'exclusive' labels genes that uniquely belong to categories; 'share' labels genes that are shared between categories.

foldChange	numeric values to color the item (e.g., fold change of gene expression values)
fc_threshold	threshold for filtering genes by absolute fold change (e.g., fc_threshold = 1 keeps only genes with <code>lfoldChange</code> > 1).
highlight	selected categories to be highlighted
highlight_alpha	transparency value for non-highlighted items
...	additional parameters
pie	one of 'equal' or 'Count' to set the slice ratio of the pies
split	apply <code>showCategory</code> to each category specified by <code>split</code> for <code>compareClusterResult</code> , e.g. <code>ONTOLOGY</code> , <code>category</code> or <code>intersect</code> .
includeAll	logical value passed to <code>fortify()</code> when selecting terms from a <code>compareClusterResult</code> .

**See Also**[cnetplot](#)

---

color_palette	<i>Create color palette for continuous data</i>
---------------	---

---

**Description**

Create color palette for continuous data

**Usage**

```
color_palette(colors)
```

**Arguments**

colors	colors of length $\geq 2$
--------	---------------------------

**Value**

color vector

**Author(s)**

guangchuang yu

**Examples**

```
color_palette(c("red", "yellow", "green"))
```

---

`dotplot`*dotplot*

---

**Description**

Dot plot for enrichment result

**Usage**

```
dotplot(object, ...)
```

```
## S4 method for signature 'enrichResult'
```

```
dotplot(  
  object,  
  x = "GeneRatio",  
  color = "p.adjust",  
  showCategory = 10,  
  size = NULL,  
  split = NULL,  
  font.size = 12,  
  title = "",  
  orderBy = "x",  
  label_format = 30,  
  ...  
)
```

```
## S4 method for signature 'gseaResult'
```

```
dotplot(  
  object,  
  x = "GeneRatio",  
  color = "p.adjust",  
  showCategory = 10,  
  size = NULL,  
  split = NULL,  
  font.size = 12,  
  title = "",  
  orderBy = "x",  
  label_format = 30,  
  ...  
)
```

```
## S4 method for signature 'compareClusterResult'
```

```
dotplot(  
  object,  
  x = "Cluster",  
  color = "p.adjust",  
  showCategory = 5,  
  split = NULL,  
  font.size = 12,  
  title = "",  
  by = "geneRatio",
```

```
    size = NULL,
    includeAll = TRUE,
    label_format = 30,
    ...
)

## S4 method for signature 'enrichResultList'
dotplot(
  object,
  x = "GeneRatio",
  color = "p.adjust",
  showCategory = 10,
  size = NULL,
  split = NULL,
  font.size = 12,
  title = "",
  orderBy = "x",
  label_format = 30,
  ...
)

## S4 method for signature 'gseaResultList'
dotplot(
  object,
  x = "GeneRatio",
  color = "p.adjust",
  showCategory = 10,
  size = NULL,
  split = NULL,
  font.size = 12,
  title = "",
  orderBy = "x",
  label_format = 30,
  ...
)

dotplot.enrichResult(
  object,
  x = "geneRatio",
  color = "p.adjust",
  showCategory = 10,
  size = NULL,
  split = NULL,
  font.size = 12,
  title = "",
  orderBy = "x",
  label_format = 30,
  decreasing = TRUE
)

dotplot.compareClusterResult(
  object,
```

```

x = "Cluster",
colorBy = "p.adjust",
showCategory = 5,
by = "geneRatio",
size = "geneRatio",
split = NULL,
includeAll = TRUE,
font.size = 12,
title = "",
label_format = 30,
group = FALSE,
shape = FALSE,
facet = NULL,
strip_width = 15
)

```

### Arguments

object	compareClusterResult object
...	additional parameters.
x	variable for x-axis, one of 'GeneRatio' and 'Count'
color	variable used to color enriched terms, e.g. 'pvalue', 'p.adjust' or 'qvalue'
showCategory	number of categories to display or a vector of terms.
size	variable used to scale the sizes of categories, one of "geneRatio", "Percentage" and "count"
split	apply showCategory to each category specified by the 'split', e.g., "ONTOLOGY", "category" and "intersect". Default is NULL and do nothing
font.size	font size
title	figure title
orderBy	The order of the Y-axis
label_format	a numeric value sets wrap length, alternatively a custom function to format axis labels. by default wraps names longer than 30 characters
by	one of "geneRatio", "Percentage" and "count"
includeAll	logical value
decreasing	logical. Should the orderBy order be increasing or decreasing?
colorBy	variable used to color enriched terms, e.g. 'pvalue', 'p.adjust' or 'qvalue'
group	a logical value, whether to connect the nodes of the same group with wires.
shape	a logical value, whether to use nodes of different shapes to distinguish the group it belongs to
facet	apply facet_grid to the plot by specified variable, e.g., "ONTOLOGY", "category" and "intersect".
strip_width	width of strip text (facet label).

### Value

plot.

