

# Package ‘enrichplot’

November 7, 2024

**Title** Visualization of Functional Enrichment Result

**Version** 1.26.2

**Description** The 'enrichplot' package implements several visualization methods for interpreting functional enrichment results obtained from ORA or GSEA analysis. It is mainly designed to work with the 'clusterProfiler' package suite. All the visualization methods are developed based on 'ggplot2' graphics.

**Depends** R (>= 3.5.0)

**Imports** aplot (>= 0.2.1), DOSE (>= 3.31.2), ggfun (>= 0.1.7), ggnewscale, ggplot2, ggrepel (>= 0.9.0), ggtangle (>= 0.0.4), graphics, grid, igraph, methods, plyr, purrr, RColorBrewer, reshape2, rlang, stats, utils, scatterpie, GOsemSim (>= 2.31.2), magrittr, ggtree, yulab.utils (>= 0.1.6)

**Suggests** clusterProfiler, dplyr, europepmc, ggarchery, ggupset, knitr, rmarkdown, org.Hs.eg.db, prettydoc, tibble, tidyr, ggforce, ggHoriPlot, AnnotationDbi, ggplotify, ggribes, grDevices, gridExtra, ggstar, scales, ggtreeExtra, tidydr

**Remotes** YuLab-SMU/tidydr

**VignetteBuilder** knitr

**License** Artistic-2.0

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

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

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

**Encoding** UTF-8

**RoxygenNote** 7.3.2

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

**git\_branch** RELEASE\_3\_20

**git\_last\_commit** 0075570

**git\_last\_commit\_date** 2024-11-06

**Repository** Bioconductor 3.20

**Date/Publication** 2024-11-06

**Author** Guangchuang Yu [aut, cre] (<<https://orcid.org/0000-0002-6485-8781>>),  
Chun-Hui Gao [ctb] (<<https://orcid.org/0000-0002-1445-7939>>)

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

## Contents

enrichplot-package	2
autofacet	3
barplot.enrichResult	3
cnetplot.enrichResult	4
color_palette	6
dotplot	7
dotplot2	10
emapplot	11
fortify.compareClusterResult	13
geom_gsea_gene	14
ggtable	15
goplot	16
gseadist	17
gseaplot	17
gseaplot2	19
gsearank	20
gsInfo	20
heatplot	21
hplot	22
pairwise_termsim	22
plotting.clusterProfile	24
pmcplot	25
reexports	25
ridgeplot	26
set_enrichplot_color	27
ssplot	28
treeplot	30
upsetplot	34
volplot	35
<b>Index</b>	<b>37</b>

---

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

---

### Description

The 'enrichplot' package implements several visualization methods for interpreting functional enrichment results obtained from ORA or GSEA analysis. It is mainly designed to work with the 'clusterProfiler' package suite. All the visualization methods are developed based on 'ggplot2' 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/biomedical-knowledge-mining-book/>
- Report bugs at <https://github.com/GuangchuangYu/enrichplot/issues>

---

autofacet

*automatically split barplot or dotplot into several facets*

---

**Description**

automatically split barplot or dotplot into several facets

**Usage**

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

**Arguments**

by	one of 'row' or 'column'
scales	wether '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 show
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
...	other parameter, ignored

**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)
categorys <- c("urinary bladder cancer", "bronchiolitis obliterans",
               "aortic aneurysm", "esophageal cancer")
barplot(x, showCategory = categorys)
```

---

cnetplot.enrichResult *cnetplot*

---

**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,
  node_label = "all",
```

```
    foldChange = 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,  
  node_label = "all",  
  foldChange = NULL,  
  highlight = "none",  
  highlight_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,  
  node_label = "all",  
  foldChange = NULL,  
  highlight = "none",  
  highlight_alpha = 0.3,  
  pie = "equal",  
  ...  
)
```

### Arguments

x	input object
layout	network layout
showCategory	selected category to be displayed
color_category	color of category node
size_category	relative size of the category
color_item	color of item node

size_item	relative size of the item (e.g., genes)
color_edge	color of edge
size_edge	relative size of edge
node_label	one of 'all', 'none', 'category' and 'item'
foldChange	numeric values to color the item (e.g, foldChange of gene expression values)
hilight	selected category to be highlighted
hilight_alpha	transparent value for not selected to be highlight
...	additional parameters
pie	one of 'equal' or 'Count' to set the slice ratio of the pies

