Package 'simPATHy'

November 23, 2021

Type Package

Title A Method for Simulating Data from Perturbed Biological Pathways

Version 0.6

Date 2021-11-23

Description Simulate data from a Gaussian graphical model or a Gaussian Bayesian network in two conditions. Given a covariance matrix of a reference condition simulate plausible disregulations. See Salviato et al. (2017) <doi:10.1093/bioinformatics/btw642>.

Depends R (>= 3.0)

Imports mvtnorm, gRbase, graph, igraph, ggm, qpgraph, R.utils, htmlwidgets, shiny, shinydashboard ,grDevices, graphics

License AGPL-3

LazyData TRUE

RoxygenNote 7.1.1

Suggests knitr, rmarkdown, clipper, topologyGSA

VignetteBuilder knitr

Encoding UTF-8

NeedsCompilation no

Author Elisa Salviato [aut, cre],
Vera Djordjilovic [aut],
Chiara Romualdi [aut],
Monica Chiogna [aut]

Maintainer Elisa Salviato <elisa.salviato.88@gmail.com>

Repository CRAN

Date/Publication 2021-11-23 09:50:02 UTC

R topics documented:

chimera 2
easyLookDys 3
easyLookShiny 3

fitSgraph 4
generatePath 5
getPathShiny 5
graphNELD3-shiny 6
makePositiveDefinite 6
plotCorGraph 7
plotGraphNELD3 9
simPATHy 10
SMLEdecomposable 13

Index **15**

`chimera` *Chimera data*

Description

A matrix containing the expression values of 3405 genes deriving from Affimetrix single channel technology, consisting of 41 observations from one experimental condition (absence of BCR/ABL gene arrangment, class 1), and 37 observations from another experimental condition (presence of BCR/ABL gene arrangment, class 2).

Usage

```
chimera
```

Format

A matrix with 8405 genes (rows) and 78 samples (columns).

Source

Sabina Chiaretti, Xiaochun Li, Robert Gentleman, Antonella Vitale, Marco Vignetti, Franco Mandelli, Jerome Ritz, and Robin Foa Gene expression profile of adult T-cell acute lymphocytic leukemia identifies distinct subsets of patients with different response to therapy and survival. Blood, 1 April 2004, Vol. 103, No. 7.

Examples

```
data(chimera)
```

easyLookDys — *Dysregulation summary*

Description

Summary of the result for a quick look of simPATHy function.

Usage

```
easyLookDys(resObj, digits = 4)
```

Arguments

`resObj`	The output of simPATHy function (simPATHy class object).
`digits`	Integer indicating the number of decimal places to be used.

Value

Nice formatted output of simPATHy dysregulation

Nicely formatted output of simPATHy dysregulation.

easyLookShiny — *Visual dysregulation summary*

Description

A Shiny application for visual summary of dysregulation.

Usage

```
easyLookShiny(resObj, graph, heightGraph = NULL, heightMatrix = NULL)
```

Arguments

`resObj`	The output of simPATHy function
`graph`	The graphNEL object given to the simPATHy function to obtain resObj.
`heightGraph, heightMatrix`	The height of the graph and correlation matrix plots in pixels. Must be a number, which will be coerced to a string and have 'px' append.

Value

Interactive plots for exploring the output of simPATHy.

See Also

`simPATHy`, `plotGraphNELD3`, `plotCorGraph`, `easyLookDys` fitSgraph *Estimate covariance matrix of a graphical model*

Description

Fit a Gaussian Graphical Model or a Gaussian Bayesian Network by maximum likelihood.

Usage

```
fitSgraph(graph, S)
```

Arguments

`graph`	A directed or undirected graph represented as a graphNEL object.
`S`	A sample covariance matrix

Details

If graph is undirected it uses the Iterative Proprotional Fitting algoritm (**qpgraph** package). If graph is directed it uses Iterative Conditional Fitting (**ggm** package).

Value

A covariance matrix with the independence constraints entailed by the graph.

References

Drton, M. & Richardson, T. S. (2003). A new algorithm for maximum likelihood estimation in Gaussian graphical models for marginal independence. Proceedings of the Ninetheen Conference on Uncertainty in Artificial Intelligence, 184-191.

Whittaker, J. Graphical models in applied multivariate statistics. Wiley, 1990.

See Also

`icfmag`, `qpIPF`

`generatePath` *Find one path in a graph*

Description

Find one shortest path in the graph between two given nodes.

Usage

```
generatePath(graph, from = NULL, to = NULL)
```

Arguments

`graph`	A directed or undirected graph represented as a graphNEL object.
`from, to`	The nodes (character node id) giving the first and the last nodes of the path to be calculated. If `NULL` then the `from` and `to` nodes are randomly choosen.