**Author(s)**

Guangchuang Yu

**Examples**

```
## Not run:
  library(DOSE)
  data(geneList)
  de <- names(geneList)[1:100]
  x <- enrichDO(de)
  dotplot(x)
# use `showCategory` to select the displayed terms. It can be a number of a vector of terms.
dotplot(x, showCategory = 10)
categories <- c("pre-malignant neoplasm", "intestinal disease",
               "breast ductal carcinoma", "non-small cell lung carcinoma")
dotplot(x, showCategory = categories)
# It can also graph compareClusterResult
data(gcSample)
library(clusterProfiler)
library(DOSE)
library(org.Hs.eg.db)
data(gcSample)
xx <- compareCluster(gcSample, fun="enrichGO", OrgDb="org.Hs.eg.db")
xx2 <- pairwise_termsim(xx)
library(ggstar)
dotplot(xx2)
dotplot(xx2, shape = TRUE)
dotplot(xx2, group = TRUE)
dotplot(xx2, x = "GeneRatio", group = TRUE, size = "count")

## End(Not run)
```

---

dotplot2

*dotplot2*

---

**Description**

compare two clusters in the compareClusterResult object

**Usage**

```
dotplot2(object, x = "FoldEnrichment", vars = NULL, label = "auto", ...)
```

**Arguments**

object	a compareClusterResult object
x	selected variable to visualize in x-axis
vars	selected Clusters to be compared, only length of two is supported
label	to label the Clusters in the plot, should be a named vector
...	additional parameters passed to dotplot

**Value**

a ggplot object

**Author(s)**

Guangchuang Yu

---

emapplot

*emapplot*

---

**Description**

Enrichment Map for enrichment result of over-representation test or gene set enrichment analysis

**Usage**

```
emapplot(x, ...)

## S4 method for signature 'enrichResult'
emapplot(x, showCategory = 30, ...)

## S4 method for signature 'gseaResult'
emapplot(x, showCategory = 30, ...)

## S4 method for signature 'compareClusterResult'
emapplot(x, showCategory = 30, ...)

emapplot_internal(
  x,
  layout = igraph::layout_with_kk,
  showCategory = 30,
  color = "p.adjust",
  size_category = 1,
  min_edge = 0.2,
  color_edge = "grey",
  size_edge = 0.5,
  node_label = "category",
  node_label_size = 5,
  pie = "equal",
  label_format = 30,
  clusterFunction = stats::kmeans,
  nWords = 4,
  nCluster = NULL
)
```

**Arguments**

x	Enrichment result.
...	Additional parameters
showCategory	number of categories to display or a vector of terms.

layout	igraph layout
color	Variable used to color enriched terms, e.g. 'pvalue', 'p.adjust' or 'qvalue'.
size_category	relative size of the categories
min_edge	The minimum similarity threshold for whether two nodes are connected, should be between 0 and 1, default value is 0.2.
color_edge	color of the network edge
size_edge	relative size of edge width
node_label	Select which labels to display, one of 'category', 'group', 'all' and 'none'.
node_label_size	size of node label, default is 5.
pie	one of 'equal' or 'Count' to set the slice ratio of the pies
label_format	a numeric value sets wrap length, alternatively a custom function to format axis labels.
clusterFunction	clustering method function, such as <code>stats::kmeans</code> (default), <code>cluster::clara</code> , <code>cluster::fanny</code> , or <code>cluster::pam</code> .
nWords	Numeric, the number of words in the cluster tags, the default value is 4.
nCluster	Numeric, the number of clusters, the default value is square root of the number of nodes.

## Details

This function visualizes gene sets as a network (i.e. enrichment map). Mutually overlapping gene sets tend to cluster together, making it easier for interpretation. When the similarity between terms meets a certain threshold (default is 0.2, adjusted by parameter `min_edge`), there will be edges between terms. The stronger the similarity, the shorter and thicker the edges. The similarity between terms is obtained by the function `pairwise_termsim`. Details of the similarity calculation can be found in its documentation: [pairwise\\_termsim\(\)](#).

## Value

ggplot object

## Author(s)

Guangchuang Yu

## Examples

```
## Not run:
library(DOSE)
data(geneList)
de <- names(geneList)[1:100]
x <- enrichDO(de)
x2 <- pairwise_termsim(x)
emapplot(x2)
# use `layout` to change the layout of map
emapplot(x2, layout = "star")
# use `showCategory` to select the displayed terms. It can be a number of a vector of terms.
emapplot(x2, showCategory = 10)
categories <- c("pre-malignant neoplasm", "intestinal disease",
```

```

                                "breast ductal carcinoma")
emapplot(x2, showCategory = categories)

# It can also graph compareClusterResult
library(clusterProfiler)
library(DOSE)
library(org.Hs.eg.db)
data(gcSample)
xx <- compareCluster(gcSample, fun="enrichGO", OrgDb="org.Hs.eg.db")
xx2 <- pairwise_termsim(xx)
emapplot(xx2)

## End(Not run)

```

---

enrichplot-common-params

*Shared parameters for enrichment plots*

---

### Description

Shared parameters for enrichment plots

### Arguments

color	variable used to color enriched terms, e.g. 'pvalue', 'p.adjust', or 'qvalue'.
showCategory	number of categories to display, or a vector of terms.
size	variable used to scale category size, one of "geneRatio", "Percentage", or "count".
split	apply showCategory to each category specified by split, e.g., "ONTOLOGY", "category", or "intersect". Default is NULL.
font.size	font size.
title	figure title.
label_format	a numeric wrap width, or a custom function to format axis labels.
includeAll	logical value.