**See Also**

[cnetplot][ggtangle::cnetplot]

---

color_palette	<i>color_palette</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

dotplot 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 that used to color enriched terms, e.g. 'pvalue', 'p.adjust' or 'qvalue'
showCategory	A number or a list of terms. If it is a number, the first n terms will be displayed. If it is a list of terms, the selected terms will be displayed.
size	variable that 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
decreasing	logical. Should the orderBy order be increasing or decreasing?
colorBy	variable that 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, a.k.a 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)
categorys <- c("pre-malignant neoplasm", "intestinal disease",
              "breast ductal carcinoma", "non-small cell lung carcinoma")
dotplot(x, showCategory = categorys)
# 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",  
  pie = "equal",  
  group = FALSE,  
  group_style = "ggforce",  
  label_group_style = "shadowntext",  
  label_format = 30,  
  clusterFunction = stats::kmeans,  
  nWords = 4,  
  nCluster = NULL  
)
```

**Arguments**

x	Enrichment result.
...	Additional parameters
showCategory	A number or a vector of terms. If it is a number, the first n terms will be displayed. If it is a vector of terms, the selected terms will be displayed.
layout	igraph layout
color	Variable that 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 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 be displayed, one of 'category', 'group', 'all' and 'none'.
pie	one of 'equal' or 'Count' to set the slice ratio of the pies
group	logical, if TRUE, group the category.
group_style	style of ellipse, one of "ggforce" an "polygon".
label_group_style	style of group label, one of "shadowtext" and "ggforce".
label_format	a numeric value sets wrap length, alternatively a custom function to format axis labels.
clusterFunction	function of Clustering method, such as stats::kmeans(the default), cluster::clara, cluster::fanny or cluster::pam.
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 function 'pairwise\_termsim', the details of 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)
  emaplot(x2)
  # use `layout` to change the layout of map
  emaplot(x2, layout = "star")
  # use `showCategory` to select the displayed terms. It can be a number of a vector of terms.
  emaplot(x2, showCategory = 10)
  categorys <- c("pre-malignant neoplasm", "intestinal disease",
                "breast ductal carcinoma")
  emaplot(x2, showCategory = categorys)

  # 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)
  emaplot(xx2)

## End(Not run)
```

---

```
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

---

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 enriched terms to display
<code>color</code>	variable that used to color enriched terms, e.g. <code>pvalue</code> , <code>p.adjust</code> or <code>qvalue</code>
<code>layout</code>	layout of the map
<code>geom</code>	label geom, one of 'label' or 'text'
<code>...</code>	additional parameter

**Value**

ggplot object



**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 analyzing result of GSEA

**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**

x	object of gsea result
geneSetID	geneSet ID
by	one of "runningScore" or "position"
title	plot title
...	additional parameters
color	color of line segments
color.line	color of running enrichment score line
color.vline	color of vertical line which 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,
  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
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	<i>heatplot</i>
----------	-----------------

---

**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,  
  symbol = "rect",  
  foldChange = NULL,  
  pvalue = NULL,  
  label_format = 30  
)
```

**Arguments**

x	enrichment result.
showCategory	number of enriched terms to display
...	Additional parameters
symbol	symbol of the nodes, one of "rect"(the default) and "dot" by default wraps names longer than 30 characters
foldChange	fold Change.
pvalue	pvalue of genes
label_format	a numeric value sets wrap length, alternatively a custom function to format axis labels.

