

# Package ‘enrichplot’

April 12, 2022

**Title** Visualization of Functional Enrichment Result

**Version** 1.14.2

**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, DOSE (>= 3.16.0), ggplot2, ggraph, graphics, grid, igraph, methods, plyr, purrr, RColorBrewer, reshape2, stats, utils, scatterpie, shadowtext, GOSemSim, magrittr, ggtree, yulab.utils (>= 0.0.4)

**Suggests** clusterProfiler, dplyr, europepmc, ggupset, knitr, rmarkdown, org.Hs.eg.db, prettydoc, tibble, tidyr, ggforce, AnnotationDbi, ggplotify, ggridges, grDevices, gridExtra, ggnewscale, ggrepel (>= 0.9.0), ggstar, treeio, scales, tidytree, ggtreeExtra

**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** 7.1.2

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

**git\_branch** RELEASE\_3\_14

**git\_last\_commit** 7ffc704

**git\_last\_commit\_date** 2022-02-23

**Date/Publication** 2022-04-12

**Author** Guangchuang Yu [aut, cre] (<<https://orcid.org/0000-0002-6485-8781>>), Erqiang Hu [ctb]

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

R topics documented:

barplot.enrichResult . . . . .	2
cnetplot . . . . .	3
color_palette . . . . .	6
dotplot . . . . .	6
emapplot . . . . .	9
emapplot_cluster . . . . .	13
fortify.compareClusterResult . . . . .	14
ggtable . . . . .	15
goplot . . . . .	15
gseadist . . . . .	17
gseaplot . . . . .	18
gseaplot2 . . . . .	19
gsearank . . . . .	20
gsInfo . . . . .	21
heatplot . . . . .	21
pairwise_termsim . . . . .	22
plotting.clusterProfile . . . . .	24
pmcplot . . . . .	25
ridgeplot . . . . .	25
treeplot . . . . .	27
upsetplot . . . . .	30
<b>Index</b>	<b>32</b>

---

barplot.enrichResult	<i>barplot</i>
----------------------	----------------

---

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("pre-malignant neoplasm", "intestinal disease",
               "breast ductal carcinoma", "non-small cell lung carcinoma")
barplot(x, showCategory = categorys)
```

---

cnetplot

*cnetplot*


---

**Description**

Gene-Concept Network

**Usage**

```
cnetplot(x, showCategory = 5, foldChange = NULL, layout = "kk", ...)

## S4 method for signature 'enrichResult'
cnetplot(x, showCategory = 5, foldChange = NULL, layout = "kk", ...)

## S4 method for signature 'list'
cnetplot(x, showCategory = 5, foldChange = NULL, layout = "kk", ...)
```

```
## S4 method for signature 'gseaResult'
cnetplot(x, showCategory = 5, foldChange = NULL, layout = "kk", ...)