---

enrichplot-term-params

*Shared term-plot parameters*

---

### Description

Shared term-plot parameters

### Arguments

showCategory	number of categories to display, or a vector of terms.
color	variable used to color enriched terms, e.g. 'pvalue', 'p.adjust', or 'qvalue'.
label_format	a numeric wrap width, or a custom function to format axis labels.

---

`enrichplot_point_shape`*Predefined color palettes*

---

**Description**

Predefined color palettes

**Usage**`enrichplot_point_shape`**Format**

An object of class `numeric` of length 1.

---

`fortify.compareClusterResult`*fortify*

---

**Description**

convert `compareClusterResult` to a `data.frame` that ready for plot  
convert `enrichResult` object for `ggplot2`

**Usage**

```
## S3 method for class 'compareClusterResult'  
fortify(  
  model,  
  data,  
  showCategory = 5,  
  by = "geneRatio",  
  split = NULL,  
  includeAll = TRUE,  
  ...  
)
```

```
## S3 method for class 'enrichResult'  
fortify(  
  model,  
  data,  
  showCategory = 5,  
  by = "Count",  
  order = FALSE,  
  drop = FALSE,  
  split = NULL,  
  ...  
)
```

**Arguments**

model	'enrichResult' or 'compareClusterResult' object
data	not use here
showCategory	Category numbers to show
by	one of Count and GeneRatio
split	separate result by 'split' variable
includeAll	logical
...	additional parameter
order	logical
drop	logical

**Value**

data.frame  
data.frame

**Author(s)**

Guangchuang Yu

---

geom_gsea_gene	<i>geom_gsea_gene</i>
----------------	-----------------------

---

**Description**

label genes in running score plot

**Usage**

```
geom_gsea_gene(  
  genes,  
  mapping = NULL,  
  geom = ggplot2::geom_text,  
  ...,  
  geneSet = NULL  
)
```

**Arguments**

genes	selected genes to be labeled
mapping	aesthetic mapping, default is NULL
geom	geometric layer to plot the gene labels, default is geom_text
...	additional parameters passed to the 'geom'
geneSet	choose which gene set(s) to be label if the plot contains multiple gene sets

**Value**

ggplot object

**Author(s)**

Guangchuang Yu

---

*get\_enrichplot\_color*    *Color utility functions for enrichplot package*

---

**Description**

This file contains all color-related helper functions Get default enrichplot colors

**Usage**

```
get_enrichplot_color(n = 2)
```

**Arguments**

n                    number of colors (2 or 3)

**Value**

color vector

---

*ggtable*                    *ggtable*

---

**Description**

plot table

**Usage**

```
ggtable(d, p = NULL)
```

**Arguments**

d                    data frame

p                    ggplot object to extract color to color rownames(d), optional

**Value**

ggplot object

**Author(s)**

guangchuang yu

---

`goplot`*goplot*

---

## Description

Plot induced GO DAG of significant terms

## Usage

```
goplot(  
  x,  
  showCategory = 10,  
  color = "p.adjust",  
  layout = "sugiyama",  
  geom = "text",  
  ...  
)  
  
## S4 method for signature 'enrichResult'  
goplot(  
  x,  
  showCategory = 10,  
  color = "p.adjust",  
  layout = igraph::layout_with_sugiyama,  
  geom = "text",  
  ...  
)  
  
## S4 method for signature 'gseaResult'  
goplot(  
  x,  
  showCategory = 10,  
  color = "p.adjust",  
  layout = igraph::layout_with_sugiyama,  
  geom = "text",  
  ...  
)
```

## Arguments

<code>x</code>	enrichment result.
<code>showCategory</code>	number of categories to display, or a vector of terms.
<code>color</code>	variable used to color enriched terms, e.g. 'pvalue', 'p.adjust', or 'qvalue'.
<code>layout</code>	layout of the map
<code>geom</code>	label geom, one of 'label' or 'text'
<code>...</code>	additional parameters.

## Value

ggplot object

**Author(s)**

Guangchuang Yu

**Examples**

```
## Not run:
library(clusterProfiler)
data(geneList, package = "DOSE")
de <- names(geneList)[1:100]
yy <- enrichGO(de, 'org.Hs.eg.db', ont="BP", pvalueCutoff=0.01)
goplot(yy)
goplot(yy, showCategory = 5)

## End(Not run)
```

---

gseadist

*gseadist*

---

**Description**

plot logFC distribution of selected gene sets

**Usage**

```
gseadist(x, IDs, type = "density")
```

**Arguments**

x	GSEA result
IDs	gene set IDs
type	one of 'density' or 'boxplot'

**Value**

distribution plot

**Author(s)**

Guangchuang Yu

---

`gseaplot`*gseaplot*

---

**Description**

Visualize GSEA analysis results

**Usage**

```
gseaplot(x, geneSetID, by = "all", title = "", ...)
```

```
## S4 method for signature 'gseaResult'
```

```
gseaplot(  
  x,  
  geneSetID,  
  by = "all",  
  title = "",  
  color = "black",  
  color.line = "green",  
  color.vline = "#FA5860",  
  ...  
)
```

```
gseaplot.gseaResult(  
  x,  
  geneSetID,  
  by = "all",  
  title = "",  
  color = "black",  
  color.line = "green",  
  color.vline = "#FA5860",  
  ...  
)
```