**Value**

ggplot object

**Author(s)**

Guangchuang Yu

**Examples**

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

---

hplot	<i>hplot</i>
-------	--------------

---

**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

---

pairwise_termsim	<i>pairwise_termsim</i>
------------------	-------------------------

---

**Description**

Get the similarity matrix

**Usage**

```
pairwise_termsim(x, method = "JC", semData = NULL, showCategory = 200)

## S4 method for signature 'enrichResult'
pairwise_termsim(x, method = "JC", semData = NULL, showCategory = 200)

## S4 method for signature 'gseaResult'
pairwise_termsim(x, method = "JC", semData = NULL, showCategory = 200)

## S4 method for signature 'compareClusterResult'
pairwise_termsim(x, method = "JC", semData = NULL, showCategory = 200)
```

```

pairwise_termsim.enrichResult(
  x,
  method = "JC",
  semData = NULL,
  showCategory = 200
)

pairwise_termsim.compareClusterResult(
  x,
  method = "JC",
  semData = NULL,
  showCategory = 200
)

```

### 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 <a href="#">godata</a> function in GOSemSim package.
showCategory	number of enriched terms to display, default value is 200.

### Details

This function add similarity matrix to the termsim slot of enrichment result. Users can use the 'method' parameter to select the method of calculating 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 GOSemSim packages), and at this time, the user needs to provide 'semData' parameter, which can be obtained through [godata](#) function in GOSemSim package.

### 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  
*plotting-clusterProfile*

---

### 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

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

### Value

ggplot object

### Author(s)

Guangchuang Yu <https://guangchuangyu.github.io>



---

pmcplot	<i>pmcplot</i>
---------	----------------

---

**Description**

PubMed Central Trend plot

**Usage**

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

**Arguments**

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

**Value**

ggplot object

**Author(s)**

guangchuang yu

---

reexports	<i>Objects exported from other packages</i>
-----------	---

---

**Description**

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

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

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

**ggtangle** [cnetplot](#)

---

 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
)
```

### Arguments

x	gseaResult object
showCategory	A number or a vector of terms. If it is a number, the first n terms will be displayed. If it is a vector of terms, the selected terms will be displayed.
fill	one of "pvalue", "p.adjust", "qvalue"
core_enrichment	whether only using core_enriched genes
label_format	a numeric value sets wrap length, alternatively a custom function to format axis labels.
...	additional parameters by default wraps names longer than 30 characters

orderBy	The order of the Y-axis
decreasing	logical. Should the orderBy order be increasing or decreasing?

**Value**

ggplot object

**Author(s)**

Guangchuang Yu

**Examples**

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

---

set\_enrichplot\_color    *set\_enrichplot\_color*

---

**Description**

helper function to set color for enrichplot

**Usage**

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

**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, default is TRUE as it is more significant for lower pvalue
...	additional parameter that passed to the color scale function

**Value**

a color scale

---

 ssplot

*ssplot*


---

### Description

Similarity space plot of enrichment analysis results.

### 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(),
  group = TRUE,
  node_label = "group",
  ...
)

ssplot.compareClusterResult(
  x,
  showCategory = 30,
  pie = "equal",
  drfun = NULL,
  dr.params = list(),
  group = TRUE,
  node_label = "group",
  ...
)
```

### Arguments

x	Enrichment result.
...	additional parameters

additional parameters can refer the following parameters.

- `color` Variable that used to color enriched terms, e.g. `'pvalue'`, `'p.adjust'` or `'qvalue'`. the starting position of each text label.
- `size_edge` Scale of line width.

- `min_edge` The minimum similarity threshold for whether two nodes are connected, should be between 0 and 1, default value is 0.2.
- `size_category` Number indicating the amount by which plotting category nodes should be scaled relative to the default.
- `label_style` style of group label, one of "shadowtext" and "ggforce".
- `group` Logical, if TRUE, the grouping legend will be displayed. The default is FALSE.
- `nWords` Numeric, the number of words in the cluster tags, the default value is 4.
- `label_format` a numeric value sets wrap length, alternatively a custom function to format axis labels.
- `clusterFunction` function of Clustering method, such as `stats::kmeans` (the default), `cluster::clara`, `cluster::fanny` or `cluster::pam`.
- `nCluster` Numeric, the number of clusters, the default value is square root of the number of nodes.

additional parameters can refer the `emapplot` function: [emapplot](#).

