Package 'ggtangle'

May 8, 2026

Title Draw Network with Data

Version 0.1.2

Description Extends the 'ggplot2' plotting system to support network visualization. Inspired by the 'Method 1' in 'ggtree' (G Yu (2018) <doi:10.1093/molbev/msy194>), 'ggtangle' is designed to work with network associated data.

Depends R (>= 4.1.0)

Imports ggfun (>= 0.1.7), ggplot2, ggrepel, igraph, rlang, yulab.utils (>= 0.2.2)

Suggests aplot, cli, ggiraph, ggnewscale, ggtree, quarto, scatterpie (>= 0.2.4)

VignetteBuilder quarto

License Artistic-2.0

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

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

Maintainer Guangchuang Yu <guangchuangyu@gmail.com>

Repository CRAN

Date/Publication 2026-04-22 14:50:02 UTC

Contents

---

cnetplot *category-item network plot*

Description

category-item network plot

Usage

```
cnetplot(
  x,
  layout = igraph::layout_nicely,
  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 'list'
cnetplot(
  x,
  layout = "nicely",
  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,
  ...
)
```

Arguments

x	input object
layout	network layout
showCategory	categories to display. Use a single number to show the first n categories, a numeric vector to select categories by index (for example, c(1, 10, 53)), or a character vector to select categories by name.
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, e.g., "black". If color = "category", then edges will be colored based on the category information.
size_edge	relative size of edge
categorySizeBy	An expression (e.g., itemNum, p.adjust) or a formula (e.g., ~ -log10(p.adjust)) to set the category node size.
node_label	one of 'all', 'none', 'category', 'item', 'exclusive' or 'share'
foldChange	numeric values to color the item (e.g, foldChange of gene expression values)
fc_threshold	threshold for absolute fold change to filter items
hilight	selected category to be highlighted
hilight_alpha	transparent value for not selected to be highlight
...	additional parameters. One important parameter is 'curvature' (default is 0), which can be used to curve the edges (e.g., curvature = 0.2).

Details

For list input, showCategory accepts either a single integer for the first n categories, a numeric vector of category indices, or a character vector of category names.

Examples

```
x <- list(A = letters[1:10], B = letters[5:12])
attr(x, "p.adjust") <- c(A = 0.01, B = 0.2)
p <- cnetplot(x, node_label = "none", categorySizeBy = ~ -log10(p.adjust))
p <- cnetplot(x, showCategory = c(1, 2))
```

drag_network *Drag the nodes of a network to update the layout of the network*

Description

Drag the nodes of a network to update the layout of the network

Usage

```
drag_network(p, g = NULL)
```

Arguments

p	the network diagram as a ggplot/gg/ggraph object.
g	an corresponding igraph object. Default is to extract from the 'ggraph' attribute.

Value

an updated ggplot/gg/ggraph object

Examples

```
## Not run:
library(igraph)
library(ggraph)

flow_info <- data.frame(from = c(1,2,3,3,4,5,6),
                        to = c(5,5,5,6,7,6,7))
g = graph_from_data_frame(flow_info)
p <- ggraph(g, layout='nicely') + geom_node_point() + geom_edge_link()
pp <- drag_network(p)

## End(Not run)
```

geom_cnet_label *geom_cnet_label*

Description

add labels of cnetplot

Usage

```
geom_cnet_label(mapping = NULL, data = NULL, node_label = "all", ...)
```

Arguments

mapping	aes mapping, default is NULL
data	plot data, default is NULL
node_label	which type of node label to be displayed, see also cnetplot
...	parameters that passed to geom_text_repel

Author(s)

Guangchuang Yu

---

geom_edge *layer to draw edges of a network*

Description

layer to draw edges of a network

Usage

```
geom_edge(mapping = NULL, data = NULL, geom = geom_segment, ...)
```

Arguments

mapping	aesthetic mapping, default is NULL
data	data to plot, default is NULL
geom	geometric layer to draw lines
...	additional parameter passed to 'geom'