**Arguments**

<code>x</code>	<code>gseaResult</code> object
<code>geneSetID</code>	geneSet ID
<code>by</code>	one of "runningScore" or "position"
<code>title</code>	plot title
<code>...</code>	additional parameters
<code>color</code>	color of line segments
<code>color.line</code>	color of running enrichment score line
<code>color.vline</code>	color of vertical line indicating the maximum/minimal running enrichment score

**Details**

Plotting function for `gseaResult`

**Value**

ggplot2 object  
ggplot2 object

**Author(s)**

Guangchuang Yu

**Examples**

```
library(DOSE)
data(geneList)
x <- gseDO(geneList)
gseaplot(x, geneSetID=1)
```

---

gseaplot2

*gseaplot2*


---

**Description**

GSEA plot that mimic the plot generated by broad institute's GSEA software

**Usage**

```
gseaplot2(
  x,
  geneSetID,
  title = "",
  color = "green",
  base_size = 11,
  rel_heights = c(1.5, 0.5, 1),
  subplots = 1:3,
  pvalue_table = FALSE,
  pvalue_table_columns = c("pvalue", "p.adjust"),
  pvalue_table_rownames = "Description",
  ES_geom = "line"
)
```

**Arguments**

x	gseaResult object
geneSetID	gene set ID
title	plot title
color	color of running enrichment score line
base_size	base font size
rel_heights	relative heights of subplots
subplots	which subplots to be displayed
pvalue_table	whether add pvalue table

pvalue_table_columns	selected columns to be plotted in the pvalue_table
pvalue_table_rownames	selected column as the rownames of the pvalue_table. If set to NULL, no rownames will be displayed.
ES_geom	geom for plotting running enrichment score, one of 'line' or 'dot'

**Value**

plot

**Author(s)**

Guangchuang Yu

---

gsearank

*gsearank*

---

**Description**

plot ranked list of genes with running enrichment score as bar height

**Usage**

```
gsearank(x, geneSetID, title = "", output = "plot")
```

**Arguments**

x	gseaResult object
geneSetID	gene set ID
title	plot title
output	one of 'plot' or 'table' (for exporting data)

**Value**

ggplot object

**Author(s)**

Guangchuang Yu

---

 gsInfo

*gsInfo*


---

**Description**

extract gsea result of selected geneSet

**Usage**

```
gsInfo(object, geneSetID)
```

**Arguments**

object	gseaResult object
geneSetID	gene set ID

**Value**

data.frame

**Author(s)**

Guangchuang Yu

---

heatplot

*heatplot*


---

**Description**

Heatmap-like plot for functional classification

**Usage**

```
heatplot(x, showCategory = 30, ...)

## S4 method for signature 'enrichResult'
heatplot(x, showCategory = 30, ...)

## S4 method for signature 'gseaResult'
heatplot(x, showCategory = 30, ...)

heatplot.enrichResult(
  x,
  showCategory = 30,
  showTop = NULL,
  symbol = "rect",
  foldChange = NULL,
  pvalue = NULL,
  label_format = 30
)
```

**Arguments**

x	enrichment result.
showCategory	number of enriched terms to display
...	Additional parameters
showTop	number of top genes ranked by $\text{abs}(\text{foldChange}) * \text{frequency}$ to be shown in the heatmap, default NULL means all genes are shown
symbol	symbol of the nodes, one of "rect" (the default) or "dot"
foldChange	fold change.
pvalue	pvalue of genes
label_format	a numeric value sets wrap length, alternatively a custom function to format axis labels. by default wraps names longer than 30 characters

**Value**

ggplot object

**Author(s)**

Guangchuang Yu

**Examples**

```
library(DOSE)
data(geneList)
de <- names(geneList)[1:100]
x <- enrichDO(de)
heatplot(x)
```

---

hplot

*hplot*

---

**Description**

Horizontal plot for GSEA result

**Usage**

```
hplot(x, geneSetID)
```

**Arguments**

x	gseaResult object
geneSetID	gene set ID

**Value**

horizontal plot

**Author(s)**

Guangchuang Yu

---

manhattanplot	<i>manhattanplot</i>
---------------	----------------------

---

## Description

Manhattan plot for enrichment result

## Usage

```
manhattanplot(x, ...)
```

```
## S4 method for signature 'enrichResult'
```

```
manhattanplot(  
  x,  
  color = "p.adjust",  
  showCategory = 5,  
  size = "Count",  
  split = NULL,  
  font.size = 12,  
  title = "",  
  label_format = 30,  
  ...  
)
```

```
## S4 method for signature 'gseaResult'
```

```
manhattanplot(  
  x,  
  color = "p.adjust",  
  showCategory = 5,  
  size = "Count",  
  split = NULL,  
  font.size = 12,  
  title = "",  
  label_format = 30,  
  ...  
)
```

```
## S4 method for signature 'compareClusterResult'
```

```
manhattanplot(  
  x,  
  color = "p.adjust",  
  showCategory = 5,  
  split = NULL,  
  font.size = 12,  
  title = "",  
  size = "Count",  
  includeAll = TRUE,  
  label_format = 30,  
  ...  
)
```

```

## S4 method for signature 'enrichResultList'
manhattanplot(
  x,
  color = "p.adjust",
  showCategory = 5,
  size = "Count",
  split = NULL,
  font.size = 12,
  title = "",
  label_format = 30,
  ...
)

## S4 method for signature 'gseaResultList'
manhattanplot(
  x,
  color = "p.adjust",
  showCategory = 5,
  size = "Count",
  split = NULL,
  font.size = 12,
  title = "",
  label_format = 30,
  ...
)

## S4 method for signature 'list'
manhattanplot(
  x,
  color = "p.adjust",
  showCategory = 5,
  size = "Count",
  split = NULL,
  font.size = 12,
  title = "",
  label_format = 30,
  ...
)