## S4 method for signature 'compareClusterResult'
cnetplot(x, showCategory = 5, foldChange = NULL, layout = "kk", ...)

cnetplot.enrichResult(
  x,
  showCategory = 5,
  foldChange = NULL,
  layout = "kk",
  colorEdge = FALSE,
  circular = FALSE,
  node_label = "all",
  cex_category = 1,
  cex_gene = 1,
  cex_label_category = 1,
  cex_label_gene = 1,
  color_category = "#E5C494",
  color_gene = "#B3B3B3",
  shadowtext = "all",
  ...
)
```

## Arguments

<code>x</code>	Enrichment result.
<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>foldChange</code>	Fold Change of nodes, the default value is NULL. If the user provides the Fold Change value of the nodes, it can be used to set the color of the gene node.
<code>layout</code>	Layout of the map, e.g. 'star', 'circle', 'gem', 'dh', 'graphopt', 'grid', 'mds', 'randomly', 'fr', 'kk', 'drl' or 'lgl'.
<code>...</code>	Additional parameters
<code>colorEdge</code>	Logical, whether coloring edge by enriched terms, the default value is FALSE.
<code>circular</code>	Logical, whether using circular layout, the default value is FALSE.
<code>node_label</code>	Select which labels to be displayed. one of 'category', 'gene', 'all'(the default) and 'none'.
<code>cex_category</code>	Number indicating the amount by which plotting category nodes should be scaled relative to the default, the default value is 1.
<code>cex_gene</code>	Number indicating the amount by which plotting gene nodes should be scaled relative to the default, the default value is 1.
<code>cex_label_category</code>	Scale of category node label size, the default value is 1.
<code>cex_label_gene</code>	Scale of gene node label size, the default value is 1.

color\_category Color of category node.  
color\_gene Color of gene node.  
shadowtext select which node labels to use shadow font, one of 'category', 'gene', 'all' and 'none', default is 'all'.

## Details

plot linkages of genes and enriched concepts (e.g. GO categories, KEGG pathways)

## 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)
cnetplot(x2)
# use `layout` to change the layout of map
cnetplot(x2, layout = "star")
# use `showCategory` to select the displayed terms. It can be a number or a vector of terms.
cnetplot(x2, showCategory = 10)
categorys <- c("pre-malignant neoplasm", "intestinal disease",
               "breast ductal carcinoma", "non-small cell lung carcinoma")
cnetplot(x2, showCategory = categorys)
# 'compareClusterResult' object is also supported.
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)
cnetplot(xx2)

## End(Not run)
```

---

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	<i>dotplot</i>
---------	----------------

---

**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,
  ...
)

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
)

```

### Arguments

<code>object</code>	compareClusterResult object
<code>...</code>	additional parameters
<code>x</code>	variable for x-axis, one of 'GeneRatio' and 'Count'
<code>color</code>	variable that used to color enriched terms, e.g. 'pvalue', 'p.adjust' or 'qvalue'
<code>showCategory</code>	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.
<code>size</code>	variable that used to scale the sizes of categories, one of "geneRatio", "Percentage" and "count"
<code>split</code>	ONTOLOGY or NULL
<code>font.size</code>	font size
<code>title</code>	figure title
<code>orderBy</code>	The order of the Y-axis
<code>label_format</code>	a numeric value sets wrap length, alternatively a custom function to format axis labels. by default wraps names longer than 30 characters
<code>by</code>	one of "geneRatio", "Percentage" and "count"
<code>includeAll</code>	logical
<code>decreasing</code>	logical. Should the orderBy order be increasing or decreasing?
<code>colorBy</code>	variable that used to color enriched terms, e.g. 'pvalue', 'p.adjust' or 'qvalue'
<code>group</code>	a logical value, whether to connect the nodes of the same group with wires.
<code>shape</code>	a logical value, whether to use nodes of different shapes to distinguish the group it belongs to



**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)
```

emapplot

*emapplot***Description**

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

**Usage**

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

## 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.enrichResult(
  x,
  showCategory = 30,
  layout = NULL,
  coords = NULL,
  color = "p.adjust",
  min_edge = 0.2,
  cex_label_category = 1,
  cex_category = 1,
  cex_line = 1,
  shadowtext = TRUE,
  label_style = "shadowtext",
  repel = FALSE,
  node_label = "category",
  with_edge = TRUE,
  group_category = FALSE,
  group_legend = FALSE,
  cex_label_group = 1,
  nWords = 4,
  label_format = 30,
  clusterFunction = stats::kmeans,
  nCluster = NULL,
  ...
)

emapplot.compareClusterResult(
  x,
  showCategory = 30,
  layout = NULL,
  coords = NULL,
  split = NULL,
  pie = "equal",
  legend_n = 5,
  cex_category = 1,
  cex_line = 1,
  min_edge = 0.2,
  cex_label_category = 1,
  shadowtext = TRUE,
  with_edge = TRUE,
  group_category = FALSE,
  label_format = 30,
```