Value

line segments layer

Examples

```
flow_info <- data.frame(from = LETTERS[c(1,2,3,3,4,5,6)],
                        to = LETTERS[c(5,5,5,6,7,6,7)])

dd <- data.frame(
    label = LETTERS[1:7],
    v1 = abs(rnorm(7)),
    v2 = abs(rnorm(7)),
    v3 = abs(rnorm(7))
)

g = igraph::graph_from_data_frame(flow_info)
```

```
p <- ggplot(g)  + geom_edge()
library(ggplot2)
library(scatterpie)

p %<+% dd +
    geom_scatterpie(cols = c("v1", "v2", "v3")) +
    geom_text(aes(label=label), nudge_y = .2) +
    coord_fixed()
```

`geom_edge_interactive` *layer to draw edges of a network interactively*

Description

layer to draw edges of a network interactively

Usage

```
geom_edge_interactive(
  mapping = NULL,
  data = NULL,
  geom = ggiraph::geom_segment_interactive,
  ...
)
```

Arguments

`mapping`	aesthetic mapping, default is NULL
`data`	data to plot, default is NULL
`geom`	geometric layer to draw lines
`...`	additional parameter passed to 'geom'

Value

line segments layer

Examples

```
library(ggiraph)
flow_info <- data.frame(from = LETTERS[c(1,2,3,3,4,5,6)],
                        to = LETTERS[c(5,5,5,6,7,6,7)])

dd <- data.frame(
    label = LETTERS[1:7],
    v1 = abs(rnorm(7)),
    v2 = abs(rnorm(7)),
    v3 = abs(rnorm(7))
```

```
)

g = igraph::graph_from_data_frame(flow_info)

p <- ggplot(g)  + geom_edge_interactive()
library(ggplot2)
library(scatterpie)

p2 <- p %<+% dd +
    geom_scatterpie(cols = c("v1", "v2", "v3")) +
   geom_text_interactive(aes(label=label, tooltip = label, data_id = label), nudge_y = .2) +
    coord_fixed()

girafe(ggobj = p2)
```

`geom_edge_text` *layer to draw edge labels of a network*

Description

layer to draw edge labels of a network

Usage

```
geom_edge_text(
  mapping = NULL,
  data = NULL,
  geom = geom_text,
  angle_calc = "none",
  label_dodge = NULL,
  ...
)
```

Arguments

`mapping`	aesthetic mapping, default is NULL
`data`	data to plot, default is NULL
`geom`	geometric layer to draw text, default is geom_text
`angle_calc`	how to calculate angle ('along' or 'none')
`label_dodge`	dodge distance
`...`	additional parameter passed to 'geom'

Value

text layer

geom_edge_text_interactive *layer to draw edge labels of a network interactively*

Description

layer to draw edge labels of a network interactively

Usage

```
geom_edge_text_interactive(
  mapping = NULL,
  data = NULL,
  geom = ggiraph::geom_text_interactive,
  angle_calc = "none",
  label_dodge = NULL,
  ...
)
```

Arguments

mapping	aesthetic mapping, default is NULL
data	data to plot, default is NULL
geom	geometric layer to draw text, default is geom_text
angle_calc	how to calculate angle ('along' or 'none')
label_dodge	dodge distance
...	additional parameter passed to 'geom'

Value

text layer

layout_circular *Circular layout*

Description

Circular layout

Usage

```
layout_circular(graph, sort.by = NULL, ...)
```

Arguments

graph	A graph object.
sort.by	The attribute to sort the nodes by. Default is NULL.
...	Additional arguments passed to igraph::layout_in_circle.

Value

A matrix of coordinates.

Author(s)

Guangchuang Yu

---

layout_fishbone *Glycan Layout Algorithm (Fishbone/SNFG-like)*

Description

Calculates node coordinates for a glycan structure to mimic SNFG style. Supports customizable direction and branch length.

Usage

```
layout_fishbone(graph, direction = "left", length = 1, angle_sep = 30, ...)
```

Arguments

graph	An igraph object.
direction	The direction of the main chain growth ("left", "right", "up", "down"). Default is "left".
length	The distance between nodes. Default is 1.
angle_sep	The angle separation for branches in degrees. Default is 30.
...	Additional arguments.

Value

A matrix of x, y coordinates.

`layout_linear` *Linear layout*

Description

Linear layout

Usage

```
layout_linear(graph, sort.by = NULL, ...)
```

Arguments

`graph`	A graph object.
`sort.by`	The attribute to sort the nodes by. Default is NULL.
`...`	Additional arguments.

Value

A matrix of coordinates.

Author(s)

Guangchuang Yu

Index