```

### Arguments

x	enrichment result.
...	additional parameters.
color	variable used to color enriched terms, e.g. 'pvalue', 'p.adjust', or 'qvalue'.
showCategory	number of categories to display, or a vector of terms.
size	variable used to scale category size, one of "geneRatio", "Percentage", or "count".
split	apply showCategory to each category specified by split, e.g., "ONTOLOGY", "category", or "intersect". Default is NULL.
font.size	font size.
title	figure title.

label\_format    a numeric wrap width, or a custom function to format axis labels.  
 includeAll     logical value.

**Value**

ggplot object

**Author(s)**

Guangchuang Yu

---

pairwise\_termsim     *pairwise\_termsim*

---

**Description**

Get the similarity matrix

**Usage**

```
pairwise_termsim(x, method = "JC", semData = NULL, showCategory = NULL)

## S4 method for signature 'enrichResult'
pairwise_termsim(x, method = "JC", semData = NULL, showCategory = NULL)

## S4 method for signature 'gseaResult'
pairwise_termsim(x, method = "JC", semData = NULL, showCategory = NULL)

## S4 method for signature 'compareClusterResult'
pairwise_termsim(x, method = "JC", semData = NULL, showCategory = NULL)

pairwise_termsim.enrichResult(
  x,
  method = "JC",
  semData = NULL,
  showCategory = NULL
)

pairwise_termsim.compareClusterResult(
  x,
  method = "JC",
  semData = NULL,
  showCategory = NULL
)
```

**Arguments**

x                    enrichment result.  
 method             method of calculating the similarity between nodes, one of "Resnik", "Lin",  
 "Rel", "Jiang", "Wang", and "JC" (Jaccard similarity coefficient) methods.  
 semData            GOSemSimDATA object, can be obtained through GOSemSim:::godata.

showCategory    number of enriched terms to be calculated. The default value is the number of enriched terms, or 200 if the number of enriched terms exceeds 200.

### Details

This function adds a similarity matrix to the termsim slot of the enrichment result. Users can use the method parameter to select the method of calculating the similarity. The Jaccard correlation coefficient (JC) is used by default, and it applies to all situations. When users want to calculate the correlation between GO terms or DO terms, they can also choose "Resnik", "Lin", "Rel" or "Jiang" (they are semantic similarity calculation methods from the 'GOSemSim' package), and at this time, the user needs to provide the semData parameter, which can be obtained through `GOSemSim::godata()`.

### Examples

```
## Not run:
library(clusterProfiler)
library(org.Hs.eg.db)
library(enrichplot)
library(GOSemSim)
library(DOSE)
data(geneList)
gene <- names(geneList)[abs(geneList) > 2]
ego <- enrichGO(gene = gene,
  universe      = names(geneList),
  OrgDb         = org.Hs.eg.db,
  ont           = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff = 0.01,
  qvalueCutoff = 0.05,
  readable     = TRUE)
d <- godata('org.Hs.eg.db', ont="BP")
ego2 <- pairwise_termsim(ego, method="Wang", semData = d)
emapplot(ego2)
emapplot_cluster(ego2)

## End(Not run)
```

---

plotting.clusterProfile

*Internal plot function for plotting compareClusterResult*

---

### Description

Internal plot function for plotting compareClusterResult

### Usage

```
plotting.clusterProfile(
  clProf.reshape.df,
  x = ~Cluster,
  type = "dot",
  colorBy = "p.adjust",
```

```

    by = "geneRatio",
    title = "",
    font.size = 12
  )

```

### Arguments

<code>clProf.reshape.df</code>	data frame of compareCluster result
<code>x</code>	x variable
<code>type</code>	one of dot and bar
<code>colorBy</code>	one of pvalue or p.adjust
<code>by</code>	one of percentage and count
<code>title</code>	graph title
<code>font.size</code>	graph font size

### Value

ggplot object

### Author(s)

Guangchuang Yu <https://yulab-smu.top>

---

pmcplot

*pmcplot*

---

### Description

PubMed Central Trend plot

### Usage

```
pmcplot(query, period, proportion = TRUE)
```

### Arguments

<code>query</code>	query terms
<code>period</code>	period of query in the unit of year
<code>proportion</code>	If TRUE, use query_hits/all_hits, otherwise use query_hits.

### Value

ggplot object

### Author(s)

Guangchuang Yu

---

reexports

*Objects exported from other packages*


---

### Description

These objects are imported from other packages. Follow the links below to see their documentation.