<code>showCategory</code>	A number or a vector of terms. If it is a number, the first n terms will be displayed. If it is a vector of terms, the selected terms will be displayed.
<code>drfun</code>	The function used for dimension reduction, e.g. <code>stats::cmdscale</code> (the default), <code>vegan::metaMDS</code> , or <code>ape::pcoa</code> .
<code>dr.params</code>	list, the parameters of <code>tidydr::dr</code> . one of 'category', 'group', 'all' and 'none'.
<code>group</code>	logical, if TRUE, group the category.
<code>node_label</code>	Select which labels to be displayed, one of 'category', 'group', 'all' and 'none'.
<code>pie</code>	one of 'equal' or 'Count' to set the slice ratio of the pies

## Value

ggplot object

## 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.

### 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.enrichResult(
  x,
  showCategory = 30,
  color = "p.adjust",
  nWords = 4,
  nCluster = 5,
  cex_category = 1,
  label_format = NULL,
  label_format_cladelab = 30,
  label_format_tiplab = NULL,
  fontsize = 4,
  offset = rel(1),
  offset_tiplab = rel(1),
  hclust_method = "ward.D",
  group_color = NULL,
  extend = 0.3,
  hilight = TRUE,
  hexpand = 0.1,
  align = "both",
  hilight.params = list(hilight = TRUE, align = "both"),
  offset.params = list(bar_tree = rel(1), tiplab = rel(1), extend = 0.3, hexpand = 0.1),
  cluster.params = list(method = "ward.D", n = 5, color = NULL, label_words_n = 4,
    label_format = 30),
  ...
)
```

```
treeplot.compareClusterResult(
  x,
  showCategory = 5,
  color = "p.adjust",
```

