Package ‘graphsim’

March 16, 2020

Type Package
Title Simulate Expression Data from ‘igraph’ Networks
Version 0.1.2
Date 2020-03-02
Description Functions to develop simulated continuous data (e.g., gene expression) from a sigma covariance matrix derived from a graph structure in ‘igraph’ objects. Intended to extend 'mvtnorm' to take 'igraph' structures rather than sigma matrices as input.

Language en-GB
License GPL-3
Depends R (>= 2.10.0)
Imports utils, igraph, mvtnorm, Matrix, matrixcalc, graphics
Suggests knitr, rmarkdown, animation, testthat, gplots, scales, devtools, vdiffr
LazyData TRUE
RoxygenNote 7.0.2
Encoding UTF-8
VignetteBuilder knitr
NeedsCompilation no
Author S. Thomas Kelly [ctb],
S. Thomas Kelly [aut, cre],
Michael A. Black [aut, ths]
Maintainer S. Thomas Kelly <tom.kelly@riken.jp>
Repository CRAN
Date/Publication 2020-03-16 14:10:02 UTC

R topics documented:

  generate_expression ........................................... 2
  make_adjmatrix ................................................. 3
  make_commonlink ............................................... 4
**Description**

Compute simulated continuous expression data from a graph network structure. Requires an igraph pathway structure and a matrix of states (1 for activating and -1 for inhibiting) for link signed correlations, from a vector of edge states to a signed adjacency matrix for use in `generate_expression`. Uses graph structure to pass a sigma covariance matrix from `make_sigma_mat_dist_graph` or `make_sigma_mat_graph` on to `rmvnorm`.

**Usage**

```r
generate_expression(
  n,
  graph,
  state = NULL,
  cor = 0.8,
  mean = 0,
  comm = FALSE,
  dist = FALSE,
  absolute = FALSE
)
```

**Arguments**

- `n` number of observations (simulated samples).
- `graph` An igraph object. May must be directed if states are used.
- `state` numeric vector. Vector of length E(graph). Sign used to calculate state matrix, may be an integer state or inferred directly from expected correlations for each edge. May be applied a scalar across all edges or as a vector for each edge respectively. May also be entered as text for "activating" or "inhibiting" or as integers for activating (0,1) or inhibiting (-1,2). Compatible with inputs for `plot_directed`. Also takes a pre-computed state matrix from `make_state_matrix` if applied to the same graph multiple times.
**make_adjmatrix**

Generate Adjacency Matrix

**Description**

Compute the adjacency matrix of a (directed) *igraph* structure, preserving node/column/row names (and direction).

**Usage**

`make_adjmatrix_graph(graph, directed = FALSE)`

**Arguments**

- **graph**: An *igraph* object. May be directed or weighted.
- **directed**: logical. Whether directed information is passed to the adjacency matrix.

**Value**

An adjacency matrix compatible with generating an expression matrix.

**cor**

numeric. Simulated maximum correlation/covariance of two adjacent nodes. Default to 0.8.

**mean**

mean value of each simulated gene. Defaults to 0. May be entered as a scalar applying to all genes or a vector with a separate value for each.

**comm, absolute**

logical. Parameters for Sigma matrix generation. Passed on to `make_sigma_mat_dist_graph` or `make_sigma_mat_graph`.

**dist**

logical. Whether a graph distance (`make_sigma_mat_dist_graph`) or derived matrix (`make_sigma_mat_graph`) is used to compute the sigma matrix.

**Value**

numeric matrix of simulated data (log-normalised counts)

**Examples**

```r
library("igraph")
graph_test_edges <- rbind(c("A", "B"), c("B", "C"), c("B", "D"))
graph_test <- graph.edgelist(graph_test_edges, directed = TRUE)
adjacency_matrix <- make_adjmatrix_graph(graph_test)
n <- 100
generate_expression(n, graph_test, cor = 0.8)
```
### Examples

```r
library("igraph")
graph_test_edges <- rbind(c("A", "B"), c("B", "C"), c("B", "D"))
graph_test <- graph.edgelist(graph_test_edges, directed = TRUE)
adjacency_matrix <- make_adjmatrix_graph(graph_test)

common_link_matrix <- make_commonlink_adjmat(adjacency_matrix)
```

---

#### make_commonlink  
Generate Common Link Matrix

**Description**

Compute the common link matrix of a (directed) `igraph` structure, preserving node / column / row names (and direction).

