

Package ‘ribiosGraph’

May 9, 2026

Type Package

Title Manipulate and Visualize Graphs in the 'ribios' Software Suite

Version 1.1.0

Date 2026-01-24

Description Tools to manipulate and visualize graphs (networks) for computational biology in drug discovery, for instance functions for creating bipartite graphs and for interactive visualizations. Zhang (2025) <<https://github.com/bedapub/ribiosGraph>>.

Depends R (>= 3.4.0), igraph

Imports magrittr, plotly, ribiosUtils

Suggests testthat

License GPL-3

URL <https://github.com/bedapub/ribiosGraph>

BugReports <https://github.com/bedapub/ribiosGraph/issues>

Encoding UTF-8

RoxygenNote 7.3.3

NeedsCompilation no

Author Jitao David Zhang [aut, cre] (ORCID:
<<https://orcid.org/0000-0002-3085-0909>>)

Maintainer Jitao David Zhang <jitao_david.zhang@roche.com>

Repository CRAN

Date/Publication 2026-02-20 10:40:14 UTC

Contents

bipartite_graph_from_data_frame	2
exportGML	3
incidence2bipartite	4
layout_as_bipartiteLR	5
list2incidenceMatrix	5
plotlyBipartiteGraph	6

`bipartite_graph_from_data_frame`*Create a bipartite graph from data frame*

Description

Create a bipartite graph from data frame

Usage

```
bipartite_graph_from_data_frame(x)
```

Arguments

`x` A data.frame with at least two columns.

Value

A `igraph` object that represents a bipartite graph. The `type` attribute of vertices is a logical vector that represents the two classes of nodes: nodes in the first column in the input data.frame are of the type `TRUE`, and those in the second column are of the type `FALSE`.

Extra columns besides the first two are used as edge attributes. See example below.

Examples

```
myDataFrame <- data.frame(Alpha=c("A", "E", "O", "U", "B", "D"),
  Type=c("Vowel", "Vowel", "Vowel", "Vowel", "Consonance", "Consonance"))
myBpGraph <- bipartite_graph_from_data_frame(myDataFrame)

myDataFrame2 <- data.frame(Alpha=c("A", "E", "O", "U", "B", "D"),
  Type=c("Vowel", "Vowel", "Vowel", "Vowel", "Consonance", "Consonance"),
  Example=c("BAT", "BED", "BOT", "BUT", "DUB", "DUB"))
myBpGraph2 <- bipartite_graph_from_data_frame(myDataFrame2)
igraph::E(myBpGraph2)$Example
```

`exportGML`*Export igraph object to GML, friendly to Cytoscape and yEd*

Description

`exportGML` exports an `igraph` object to GML files complying with specifications defined by Cytoscape and `yEd`. Compared to the native `write.graph` function provided by the `igraph` package, GML files exported with `exportGML` can be directly read and properly visualized by Cytoscape and `yEd`.

Currently the function uses supports following attributes: Node name: `V(igraph)$name` Node label: `V(igraph)$label` Node isInput: `V(igraph)$isInput`, controlling node shapes Edge label: `E(igraph)$label`, determining edge target arrow

So far the function is mainly used by the `ronet.Rscript` script in the package. Users are invited to adapt the function for other purposes.

Usage

```
exportGML(igraph, filename)
```

Arguments

<code>igraph</code>	An <code>igraph</code> object
<code>filename</code>	Filename

Value

Invisible NULL

Author(s)

Jitao David Zhang, <jitao_david.zhang@roche.com>

See Also

[write.graph](#)

Examples

```
g <- barabasi.game(100, directed=FALSE)
V(g)$label <- c(paste("node", 1:99, sep=""), "--")
V(g)$name <- 1:100
V(g)$isInput <- rbinom(100,1, 0.5)
E(g)$label <- "Expression"
gPosE <- as.logical(rbinom(ecount(g), 1, 0.25))
gNegE <- as.logical(rbinom(ecount(g), 1, 0.25))
E(g)$label[gPosE] <- "Expressoion_Positive"
E(g)$label[gNegE] <- "Expressoion_Negative"
gFile <- tempfile()
```

```
exportGML(g, gFile)
```

```
incidence2bipartite Build a bipartite graph with an incidence matrix
```

Description

Build a bipartite graph with an incidence matrix

Usage

```
incidence2bipartite(
  matrix,
  size = c(12, 9),
  color = c("orange", "lightblue"),
  label.cex = c(1.1, 0.95),
  label.color = c("black", "navyblue"),
  V = list(),
  E = list(color = "black")
)
```

Arguments

matrix	An incidence matrix
size	A vector of length 2, size of nodes in rows and in columns
color	A vector of length 2, color of nodes in rows and in columns
label.cex	A vector of length 2, font size of labels of nodes in rows and in columns
label.color	A vector of length 2, color of labels of nodes in rows and in columns
V	A named list of other node styles, each item of length 1 or 2. In the latter case, the first value is used for nodes in rows and the second for nodes in columns
E	A named list of edge styles. Each item must be length of 1.