Value

A list of edges in edgesList format (see **gRbase**).

See Also

`get.all.shortest.paths`

`getPathShiny` *Choose a path in a graph from an interactive shiny app*

Description

Choose a path in a graph from an interactive shiny app with the rigth format for simPATHy function.

Usage

```
getPathShiny(graph)
```

Arguments

`graph`	A graphNEL object.

Value

Selected path with the right format for simPATHy function.

See Also

`simPATHy`

Examples

```
if(require(gRbase)){
 graph <- gRbase::dag(~c:a, ~c:b, ~d:c, ~e:d)

 # Launch the interactive plot
 # path <- getPathShiny(graph)
}
```

---

graphNELD3-shiny *Shiny bindings for plotGraphNELD3*

Description

Output and render functions for using plotGraphNELD3 within Shiny applications and interactive Rmd documents.

Usage

```
graphNELD3Output(outputId, width = "100%", height = "400px")

renderGraphNELD3(expr, env = parent.frame(), quoted = FALSE)
```

Arguments

`outputId`	Output variable to read from
`width, height`	Must be a valid CSS unit (like `'100%'`, `'400px'`, `'auto'`) or a number, which will be coerced to a string and have `'px'` appended.
`expr`	An expression that generates a graphNELD3
`env`	The environment in which to evaluate `expr`.
`quoted`	Is `expr` a quoted expression (with `quote()`)? This is useful if you want to save an expression in a variable.

---

makePositiveDefinite *Positive definite matrix*

Description

Adjust the diagonal of a symmetric square matrix, by the smallest eigenvalue method, in order to make it positive definite.

Usage

```
makePositiveDefinite(M1, M2 = NULL, threshold = 0.1)
```

Arguments

`M1, M2`	A squared numeric matrix, typically a correlation or a covariance matrix. It must be symmetric.
`threshold`	A correction factor.

Details

Finds the smallest eigenvalue lambda of `M1` (or `M1` and `M2` if supplied) and adds (threshold-lambda) to the diagonal to make it positive definite.

Value

A list with the corrected input matrices and the correction `threshold`-lambda.

`plotCorGraph` *Plot correlation or partial correlation matrix*

Description

Plot a correlation or partial correlation matrix with the possibility to emphasize the graphical structure.

Usage

```
plotCorGraph(
  S1,
  type = "cor",
  S2 = NULL,
  graph = NULL,
  path = NULL,
  main = "",
  colLim = c(-1, 1),
  legendColor = TRUE
)
```

Arguments

`S1, S2`	Sample covariance matrix. If `S2` supplied, the difference between the two corresponding correlation or partial correlation matrices is plotted.
`type`	Character string specifying which matrix is to be plotted. Either `cor` for correlation matrix, or `pcor` for partial correlation matrix.
`graph`	A graphNEL object.
`path`	A list of edges in edgesList format (see **gRbase**).
`main`	The main title.
`colLim`	Numeric vector of length two specifying the lower and upper bound of the color range (see Details).
`legendColor`	Logical value indicating whether the color legend should be added to the plot.

Details

If the graph is supplied, the zero elements of the adjacency matrix are represented as shaded squares, whereas non-zero elements are represented as squares with grey borderline.

Admissible values for colLim are contained in the interval [-1,1] when S2=NULL, otherwise the admissible interval is [-2,2]. When an element is outside of the colLim interval, it is colored gray.

Value

Correlation or partial correlation matrix plot.

Examples

```
if( require(gRbase) & require(graph)){
 graph <- gRbase::ug(~a:b, ~a:c, ~c:d, ~b:d, ~b:c)

 S <- matrix(c(2,  0.8,0.5,-0.3,
             0.8,1.5,0.6,-0.7,
             0.5,0.6,1,  0.7,
             -0.3,-0.7,0.7,3), ncol=4,nrow=4)
 colnames(S) <- rownames(S) <- graph::nodes(graph)

 # Plot the correlation matrix of S
 plotCorGraph(S)


 S<-fitSgraph(graph = graph,S = S)
 # Change the color range
 plotCorGraph(S, colLim=c(-0.5,0.5))

 # Visualize the adjacency matrix
 plotCorGraph(S, type="cor", graph = graph)

 # Show the partial correlation matrix
 plotCorGraph(S, type="pcor", graph = graph)

 # Plot the difference between two matrices
 S2 <- S
 # Change the element c~a
 S2["a","c"] <- S2["c","a"]<- -0.1
 plotCorGraph(S1=S, S2=S2)
 plotCorGraph(S1=S, S2=S2, type="pcor")

 S2<-fitSgraph(graph = graph,S = S2)
 # Highlight the graphical structure
 plotCorGraph(S1=S, S2=S2, type="pcor",graph = graph)
 # Highlight the element c~a
 plotCorGraph(S1=S, S2=S2, type="pcor",graph = graph,path = list(c("a","c")))

}
```

plotGraphNELD3 *Dynamic plot of a graph*

Description

Dynamic plot of a graphNEL object with the possibility to emphasize the strength of relations between nodes, represented by either a pairwise correlation or a partial correlation coefficient.