**Usage**

```r
make_commonlink_adjmat(adj_mat)
make_commonlink_graph(graph, directed = FALSE)
```

**Arguments**

- `adj_mat`: precomputed adjacency matrix.
- `graph`: An `igraph` object. May be directed or weighted.
- `directed`: logical. Whether directed information is passed to the adjacency matrix.

**Value**

An integer matrix of number of links shared between nodes

**Examples**

```r
library("igraph")
graph_test_edges <- rbind(c("A", "B"), c("B", "C"), c("B", "D"))
graph_test <- graph.edgelist(graph_test_edges, directed = TRUE)
adjacency_matrix <- make_adjmatrix_graph(graph_test)
common_link_matrix <- make_commonlink_adjmat(adjacency_matrix)
```
**make_distance_graph**

*Generate Distance Matrix*

**Description**

Compute the distance matrix of using shortest paths of a (directed) **igraph** structure, normalising by the diameter of the network, preserving node/column/row names (and direction).

**Usage**

```r
make_distance_graph(graph, directed = TRUE, absolute = FALSE)
make_distance_adjmat(mat, directed = TRUE, absolute = FALSE)
```

**Arguments**

- `graph` An **igraph** object. May be directed or weighted.
- `directed` logical. Whether directed information is passed to the distance matrix.
- `absolute` logical. Whether distances are scaled as the absolute difference from the diameter (maximum possible). Defaults to TRUE. The alternative is to calculate a relative difference from the diameter for a geometric decay in distance.
- `mat` precomputed adjacency or commonlink matrix.

**Value**

A numeric matrix of values in the range [0, 1] where lower values are closer

**Examples**

```r
library("igraph")
graph_test_edges <- rbind(c("A", "B"), c("B", "C"), c("B", "D"))
graph_test <- graph.edgelist(graph_test_edges, directed = TRUE)
adjacency_matrix <- make_adjmatrix_graph(graph_test)
distance_matrix <- make_distance_adjmat(adjacency_matrix)
```

**make_sigma**

*Generate Sigma Matrix*

**Description**

Compute the Sigma matrix from an **igraph** structure or pre-computed matrix. These are compatible with `rmvnorm` and `generate_expression`.

```r
```

**Examples**

```r
library("igraph")
graph_test_edges <- rbind(c("A", "B"), c("B", "C"), c("B", "D"))
graph_test <- graph.edgelist(graph_test_edges, directed = TRUE)
adjacency_matrix <- make_adjmatrix_graph(graph_test)
distance_matrix <- make_distance_adjmat(adjacency_matrix)
```
Usage

make_sigma_mat_adjmat(mat, cor)

make_sigma_mat_graph(graph, cor, comm = FALSE, directed = FALSE)

make_sigma_mat_dist_adjmat(mat, cor, absolute = FALSE)

make_sigma_mat_dist_graph(graph, cor, absolute = FALSE)

Arguments

- **mat**: precomputed adjacency, commonlink, or scaled distance matrix.
- **cor**: numeric. Simulated maximum correlation/covariance of two adjacent nodes. Default to 0.8.
- **graph**: An igraph object. May be directed or weighted.
- **comm**: logical whether a common link matrix is used to compute sigma. Defaults to FALSE (adjacency matrix).
- **directed**: logical. Whether directed information is passed to the distance matrix.
- **absolute**: logical. Whether distances are scaled as the absolute difference from the diameter (maximum possible). Defaults to TRUE. The alternative is to calculate a relative difference from the diameter for a geometric decay in distance.

Value

a numeric covariance matrix of values in the range [-1, 1]

Examples

```r
library("igraph")

graph_test_edges <- rbind(c("A", "B"), c("B", "C"), c("B", "D"))

graph_test <- graph.edgelist(graph_test_edges, directed = TRUE)

adjacency_matrix <- make_adjmatrix_graph(graph_test)

sigma_matrix <- make_sigma_mat_adjmat(adjacency_matrix, cor = 0.8)
```

Description

Functions to compute the matrix of states (1 for activating and -1 for inhibiting) for link signed correlations, from a vector of edge states to a signed adjacency matrix for use in generate_expression.

Usage

make_state_matrix(graph, state = "activating")
**Arguments**

- **graph**: An igraph object. May be directed or weighted as long as a shortest path can be computed.
- **state**: numeric vector. Vector of length E(graph). Sign used to calculate state matrix, may be an integer state or inferred directly from expected correlations for each edge. May be applied a scalar across all edges or as a vector for each edge respectively. May also be entered as text for “activating” or “inhibiting” or as integers for activating (0,1) or inhibiting (-1,2). Compatible with inputs for `plot_directed`.