```

    group_legend = FALSE,
    node_label = "category",
    label_style = "shadowtext",
    repel = FALSE,
    cex_label_group = 1,
    nWords = 4,
    clusterFunction = stats::kmeans,
    nCluster = NULL,
    cex_pie2axis = 1,
    ...
)

```

### Arguments

x	Enrichment result.
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.
...	additional parameters additional parameters can refer the following parameters. <ul style="list-style-type: none"> <li>• force Force of repulsion between overlapping text labels. Defaults to 1.</li> <li>• nudge_x, nudge_y Horizontal and vertical adjustments to nudge the starting position of each text label.</li> <li>• direction "both", "x", or "y" – direction in which to adjust position of labels.</li> <li>• ellipse_style style of ellipse, one of "ggforce" and "polygon".</li> <li>• ellipse_pro numeric indicating confidence value for the ellipses, it can be used only when ellipse_style = "polygon".</li> <li>• alpha the transparency of ellipse fill.</li> <li>• type The type of ellipse. The default "t" assumes a multivariate t-distribution, and "norm" assumes a multivariate normal distribution. "euclid" draws a circle with the radius equal to level, representing the euclidean distance from the center.</li> </ul>
layout	Layout of the map, e.g. 'star', 'circle', 'gem', 'dh', 'graphopt', 'grid', 'mds', 'randomly', 'fr', 'kk', 'drl' or 'lgl'.
coords	a data.frame with two columns: 'x' for X-axis coordinate and 'y' for Y-axis coordinate.
color	Variable that used to color enriched terms, e.g. 'pvalue', 'p.adjust' or 'qvalue'.
min_edge	The minimum similarity threshold for whether two nodes are connected, should be between 0 and 1, default value is 0.2.
cex_label_category	Scale of category node label size.
cex_category	Number indicating the amount by which plotting category nodes should be scaled relative to the default.
cex_line	Scale of line width
shadowtext	a logical value, whether to use shadow font.

label_style	style of group label, one of "shadowtext" and "ggforce".
repel	whether to correct the position of the label. Defaults to FALSE.
node_label	Select which labels to be displayed, one of 'category', 'group', 'all' and 'none'.
with_edge	Logical, if TRUE (the default), draw the edges of the network diagram.
group_category	a logical, if TRUE(the default), group the category.
group_legend	Logical, if TRUE, the grouping legend will be displayed. The default is FALSE.
cex_label_group	Numeric, scale of group labels size, the default value is 1.
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.
split	separate result by 'category' variable
pie	proportion of clusters in the pie chart, one of 'equal' (default) and 'Count'
legend_n	number of circle in legend
cex_pie2axis	It is used to adjust the relative size of the pie chart on the coordinate axis, the default value is 1.

## 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)
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 or a vector of terms.
emapplot(x2, showCategory = 10)
categorys <- c("pre-malignant neoplasm", "intestinal disease",
               "breast ductal carcinoma", "non-small cell lung carcinoma")
emapplot(x2, 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)
emapplot(xx2)

## End(Not run)

```

---

emapplot_cluster	<i>Functional grouping network diagram for enrichment result of over-representation test or gene set enrichment analysis</i>
------------------	--

---

## Description

This function has been replaced by ‘emapplot’.

## Usage

```
emapplot_cluster(x, ...)
```

## Arguments

x	enrichment result
...	additional parameters. Please refer to: <a href="#">emapplot</a> .

## Value

ggplot2 object

---

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
order	logical
drop	logical
...	additional parameter

**Value**

data.frame  
data.frame

**Author(s)**

Guangchuang Yu

---

ggtable	<i>ggtable</i>
---------	----------------

---

**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	<i>goplot</i>
--------	---------------

---

**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 = "sugiyama",
  geom = "text",
  ...
)

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

goplot.enrichResult(
  x,
  showCategory = 10,
  color = "p.adjust",
  layout = "sugiyama",
  geom = "text",
  ...
)

```

**Arguments**

x	enrichment result.
showCategory	number of enriched terms to display
color	variable that used to color enriched terms, e.g. pvalue, p.adjust or qvalue
layout	layout of the map
geom	label geom, one of 'label' or 'text'
...	additional parameter



**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)
gplot(yy)
gplot(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 <- gseD0(geneList)
gseaplot(x, geneSetID=1)
```