The interactive graph is an implementation of the javascript D3.js package (force-layout) for undirected and directed graphNEL objects (see references).

Usage

```
plotGraphNELD3(
  graph,
  type = "cor",
  S1 = NULL,
  S2 = NULL,
  colLim = c(-1, 1),
  legendColor = TRUE,
  colNode = "#c0c0c0"
)
```

Arguments

`graph`	A graphNEL object.
`type`	Character string specifying which matrix is to be used. Either cor for correlation matrix, or pcor for partial correlation matrix.
`S1, S2`	Sample covariance matrix. If `S1` is supplied edges between nodes are colored in accordance with pairwise correlation or partial correlation coefficients. If `S2` supplied, the difference between the two corresponding correlation or partial correlation matrices is plotted.
`colLim`	Numeric vector of length two specifying the lower and upper bound of the color range (see Details).
`legendColor`	ogical value indicating whether the color legend should be added to the plot.
`colNode`	A character string specifying the colour of the nodes. The colour node is common for all nodes.

Details

Admissible values for `colLim` are contained in the interval `[-1,1]` when `S2=NULL`, otherwise the admissible interval is `[-2,2]`. When an element is outside of the colLim interval, it is colored gray and represented as a dashed link.

Value

Dynamic plot of a graphNEL object.

References

https://d3js.org (Micheal Bostock).

http://www.htmlwidgets.org (Ramnath Vaidyanathan, Kenton Russell, and RStudio).

https://christophergandrud.github.io/networkD3/ (Christopher Gandrud, JJ Allaire, & Kent Russell)

Examples

```
if(require(gRbase) & require(graph)){
  graph <- gRbase::ug(~a:b, ~a:c, ~c:d, ~b:d, ~b:c)
  # Plot a graphNEL
  plotGraphNELD3(graph)

 # Plot a graphNEL coloring edges in correspondance with pairwise correlation coefficients
  S <- matrix(c(2,  0.8,0.5,-0.3,
                0.8,1.5,0.6,-0.7,
                0.5,0.6,1,  0.7,
                -0.3,-0.7,0.7,3), ncol=4,nrow=4)
  colnames(S) <- rownames(S) <- graph::nodes(graph)
  plotGraphNELD3(graph, S1=S)

 # Plot a graphNEL coloring edges in correspondance with partial correlation coefficients
  plotGraphNELD3(graph, S1=S, type="pcor")

  # Change the color range
  plotGraphNELD3(graph, S1=S, type="cor", colLim=c(-0.7,0.8))
  # Change nodes color
  plotGraphNELD3(graph, S1=S, type="cor", colNode = "pink")

  # Plot the difference between two graphical models
  S2 <- S
  S2[1,3] <- S2[3,1]<- -0.1
  plotGraphNELD3(graph,S1=S, S2=S2)

}
```

---

simPATHy *Simulate data from a graphical model*

Description

Simulate data in two different conditions with a common structure of dependences. The two different conditions are characterized by different strengths of the links between nodes (dysregulation).

Usage

```
simPATHy(
  graph,
```

```
  path = NULL,
  S = NULL,
  min = 2,
  max = 3,
  prob = 1,
  n1 = 500,
  n2 = n1,
  digits = 5,
  mu1 = 0,
  mu2 = mu1,
  muRandom = FALSE
)
```

Arguments

graph	A graphNEL object.
path	A list of edges in edgesList format (see **gRbase**).
S	The sample covariance matrix.
min, max	Vectors of length 1 or of the same length as path containing the lower and upper limits of a uniform distribution. The strength of dysregulation is sampled uniformly from the interval [min, max]: a value smaller than 1 represents deactivation, a value greater than 1 represents activation. If path=NULL only the first element is used.
prob	A vector of size 1 or of the same length as path, giving the probability to change the sign of the correlation coefficient for each edge. prob=0 implying that the sign of the dysregulation should be changed, and prob=1 implying that the sign should be left unaltered (default). Values between these two extremes allow for random sign switch: the sign is changed with probability 1-prob.
n1, n2	Number of observations to generate from the two conditions.
digits	Integer indicating the number of decimal places to be used.
mu1, mu2	A vector of size 1 or of the length equal to the number of nodes in the graph. Means of the multivariate normal distributions from which observations are generated. If mu1 (and/or mu2) is a vector it has to be named in accordance with the names of the nodes of the graph.
muRandom	Logical. If muRandom=TRUE the means of the variables are randomly generated.