**Value**

An integer matrix indicating the resolved state (activating or inhibiting for each edge or path between nodes)

**Examples**

```r
library("igraph")
graph_test_edges <- rbind(c("A", "B"), c("B", "C"), c("B", "D"))
graph_test <- graph.edgelist(graph_test_edges, directed = TRUE)
state_matrix <- make_state_matrix(graph_test)
```

---

**Description**

Reactome pathway R-HSA-198203 for the interactions in the phosphoinositide-3-kinase activation of Protein kinase B (PKB), also known as Akt

**Usage**

Pi3K_AKT_graph

**Format**

A graph object of 275 vertices and 21106 edges:

- **V**: gene symbol (human)
- **E**: directed relationship for pathway
- **state**: type of relationship (activating or inhibiting) as edge attribute

**Source**

PathwayCommons [https://reactome.org/content/detail/R-HSA-198203](https://reactome.org/content/detail/R-HSA-198203)
### Pi3K_graph  
**PI3K Cascade**

#### Description
Reactome pathway R-HSA-109704 for the interactions in the phosphoinositide-3-kinase cascade

#### Usage
Pi3K_graph

#### Format
A graph object of 35 vertices and 251 edges:
- **V**  gene symbol (human)
- **E**  directed relationship for pathway
- **state**  type of relationship (activating or inhibiting) as edge attribute

#### Source
PathwayCommons  [https://reactome.org/content/detail/R-HSA-109704](https://reactome.org/content/detail/R-HSA-109704)

---

### plot_directed  
**Extensions to iGraph for Customising plots**

#### Description
Functions to plot_directed or graph structures including customised colours, layout, states, arrows. Uses graphs functions as an extension of igraph. Designed for plotting directed graphs.

#### Usage
plot_directed(
    graph,  
    state = NULL,  
    labels = NULL,  
    layout = layout.fruchterman.reingold,  
    cex.node = 1,  
    cex.label = 0.75,  
    cex.arrow = 1.25,  
    cex.main = 0.8,  
    arrow_clip = 0.075,  
    pch = 21,  
    border.node = "grey33",  
)
plot_directed

```r
fill.node = "grey66",
col.label = NULL,
col.arrow = NULL,
main = NULL,
sub = NULL,
xlab = "",
ylab = "",
frame.plot = F
```

**Arguments**

- `graph` An igraph object. Must be directed with known states.
- `state` character or integer. Defaults to "activating" if no "state" edge attribute found. May be applied a scalar across all edges or as a vector for each edge respectively. Accepts non-integer values for weighted edges provided that the sign indicates whether links are activating (positive) or inhibiting (negative). May also be entered as text for "activating" or "inhibiting" or as integers for activating (0,1) or inhibiting (-1,2). Compatible with inputs for make_state_matrix or generate_expression_graph in the graphsim package [https://github.com/TomKellyGenetics/graphsim](https://github.com/TomKellyGenetics/graphsim).
- `labels` character vector. For labels to plot nodes. Defaults to vertex names in graph object. Entering "" would yield unlabelled nodes.
- `layout` function. Layout function as selected from layout_. Defaults to layout.fruchterman.reingold. Alternatives include layout.kamada.kawai, layout.reingold.tilford, layout.sugiyama, and layout.davidson.harel. A 2-column layout matrix giving x and y co-ordinates of each node can be given.
- `cex.node` numeric. Defaults to 1.
- `cex.label` numeric. Defaults to 0.75.
- `cex.arrow` numeric Defaults to 1.25. May take a scalar applied to all edges or a vector with values for each edge respectively.
- `cex.main` numeric. Defaults to 0.8.
- `arrow_clip` numeric Defaults to 0.075 (7.5%).
- `pch` parameter passed to plot. Defaults to 21. Recommends using selecting between 21-25 to preserve colour behaviour. Otherwise entire node will inherit border.node as it’s colour, in which case a light colour is recommended to see labels.
- `border.node` character. Specifies the colours of node border passed to plot. Defaults to grey33. Applies to whole node shape if pch has only one colour.
- `fill.node` character. Specifies the colours of node fill passed to plot. Defaults to grey66.
- `col.label` character. Specifies the colours of node labels passed to plot. Defaults to par("fg").
- `col.arrow` character. Specifies the colours of arrows passed to plot. Defaults to par("fg"). May take a scalar applied to all edges or a vector with colours for each edge respectively.
main, sub, xlab, ylab

Plotting parameters to specify plot titles or axes labels

frame.plot logical. Whether to frame plot with a box. Defaults to FALSE.