**aplot** [plot\\_list](#)

**DOSE** [theme\\_dose](#)

**enrichit** [geneID](#), [geneInCategory](#), [gseaScores](#)

**ggplot2** [facet\\_grid](#), [ggtitle](#)

**ggtangle** [cnetplot](#), [geom\\_cnet\\_label](#)

---

ridgeplot

*ridgeplot*


---

### Description

Ridgeline plot for GSEA result

### Usage

```
ridgeplot(
  x,
  showCategory = 30,
  fill = "p.adjust",
  core_enrichment = TRUE,
  label_format = 30,
  ...
)

## S4 method for signature 'gseaResult'
ridgeplot(
  x,
  showCategory = 30,
  fill = "p.adjust",
  core_enrichment = TRUE,
  label_format = 30,
  ...
)

ridgeplot.gseaResult(
  x,
  showCategory = 30,
  fill = "p.adjust",
  core_enrichment = TRUE,
  label_format = 30,
  orderBy = "NES",
```

```

    decreasing = FALSE,
    stat = "density_ridges"
  )

```

### Arguments

x	gseaResult object
showCategory	number of categories to display or a vector of terms.
fill	one of "pvalue", "p.adjust", "qvalue"
core_enrichment	whether to use only core_enriched genes
label_format	a numeric value setting the wrap length, alternatively a custom function to format axis labels.
...	additional parameters.
orderBy	The order of the Y-axis
decreasing	logical. Should the orderBy order be increasing or decreasing?
stat	statistic passed to ggridges::geom_density_ridges().

### Value

ggplot object

### Author(s)

Guangchuang Yu

### Examples

```

library(DOSE)
data(geneList)
x <- gseDO(geneList)
ridgeplot(x)

```

---

set\_enrichplot\_color *Helper function to set color scale for enrichplot*

---

### Description

Helper function to set color scale for enrichplot

### Usage

```

set_enrichplot_color(
  colors = get_enrichplot_color(2),
  type = "color",
  name = NULL,
  .fun = NULL,
  reverse = TRUE,
  transform = "identity",
  ...
)

```

**Arguments**

colors	user provided color vector
type	one of 'color', 'colour' or 'fill'
name	name of the color legend
.fun	force to use user provided color scale function
reverse	whether reverse the color scheme
transform	transform the color scale
...	additional parameters

**Value**

a color scale

**Author(s)**

Guangchuang Yu

---

ssplot	<i>ssplot</i>
--------	---------------

---

**Description**

Similarity Space Plot for enrichment analysis

**Usage**

```
ssplot(x, ...)
```

```
## S4 method for signature 'enrichResult'
```

```
ssplot(x, showCategory = 30, ...)
```

```
## S4 method for signature 'gseaResult'
```

```
ssplot(x, showCategory = 30, ...)
```

```
## S4 method for signature 'compareClusterResult'
```

```
ssplot(x, showCategory = 30, ...)
```

```
ssplot.enrichResult(
  x,
  showCategory = 30,
  drfun = NULL,
  dr.params = list(),
  node_label = "group",
  ...
)
```

```
ssplot.compareClusterResult(
  x,
  showCategory = 30,
```

```

    pie = "equal",
    drfun = NULL,
    dr.params = list(),
    node_label = "group",
    ...
  )

```

### Arguments

x	Enrichment result.
...	additional parameters Additional plotting parameters are inherited from <code>emapplot()</code> .
showCategory	number of categories to display or a vector of terms.
drfun	The function used for dimension reduction, e.g. <code>stats::cmdscale</code> (the default), <code>vegan::metaMDS</code> , or <code>ape::pcoa</code> .
dr.params	list, the parameters of <code>tidydr::dr</code> .
node_label	Select which labels to display, one of 'category', 'group', 'all' and 'none'.
pie	one of 'equal' or 'Count' to set the slice ratio of the pies

### Details

Creates 2D visualization of enrichment results using dimension reduction techniques to show relationships between terms based on similarity.

### Value

ggplot object

### Author(s)

Guangchuang Yu

### Examples

```

## Not run:
library(clusterProfiler)
library(org.Hs.eg.db)
library(enrichplot)
library(GOsemSim)
library(DOSE)
data(geneList)
gene <- names(geneList)[abs(geneList) > 2]
ego <- enrichGO(gene = gene,
  universe = names(geneList),
  OrgDb = org.Hs.eg.db,
  ont = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff = 0.01,
  qvalueCutoff = 0.05,
  readable = TRUE)
d <- godata('org.Hs.eg.db', ont="BP")
ego2 <- pairwise_termsim(ego, method = "Wang", semData = d)
ssplot(ego2)

```

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

---

```
treeplot          treeplot
```

---

### Description

Functional grouping tree diagram for enrichment result of over-representation test or gene set enrichment analysis.

Creates hierarchical tree visualization of enriched terms based on similarity

### Usage

```
treeplot(x, ...)

## S4 method for signature 'enrichResult'
treeplot(x, ...)

## S4 method for signature 'gseaResult'
treeplot(x, ...)

## S4 method for signature 'compareClusterResult'
treeplot(x, ...)

treeplot_internal(
  x,
  showCategory = 30,
  color = "p.adjust",
  size_var = c("Count", "setSize"),
  nCluster = 5,
  cluster_method = "ward.D",
  label_format = 30,
  fontsize_tiplab = 4,
  fontsize_cladelab = 4,
  group_color = NULL,
  extend = 0.3,
  highlight = TRUE,
  align = "both",
  hexpand = 0.1,
  tiplab_offset = 0.2,
  cladelab_offset = 1
)
```