Details

If the matrix S does not reflect conditional independence constraints imposed by the graph simPATHy uses the maximum likelihood estimation of covariance matrices for graphical models via internal function fitSgraph.

When the dysregulation of the initial (reference condition) covariance matrix leads to a matrix that is no longer positive definite, the resulting matrix is corrected via internal function makePositiveDefinite.

To avoid excessively strong dysregulations, the upper limit for the absolute value of the dysregulated correlation coefficient is set to:

$$min(0.9, 1.25 * max(abs(C[upper.tri(C)])))$$

where C is the correlation matrix of the reference condition.

Value

It returns a list containing:

- `data` random samples generated from multivariate normal distributions with covariance matrices `S1` (reference condition) and `S2` (dysregulated condition);
- `S1,S2` two covariance matrices;
- `path` the dysregulated path;
- `strength` the dysregulation strength for each edge in the path;
- `mu1,mu2` two mean vectors;
- `correction` correction details.

See Also

`easyLookDys`, `easyLookShiny`, `plotCorGraph`, `plotGraphNELD3`

Examples

```
if(require(gRbase) & require(graph)){

  ## Directed graph
  ## sub-graph Acute Myel... Leukemia
  graph<-gRbase::dag(~867:25+867:613+5295:867+5294:867+
                     + 207:5295+207:5294+4193:207+3551:207+
                     + 4792:3551+7157:4193+3265:6654+
                     + 3845:6654+6654:2885+2885:25+2885:613)
  genes<-graph::nodes(graph)

  # covariance matrix of the reference condition
  data<-t(chimera[genes,colnames(chimera)==1])
  S<-cov(data)
  S<-fitSgraph(graph,S)

  # select path to dysregulate
  path<-list(c("613","867"),c("867","5295"),c("5295","207"),
             c("207","4193"),c("4193","7157"))
  ## ..or select the path in an interactive plot
  # path<-getPathShiny(graph)

  # select parameters of the dysregulation
  min<-c(2,8,2,0.1,0.5)
  max<-c(2,10,2,4,0.5)
  prob<-c(1,0,0,0.5,1)
```

```
# activation, switch, switch, random, deactivation
dys<-cbind(min,max,prob)
rownames(dys)<-sapply(path,paste,collapse = "~")
dys

set.seed(123)
# main function
Result<-simPATHy(graph,path,S,min,max,prob)
class(Result)
names(Result)

# simulated data from two conditions
round(Result$dataset[c(1:3,501:503),1:5],3)

# Summary
easyLookDys(Result)
# ..or interactive summary
# easyLookShiny(resObj=Result,graph=graph)


# Visualization
plotCorGraph(S1=Result$S1,S2 = Result$S2,graph = graph,path = path,colLim = c(-0.3,0.3))
plotGraphNELD3(S1=Result$S1,S2 = Result$S2,graph = graph,colLim = c(-0.3,0.3))


## Undirected graph
graph <- gRbase::ug(~a:b, ~a:c, ~c:d, ~b:d, ~b:c)
# when reference condition covariance matrix is not supplied simPATHy generate a random one
Result_ug<-simPATHy(graph)
easyLookDys(Result_ug)
plotGraphNELD3(S1=Result_ug$S1,S2 = Result_ug$S2,graph = graph,colLim = c(-0.5,0.5))

}
```

SMLEdecomposable   *Local Maximum Likelihood Estimation*

Description

Compute a maximum likelihood estimate of a covariance matrix in a decomposable Gaussian graphical model.

Usage

```
SMLEdecomposable(S, graph)
```

Arguments

S	a covariance matrix.
graph	a decomposable graph represented as a graphNEL object.

Value

The MLE of a covarince matrix.

References

Lauritzen, S. L. (1996). Graphical Models. Clarendon Press, Oxford.

Index

∗ **datasets**
    chimera, 2

chimera, 2

easyLookDys, *3*, 3, *12*
easyLookShiny, 3, *12*

fitSgraph, 4, *11*

generatePath, 5
get.all.shortest.paths, *5*
getPathShiny, 5
graphNELD3-shiny, 6
graphNELD3Output *(*graphNELD3-shiny*)*, 6

icfmag, *4*

makePositiveDefinite, 6, *11*

plotCorGraph, *3*, 7, *12*
plotGraphNELD3, *3*, 9, *12*

qpIPF, *4*

renderGraphNELD3 *(*graphNELD3-shiny*)*, 6

simPATHy, *3*, *5*, 10
SMLEdecomposable, 13