```

nWords = 4,
nCluster = 5,
cex_category = 1,
split = NULL,
label_format = NULL,
label_format_cladelab = 30,
label_format_tiplab = NULL,
fontsize = 4,
offset = rel(1),
pie = "equal",
legend_n = 3,
offset_tiplab = rel(1),
hclust_method = "ward.D",
group_color = NULL,
extend = 0.3,
highlight = TRUE,
geneClusterPanel = "heatMap",
hexpand = 0.1,
align = "both",
cluster.params = list(method = "ward.D", n = 5, color = NULL, label_words_n = 4,
  label_format = 30),
highlight.params = list(highlight = TRUE, align = "both"),
clusterPanel.params = list(clusterPanel = "heatMap", pie = "equal", legend_n = 3,
  colnames_angle = 0),
offset.params = list(bar_tree = rel(1), tiplab = rel(1), extend = 0.3, hexpand = 0.1),
...
)

```

### Arguments

x	enrichment result.
...	additional parameters
showCategory	number of enriched terms to display
color	variable that used to color enriched terms, e.g. pvalue, p.adjust or qvalue
nWords	The number of words in the cluster tags. Will be removed in the next version.
nCluster	The number of clusters, the default value is 5. Will be removed in the next version.
cex_category	Number indicating the amount by which plotting category. nodes should be scaled relative to the default. Will be removed in the next version.
label_format	a numeric value sets wrap length, alternatively a custom function to format axis labels.
label_format_cladelab	label_format for group labels, a numeric value sets wrap length, alternatively a custom function to format axis labels. Will be removed in the next version.
label_format_tiplab	label_format for tiplabs, a numeric value sets wrap length, alternatively a custom function to format axis labels. Will be removed in the next version.
fontsize	The size of text, default is 4.
offset	rel object or numeric value, distance bar and tree, offset of bar and text from the clade, default is rel(1), meaning 1 * 1.2 * x_range_of_tree plus distance_between_tree_and_tiplab

	( $1 * (1.2 * x\_range\_of\_tree + distance\_between\_tree\_and\_tiplab)$ ). Will be removed in the next version.
offset_tiplab	tiplab offset, rel object or numeric value, the bigger the number, the farther the distance between the node and the branch. The default is rel(1), when geneClusterPanel = "pie", meaning $1 * max\_radius\_of\_the\_pies$ ; when geneClusterPanel = "heatMap", meaning $1 * 0.16 * column\_number\_of\_heatMap * x\_range\_of\_tree$ ; when geneClusterPanel = "dotplot", meaning $1 * 0.09 * column\_number\_of\_dotplot * x\_range\_of\_tree$ . Will be removed in the next version.
hclust_method	Method of hclust. This should be (an unambiguous abbreviation of) one of "ward.D", "ward.D2", "single", "complete", "average" (= UPGMA), "mcquitty" (= WPGMA), "median" (= WPGMC) or "centroid" (= UPGMC). Will be removed in the next version.
group_color	A vector of group colors, the length of the vector should be the same as nCluster. Will be removed in the next version.
extend	Numeric, extend the length of bar, default is 0.3. Will be removed in the next version.
highlight	Logical value, if TRUE(default), add ggtree::geom_highlight() layer. Will be removed in the next version.
hexpand	expand x limits by amount of xrange * hexpand. Will be removed in the next version.
align	control the align direction of the edge of high light rectangular. Options is 'none', 'left', 'right', 'both (default)'. Will be removed in the next version.
highlight.params	list, the parameters to control the attributes of highlight layer. see the highlight.params in the following. highlight.params control the attributes of highlight layer, it can be referred to the following parameters: <ul style="list-style-type: none"> <li>• highlight Logical value, if TRUE(default), add ggtree::geom_highlight() layer.</li> <li>• align control the align direction of the edge of high light rectangular. Options is 'none', 'left', 'right', 'both (default)'.</li> </ul>
offset.params	list, the parameters to control the offset. see the offset.params in the following. offset.params control the attributes of offset, it can be referred to the following parameters: <ul style="list-style-type: none"> <li>• bar_tree rel object or numeric value, distance bar and tree, offset of bar and text from the clade, default is rel(1), meaning <math>1 * 1.2 * x\_range\_of\_tree</math> plus distance_between_tree_and_tiplab (<math>1 * (1.2 * x\_range\_of\_tree + distance\_between\_tree\_and\_tiplab)</math>).</li> <li>• tiplab tiplab offset, rel object or numeric value, the bigger the number, the farther the distance between the node and the branch. The default is rel(1), when clusterPanel = "pie", meaning <math>1 * max\_radius\_of\_the\_pies</math>; when clusterPanel = "heatMap", meaning <math>1 * 0.16 * column\_number\_of\_heatMap * x\_range\_of\_tree</math>; when clusterPanel = "dotplot", meaning <math>1 * 0.09 * column\_number\_of\_dotplot * x\_range\_of\_tree</math>.</li> <li>• extend Numeric, extend the length of bar, default is 0.3.</li> <li>• hexpand expand x limits by amount of xrange * hexpand.</li> </ul>
cluster.params	list, the parameters to control the attributes of highlighted nodes and edges. see the cluster.params in the following. cluster.params control the attributes of highlight, it can be referred to the following parameters: <ul style="list-style-type: none"> <li>• method function of Clustering method, such as stats::kmeans(the default), cluster::clara, cluster::fanny or cluster::pam.</li> </ul>



- `n` Numeric, the number of clusters, the default value is square root of the number of nodes.
  - `color` A vector of group colors, the length of the vector should be the same as `nCluster`.
  - `label_words_n` Numeric, the number of words in the cluster tags, the default value is 4.
  - `label_format` A numeric value sets wrap length, alternatively a custom function to format axis labels.
- `split` Separate result by 'category' variable.
- `pie` Used only when `geneClusterPanel = "pie"`, proportion of clusters in the pie chart, one of 'equal' (default) and 'Count'. Will be removed in the next version.
- `legend_n` Number of circle in legend, the default value is 3. Will be removed in the next version.
- `geneClusterPanel` one of "heatMap"(default), "dotplot", "pie". Will be removed in the next version.
- `clusterPanel.params` list, the parameters to control the attributes of cluster panel. see the `clusterPanel.params` in the following. `clusterPanel.params` control the attributes of cluster panel, it can be referred to the following parameters:
- `clusterPanel` one of "heatMap"(default), "dotplot", "pie".
  - `pie` Used only when `ClusterPanel = "pie"`, proportion of clusters in the pie chart, one of 'equal' (default) and 'Count'.
  - `legend_n` number of circle in legend.
  - `colnames_angle` set the angle of colnames.

## 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

## 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, highlight = FALSE)
# use `offset` parameter to adjust the distance of bar and tree.
treeplot(ego2, showCategory = 30, highlight = FALSE, offset = rel(1.5))
# use `offset_tiplab` parameter to adjust the distance of nodes and branches.
treeplot(ego2, showCategory = 30, highlight = 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

## 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**