Value

base R graphics

Examples

#generate example graphs
library("igraph")

graph_test4_edges <- rbind(c("A", "C"), c("B", "C"), c("C", "D"), c("D", "E"),
                          c("D", "F"), c("F", "G"), c("F", "I"), c("H", "I"))

graph_test4 <- graph.edgelist(graph_test4_edges, directed = TRUE)

#plots with igraph defaults
plot(graph_test4, layout = layout.fruchterman.reingold)
plot(graph_test4, layout = layout.kamada.kawai)

#plots with scalar states
plot_directed(graph_test4, state="activating")
plot_directed(graph_test4, state="inhibiting")

#plots with vector states
plot_directed(graph_test4, state=c(1, 1, 1, 1, -1, 1, 1, 1))
plot_directed(graph_test4, state=c(1, 1, -1, 1, -1, 1, -1, 1))

#plots states with graph attributes
E(graph_test4)$state <- 1
plot_directed(graph_test4)
E(graph_test4)$state <- c(1, 1, -1, 1, -1, 1, -1, 1)
plot_directed(graph_test4)

#plot layout customised
plot_directed(graph_test4, state=c(1, 1, -1, 1, -1, 1, -1, 1), layout = layout.kamada.kawai)

---

RAF_MAP_graph # 'RAF/MAP kinase cascade

Description

Reactome pathway R-HSA-5673001 for the interactions in the RAF/MAP kinase cascade

Usage

RAF_MAP_graph
TGFBeta_Smad_graph

Format
A graph object of 17 vertices and 121 edges:
V  gene symbol (human)
E  directed relationship for pathway

Source
PathwayCommons https://reactome.org/content/detail/R-HSA-5673001

TGFBeta_Smad_graph  TGF-β receptor signaling activates SMADs

Description
Reactome pathway R-HSA-2173789 for the interactions in the TGF-β receptor signaling activates SMADs

Usage
TGFBeta_Smad_graph

Format
A graph object of 32 vertices and 173 edges:
V  gene symbol (human)
E  directed relationship for pathway
state  type of relationship (activating or inhibiting) as edge attribute

Source
PathwayCommons https://reactome.org/content/detail/R-HSA-2173789
Index

*Topic adjacency
  make_adjmatrix, 3
  make_distance_graph, 5
*Topic datasets
  Pi3K_AKT_graph, 7
  Pi3K_graph, 8
  RAF_MAP_graph, 10
  TGFBeta_Smad_graph, 11
*Topic graph
  generate_expression, 2
  make_adjmatrix, 3
  make_commonlink, 4
  make_distance_graph, 5
  make_sigma, 5
  make_state_matrix, 6
  plot_directed, 8
*Topic igraph
  generate_expression, 2
  make_adjmatrix, 3
  make_commonlink, 4
  make_distance_graph, 5
  make_sigma, 5
  make_state_matrix, 6
  plot_directed, 8
*Topic mvtnorm
  generate_expression, 2
  make_sigma, 5
  make_state_matrix, 6
*Topic neighbourhood
  make_commonlink, 4
*Topic network
  generate_expression, 2
  make_adjmatrix, 3
  make_commonlink, 4
  make_distance_graph, 5
  make_sigma, 5
  make_state_matrix, 6
*Topic plot
  plot_directed, 8
*Topic simulation
  generate_expression, 2
  make_state_matrix, 6
  generate_expression, 2, 2, 5, 6
  igraph, 2–9
  layout_, 9
  make_adjmatrix, 3
  make_adjmatrix_graph(make_adjmatrix), 3
  make_commonlink, 4
  make_commonlink_adjmat
    (make_commonlink), 4
  make_commonlink_graph
    (make_commonlink), 4
  make_distance_adjmat
    (make_distance_graph), 5
  make_distance_graph, 5
  make_relationship
    (make_distance_graph), 5
  make_sigma, 5
  make_sigma_mat_adjmat(make_sigma), 5
  make_sigma_mat_dist_adjmat
    (make_sigma), 5
  make_sigma_mat_dist_graph, 2, 3
  make_sigma_mat_dist_graph(make_sigma), 5
  make_sigma_mat_graph, 2, 3
  make_sigma_mat_graph(make_sigma), 5
  make_state_matrix, 2, 6
  Pi3K_AKT_graph, 7
  Pi3K_graph, 8
  plot_directed, 2, 7, 8
  RAF_MAP_graph, 10
  rmvnorm, 2, 5
  TGFBeta_Smad_graph, 11