---

<code>gseaplot2</code>	<i>gseaplot2</i>
------------------------	------------------

---

**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**

<code>x</code>	gseaResult object
<code>geneSetID</code>	gene set ID
<code>title</code>	plot title
<code>color</code>	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 = "")
```

**Arguments**

x	gseaResult object
geneSetID	gene set ID
title	plot title

**Value**

ggplot object

**Author(s)**

Guangchuang Yu

---

gsInfo	<i>gsInfo</i>
--------	---------------

---

**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, foldChange = NULL, label_format = 30)

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

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

heatplot.enrichResult(
  x,
  showCategory = 30,
  foldChange = NULL,
  label_format = 30
)
```

**Arguments**

<code>x</code>	enrichment result.
<code>showCategory</code>	number of enriched terms to display
<code>foldChange</code>	fold Change.
<code>label_format</code>	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  
Guangchuang Yu

**Examples**

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

---

<code>pairwise_termsim</code>	<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

---

ridgeplot	<i>ridgeplot</i>
-----------	------------------

---

**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,
```

```

    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

<code>x</code>	<code>gseaResult</code> object
<code>showCategory</code>	number of categories for plotting
<code>fill</code>	one of "pvalue", "p.adjust", "qvalue"
<code>core_enrichment</code>	whether only using core_enriched genes
<code>label_format</code>	a numeric value sets wrap length, alternatively a custom function to format axis labels.
<code>...</code>	additional parameters by default wraps names longer than 30 characters
<code>orderBy</code>	The order of the Y-axis
<code>decreasing</code>	logical. Should the orderBy order be increasing or decreasing?

### Value

ggplot object

### Author(s)

Guangchuang Yu

### Examples

```

library(DOSE)
data(geneList)
x <- gseD0(geneList)
ridgeplot(x)
```

---

treeplot	<i>treeplot</i>
----------	-----------------

---

## Description

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

## Usage

```
treeplot(x, showCategory = 30, color = "p.adjust", label_format = 30, ...)
```

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

```
treeplot(x, showCategory = 30, color = "p.adjust", label_format = 30, ...)
```

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

```
treeplot(x, showCategory = 30, color = "p.adjust", label_format = 30, ...)
```

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

```
treeplot(x, showCategory = 5, color = "p.adjust", label_format = 30, ...)
```

```
treeplot.enrichResult(  
  x,  
  showCategory = 30,  
  color = "p.adjust",  
  nWords = 4,  
  nCluster = 5,  
  cex_category = 1,  
  label_format = 30,  
  fontsize = 4,  
  offset = 1,  
  offset_tiplab = 1,  
  hclust_method = "ward.D",  
  group_color = NULL,  
  extend = 0.3,  
  highlight = TRUE,  
  hexpand = 0.1,  
  align = "both",  
  ...  
)
```

```
treeplot.compareClusterResult(  
  x,  
  showCategory = 5,  
  color = "p.adjust",  
  nWords = 4,  
  nCluster = 5,
```

```

    cex_category = 1,
    split = NULL,
    label_format = 30,
    fontsize = 4,
    offset = 1,
    pie = "equal",
    legend_n = 3,
    offset_tiplab = 1,
    hclust_method = "ward.D",
    group_color = NULL,
    extend = 0.3,
    highlight = TRUE,
    geneClusterPanel = "heatMap",
    hexpand = 0.1,
    align = "both",
    ...
)