```
require(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 sets 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

- \* **internal**
  - enrichplot-package, 2
  - reexports, 25
- autofacet, 3
- barplot.enrichResult, 3
- cnetplot, 25
- cnetplot (reexports), 25
- cnetplot.compareClusterResult (cnetplot.enrichResult), 4
- cnetplot.enrichResult, 4
- cnetplot.gseaResult (cnetplot.enrichResult), 4
- color\_palette, 6
- dotplot, 7
- dotplot,compareClusterResult,ANY-method (dotplot), 7
- dotplot,compareClusterResult-method (dotplot), 7
- dotplot,enrichResult-method (dotplot), 7
- dotplot,enrichResultList,ANY-method (dotplot), 7
- dotplot,enrichResultList-method (dotplot), 7
- dotplot,gseaResult-method (dotplot), 7
- dotplot,gseaResultList,ANY-method (dotplot), 7
- dotplot,gseaResultList-method (dotplot), 7
- dotplot.compareClusterResult (dotplot), 7
- dotplot.enrichResult (dotplot), 7
- dotplot2, 10
- emapplot, 11, 29
- emapplot,compareClusterResult-method (emapplot), 11
- emapplot,enrichResult-method (emapplot), 11
- emapplot,gseaResult-method (emapplot), 11
- emapplot\_internal (emapplot), 11
- enrichplot (enrichplot-package), 2
- enrichplot-package, 2
- facet\_grid, 25
- facet\_grid (reexports), 25
- fortify.compareClusterResult, 13
- fortify.enrichResult (fortify.compareClusterResult), 13
- geom\_gsea\_gene, 14
- ggtable, 15
- ggtitle, 25
- ggtitle (reexports), 25
- godata, 23
- goplot, 16
- goplot,enrichResult-method (goplot), 16
- goplot,gseaResult-method (goplot), 16
- gseadist, 17
- gseaplot, 17
- gseaplot,gseaResult-method (gseaplot), 17
- gseaplot.gseaResult (gseaplot), 17
- gseaplot2, 19
- gsearank, 20
- gsInfo, 20
- heatplot, 21
- heatplot,enrichResult-method (heatplot), 21
- heatplot,gseaResult-method (heatplot), 21
- heatplot.enrichResult (heatplot), 21
- hplot, 22
- pairwise\_termsim, 12, 22
- pairwise\_termsim,compareClusterResult-method (pairwise\_termsim), 22
- pairwise\_termsim,enrichResult-method (pairwise\_termsim), 22
- pairwise\_termsim,gseaResult-method (pairwise\_termsim), 22
- pairwise\_termsim.compareClusterResult (pairwise\_termsim), 22

- pairwise\_termsim.enrichResult
  - (pairwise\_termsim), 22
- plot\_list, 25
- plot\_list(reexports), 25
- plotting.clusterProfile, 24
- pmcplot, 25
  
- reexports, 25
- ridgeplot, 26
- ridgeplot,gseaResult-method
  - (ridgeplot), 26
- ridgeplot.gseaResult(ridgeplot), 26
  
- set\_enrichplot\_color, 27
- ssplot, 28
- ssplot,compareClusterResult-method
  - (ssplot), 28
- ssplot,enrichResult-method(ssplot), 28
- ssplot,gseaResult-method(ssplot), 28
- ssplot.compareClusterResult(ssplot), 28
- ssplot.enrichResult(ssplot), 28
  
- treepplot, 30
- treepplot,compareClusterResult-method
  - (treepplot), 30
- treepplot,enrichResult-method
  - (treepplot), 30
- treepplot,gseaResult-method(treepplot), 30
- treepplot.compareClusterResult
  - (treepplot), 30
- treepplot.enrichResult(treepplot), 30
  
- upsetplot, 34
- upsetplot,enrichResult,ANY-method
  - (upsetplot), 34
- upsetplot,enrichResult-method
  - (upsetplot), 34
- upsetplot,gseaResult(upsetplot), 34
- upsetplot,gseaResult-method
  - (upsetplot), 34
  
- volplot, 35
- volplot,enrichResult-method(volplot), 35
- volplot.enrichResult(volplot), 35