### Arguments

x	enrichment result.
...	additional parameters
showCategory	number of enriched terms to display
color	variable to color nodes, e.g. 'p.adjust', 'pvalue', or 'qvalue'

size_var	variable for node size, e.g. 'Count' (for enrichResult) or 'setSize' (for gseaResult)
nCluster	number of clusters for tree cutting
cluster_method	hierarchical clustering method
label_format	wrap length for labels or custom formatting function
fontsize_tiplab	font size for tip labels
fontsize_cladelab	font size for clade labels
group_color	vector of colors for groups
extend	extend length for clade labels
highlight	whether to highlight clades
align	alignment for highlight rectangles
hexpand	expand x limits by amount of xrange * hexpand
tiplab_offset	offset for tip labels
cladelab_offset	offset for clade labels

### Details

This function visualizes gene sets as a tree. Gene sets with high similarity tend to cluster together, making it easier for interpretation.

### Value

ggplot object  
ggplot2 object representing the tree plot

### Examples

```
## Not run:
library(clusterProfiler)
library(org.Hs.eg.db)
library(enrichplot)
library(GOsemSim)
library(ggplot2)
library(DOSE)
data(geneList)
gene <- names(geneList)[abs(geneList) > 2]
ego <- enrichGO(gene = gene,
  universe = names(geneList),
  OrgDb = org.Hs.eg.db,
  ont = "BP",
  pAdjustMethod = "BH",
  pvalueCutoff = 0.01,
  qvalueCutoff = 0.05,
  readable = TRUE)
d <- godata('org.Hs.eg.db', ont="BP")
ego2 <- pairwise_termsim(ego, method = "Wang", semData = d)
treepLOT(ego2, showCategory = 30)
# use `highlight = FALSE` to remove ggtree::geom_highlight() layer.
```

```

treeplot(ego2, showCategory = 30, hilight = FALSE)
# use `offset` parameter to adjust the distance of bar and tree.
treeplot(ego2, showCategory = 30, hilight = FALSE, offset = rel(1.5))
# use `offset_tiplab` parameter to adjust the distance of nodes and branches.
treeplot(ego2, showCategory = 30, hilight = FALSE, offset_tiplab = rel(1.5))
keep <- rownames(ego2@termsim)[c(1:10, 16:20)]
keep
treeplot(ego2, showCategory = keep)
treeplot(ego2, showCategory = 20,
  group_color = c("#999999", "#E69F00", "#56B4E9", "#009E73", "#F0E442"))
# It can also graph compareClusterResult
data(gcSample)
xx <- compareCluster(gcSample, fun="enrichKEGG",
  organism="hsa", pvalueCutoff=0.05)
xx <- pairwise_termsim(xx)
treeplot(xx)

# use `geneClusterPanel` to change the gene cluster panel.
treeplot(xx, geneClusterPanel = "dotplot")

treeplot(xx, geneClusterPanel = "pie")

## End(Not run)

```

---

upsetplot

*upsetplot method*


---

## Description

upsetplot method generics

Upsetplot

## Usage

```
upsetplot(x, ...)
```

```
## S4 method for signature 'enrichResult'
upsetplot(x, n = 10, ...)
```

```
## S4 method for signature 'gseaResult'
upsetplot(x, n = 10, ...)
```

## Arguments

x	object
...	additional parameters
n	number of categories to be plotted

## Value

plot

**Author(s)**

Guangchuang Yu

**Examples**

```
library(DOSE)
data(geneList)
de <- names(geneList)[1:100]
x <- enrichDO(de)
upsetplot(x, 8)
```

---

`volplot`*volplot*

---

**Description**

Volcano plot for enrichment result

**Usage**

```
volplot(
  x,
  color = "zScore",
  xintercept = 1,
  yintercept = 2,
  showCategory = 5,
  label_format = 30,
  ...
)

## S4 method for signature 'enrichResult'
volplot(
  x,
  color = "zScore",
  xintercept = 1,
  yintercept = 2,
  showCategory = 5,
  label_format = 30,
  ...
)

volplot.enrichResult(
  x,
  color = "zScore",
  xintercept = 1,
  yintercept = 2,
  showCategory = 5,
  label_format = 30,
  font.size = 12,
  size = 5
)
```

**Arguments**

x	enrichment result.
color	selected variable to color the dots
xintercept	value to set x-intercept
yintercept	value to set y-intercept
showCategory	number of most significant enriched terms or selected terms to display determined by the variable selected to color the dots
label_format	a numeric value setting the wrap length, alternatively a custom function to format axis labels.
...	Additional parameters
font.size	font size for theme_dose()
size	font size to label selected categories specified by showCategory