```

### Arguments

x	enrichment result.
showCategory	number of enriched terms to display
color	variable that used to color enriched terms, e.g. pvalue, p.adjust or qvalue
label_format	a numeric value sets wrap length, alternatively a custom function to format axis labels.
...	additional parameters
nWords	The number of words in the cluster tags.
nCluster	The number of clusters, the default value is 5.
cex_category	Number indicating the amount by which plotting category. nodes should be scaled relative to the default. displayed completely, the user can increase this value.
fontsize	The size of text, default is 4.
offset	numeric, distance bar and tree, offset of bar and text from the clade, default is 1, meaning $1 * 1.2 * x\_range\_of\_tree$ plus distance_between_tree_and_tiplab ( $1 * (1.2 * x\_range\_of\_tree + distance\_between\_tree\_and\_tiplab)$ ).
offset_tiplab	tiplab offset, the bigger the number, the farther the distance between the node and the branch. The default is 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 $1 * 0.09 * column\_number\_of\_dotplot * x\_range\_of\_tree$ .
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).
group_color	A vector of group colors, the length of the vector should be the same as nCluster.
extend	Numeric, extend the length of bar, default is 0.3.

highlight	Logical value, if TRUE(default), add ggtree::geom_highlight() layer.
hexexpand	expand x limits by amount of xrange * hexexpand.
align	control the align direction of the edge of high light rectangular. Options is 'none', 'left', 'right', 'both' (default).
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'.
legend_n	Number of circle in legend, the default value is 3.
geneClusterPanel	one of "heatMap"(default), "dotplot", "pie".

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

# Index

barplot.enrichResult, 2

cnetplot, 3

cnetplot,compareClusterResult-method  
(cnetplot), 3

cnetplot,enrichResult-method  
(cnetplot), 3

cnetplot,gseaResult-method (cnetplot), 3

cnetplot,list-method (cnetplot), 3

cnetplot.enrichResult (cnetplot), 3

color\_palette, 6

dotplot, 6

dotplot,compareClusterResult,ANY-method  
(dotplot), 6

dotplot,compareClusterResult-method  
(dotplot), 6

dotplot,enrichResult-method (dotplot), 6

dotplot,gseaResult-method (dotplot), 6

dotplot.compareClusterResult (dotplot),  
6

dotplot.enrichResult (dotplot), 6

emapplot, 9, 13

emapplot,compareClusterResult-method  
(emapplot), 9

emapplot,enrichResult-method  
(emapplot), 9

emapplot,gseaResult-method (emapplot), 9

emapplot.compareClusterResult  
(emapplot), 9

emapplot.enrichResult (emapplot), 9

emapplot\_cluster, 13

fortify.compareClusterResult, 14

fortify.enrichResult  
(fortify.compareClusterResult),  
14

ggtable, 15

godata, 23

goplot, 15

goplot,enrichResult-method (goplot), 15

goplot,gseaResult-method (goplot), 15

goplot.enrichResult (goplot), 15

gseadist, 17

gseaplot, 18

gseaplot,gseaResult-method (gseaplot),  
18

gseaplot.gseaResult (gseaplot), 18

gseaplot2, 19

gsearank, 20

gsInfo, 21

heatplot, 21

heatplot,enrichResult-method  
(heatplot), 21

heatplot,gseaResult-method (heatplot),  
21

heatplot.enrichResult (heatplot), 21

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

plotting.clusterProfile, 24

pmcplot, 25

ridgeplot, 25

ridgeplot,gseaResult-method  
(ridgeplot), 25

ridgeplot.gseaResult (ridgeplot), 25

treeplot, 27



treeplot,compareClusterResult-method  
    (treeplot), [27](#)  
treeplot,enrichResult-method  
    (treeplot), [27](#)  
treeplot,gseaResult-method (treeplot),  
    [27](#)  
treeplot.compareClusterResult  
    (treeplot), [27](#)  
treeplot.enrichResult (treeplot), [27](#)  
  
upsetplot, [30](#)  
upsetplot,enrichResult,ANY-method  
    (upsetplot), [30](#)  
upsetplot,enrichResult-method  
    (upsetplot), [30](#)  
upsetplot,gseaResult (upsetplot), [30](#)  
upsetplot,gseaResult-method  
    (upsetplot), [30](#)