Value

An instance of igraph graph

Examples

```
myIncMat <- matrix(c(0, 0, 1,
  0, 1, 0,
  1, 0, 0,
  0, 1, 1,
  1, 1, 1),
  ncol=3, byrow=TRUE, dimnames=list(LETTERS[1:5], letters[1:3]))
myGraph <- incidence2bipartite(myIncMat,
```

```
size=c(18,12),
V=list(shape=c("rectangle", "circle"),
       frame.color="lightgray"))
if(requireNamespace("igraph")) {
  igraph::plot.igraph(myGraph)
}
```

layout_as_bipartiteLR *Layout a bipartite graph from left to right*

Description

Layout a bipartite graph from left to right

Usage

```
layout_as_bipartiteLR(g)
```

Arguments

`g` A igraph object

Value

A two-column matrix, the layout of the graph

The function simply calls [layout_as_bipartite](#) and reverses the X and Y coordinates.

Examples

```
myDataFrame <- data.frame(Alpha=c("A", "E", "O", "U", "B", "D"),
                          Type=c("Vowel", "Vowel", "Vowel", "Vowel", "Consonance", "Consonance"))
myBpGraph <- bipartite_graph_from_data_frame(myDataFrame)
myLayout <- layout_as_bipartiteLR(myBpGraph)
```

list2incidenceMatrix *Convert a list of character strings into an incidence matrix*

Description

Convert a list of character strings into an incidence matrix

Usage

```
list2incidenceMatrix(list, type = c("binary", "count"))
```

Arguments

<code>list</code>	A list of character strings, can be unique or redundant
<code>type</code>	How the values of the incidence matrix will be filled, see details.

Value

An incidence matrix, containing either binary (TRUE/FALSE) or integer values.

Type 'binary' will produce a logical matrix, whereas 'count' will produce a matrix where the frequency of the character strings in the list.

Author(s)

Jitao David Zhang, <jitao_david.zhang@roche.com>

Examples

```
wordList <- list("2006"=c("HSV", "BVB", "FCB"),
  "2007"=c("BVB", "VFB", "STP"),
  "2008"=c("VFL", "BVB", "HSV"))
list2incidenceMatrix(wordList, type="binary")

letterList <- list("First"=c("A", "a", "A", "a"), "Second"=c("B", "b", "A"))
list2incidenceMatrix(letterList, type="count")
list2incidenceMatrix(letterList, type="binary")
```

plotlyBipartiteGraph *Plot a bipartite graph using plot_ly*

Description

Plot a bipartite graph using plot_ly

Usage

```
plotlyBipartiteGraph(
  g,
  layout = layout_as_bipartiteLR(g),
  edge.line = list(color = "#030303", width = 0.3),
  axis = list(title = "", showgrid = FALSE, showticklabels = FALSE, zeroline = FALSE),
  title = ""
)
```

Arguments

<code>g</code>	A igraph object of a bipartite graph
<code>layout</code>	The layout, the LR layout is used by default
<code>edge.line</code>	List, specifying edge lines
<code>axis</code>	List, specifying axes
<code>title</code>	Character string, plot title

Value

A plotly and htmlwidget object

If the layout is left-right, the function takes care of the alignment of labels

Examples

```
myDataFrame <- data.frame(word=c("ja", "nein", "yes", "no", "stark", "stark"),
  language=c("German", "German", "English", "English", "English", "German"))
myBpGraph <- bipartite_graph_from_data_frame(myDataFrame)
plotlyBipartiteGraph(myBpGraph)
```

Index

`bipartite_graph_from_data_frame`, [2](#)

`exportGML`, [3](#)

`incidence2bipartite`, [4](#)

`layout_as_bipartite`, [5](#)

`layout_as_bipartiteLR`, [5](#)

`list2incidenceMatrix`, [5](#)

`plotlyBipartiteGraph`, [6](#)

`write.graph`, [3](#)