**Value**

ggplot object

**Author(s)**

Guangchuang Yu

**Examples**

```
library(DOSE)
data(geneList)
de <- names(geneList)[1:100]
x <- enrichDO(de)
volplot(x)
```

# Index

- \* **datasets**
  - enrichplot\_point\_shape, 15
- \* **internal**
  - enrichplot-common-params, 14
  - enrichplot-package, 2
  - enrichplot-term-params, 14
  - reexports, 30
- as.data.frame.compareClusterResult, 3
- autofacet, 3
- barplot.enrichResult, 4
- cnetplot, 7, 30
- cnetplot (reexports), 30
- cnetplot.compareClusterResult (cnetplot.enrichResult), 5
- cnetplot.enrichResult, 5
- cnetplot.gseaResult (cnetplot.enrichResult), 5
- color\_palette, 7
- dotplot, 8
- dotplot,compareClusterResult,ANY-method (dotplot), 8
- dotplot,compareClusterResult-method (dotplot), 8
- dotplot,enrichResult-method (dotplot), 8
- dotplot,enrichResultList,ANY-method (dotplot), 8
- dotplot,enrichResultList-method (dotplot), 8
- dotplot,gseaResult-method (dotplot), 8
- dotplot,gseaResultList,ANY-method (dotplot), 8
- dotplot,gseaResultList-method (dotplot), 8
- dotplot.compareClusterResult (dotplot), 8
- dotplot.enrichResult (dotplot), 8
- dotplot2, 11
- emapplot, 12
- emapplot(), 33
- emapplot,compareClusterResult-method (emapplot), 12
- emapplot,enrichResult-method (emapplot), 12
- emapplot,gseaResult-method (emapplot), 12
- emapplot\_internal (emapplot), 12
- enrichplot (enrichplot-package), 2
- enrichplot-common-params, 14
- enrichplot-package, 2
- enrichplot-term-params, 14
- enrichplot\_point\_shape, 15
- facet\_grid, 30
- facet\_grid (reexports), 30
- fortify.compareClusterResult, 15
- fortify.enrichResult (fortify.compareClusterResult), 15
- geneID, 30
- geneID (reexports), 30
- geneInCategory, 30
- geneInCategory (reexports), 30
- geom\_cnet\_label, 30
- geom\_cnet\_label (reexports), 30
- geom\_gsea\_gene, 16
- get\_enrichplot\_color, 17
- ggtable, 17
- ggtitle, 30
- ggtitle (reexports), 30
- goplot, 18
- goplot,enrichResult-method (goplot), 18
- goplot,gseaResult-method (goplot), 18
- GOSemSim::godata(), 28
- gseadist, 19
- gseaplot, 20
- gseaplot,gseaResult-method (gseaplot), 20
- gseaplot.gseaResult (gseaplot), 20
- gseaplot2, 21
- gsearank, 22
- gseaScores, 30
- gseaScores (reexports), 30

- gsInfo, [23](#)
- heatmap, [23](#)
- heatmap, enrichResult-method (heatmap), [23](#)
- heatmap, gseaResult-method (heatmap), [23](#)
- heatmap.enrichResult (heatmap), [23](#)
- hplot, [24](#)
- manhattanplot, [25](#)
- manhattanplot, compareClusterResult, ANY-method (manhattanplot), [25](#)
- manhattanplot, compareClusterResult-method (manhattanplot), [25](#)
- manhattanplot, enrichResult-method (manhattanplot), [25](#)
- manhattanplot, enrichResultList, ANY-method (manhattanplot), [25](#)
- manhattanplot, enrichResultList-method (manhattanplot), [25](#)
- manhattanplot, gseaResult-method (manhattanplot), [25](#)
- manhattanplot, gseaResultList, ANY-method (manhattanplot), [25](#)
- manhattanplot, gseaResultList-method (manhattanplot), [25](#)
- manhattanplot, list, ANY-method (manhattanplot), [25](#)
- manhattanplot, list-method (manhattanplot), [25](#)
- pairwise\_termsim, [27](#)
- pairwise\_termsim(), [13](#)
- pairwise\_termsim, compareClusterResult-method (pairwise\_termsim), [27](#)
- pairwise\_termsim, enrichResult-method (pairwise\_termsim), [27](#)
- pairwise\_termsim, gseaResult-method (pairwise\_termsim), [27](#)
- pairwise\_termsim.compareClusterResult (pairwise\_termsim), [27](#)
- pairwise\_termsim.enrichResult (pairwise\_termsim), [27](#)
- plot\_list, [30](#)
- plot\_list (reexports), [30](#)
- plotting.clusterProfile, [28](#)
- pncplot, [29](#)
- reexports, [30](#)
- ridgeplot, [30](#)
- ridgeplot, gseaResult-method (ridgeplot), [30](#)
- ridgeplot.gseaResult (ridgeplot), [30](#)
- set\_enrichplot\_color, [31](#)
- ssplot, [32](#)
- ssplot, compareClusterResult-method (ssplot), [32](#)
- ssplot, enrichResult-method (ssplot), [32](#)
- ssplot, gseaResult-method (ssplot), [32](#)
- ssplot.compareClusterResult (ssplot), [32](#)
- ssplot.enrichResult (ssplot), [32](#)
- theme\_dose, [30](#)
- theme\_dose (reexports), [30](#)
- treepLOT, [34](#)
- treepLOT, compareClusterResult-method (treepLOT), [34](#)
- treepLOT, enrichResult-method (treepLOT), [34](#)
- treepLOT, gseaResult-method (treepLOT), [34](#)
- treepLOT\_internal (treepLOT), [34](#)
- upsetplot, [36](#)
- upsetplot, enrichResult, ANY-method (upsetplot), [36](#)
- upsetplot, enrichResult-method (upsetplot), [36](#)
- upsetplot, gseaResult (upsetplot), [36](#)
- upsetplot, gseaResult-method (upsetplot), [36](#)
- volplot, [37](#)
- volplot, enrichResult-method (volplot), [37](#)
- volplot.enrichResult (volplot), [37](#)