

Package ‘BDgraph’

May 3, 2021

Title Bayesian Structure Learning in Graphical Models using Birth-Death MCMC

Version 2.64

Description Statistical tools for Bayesian structure learning in undirected graphical models for continuous, discrete, and mixed data. The package is implemented the recent improvements in the Bayesian graphical models literature, including Mohammadi and Wit (2015) <doi:10.1214/14-BA889>, Mohammadi and Wit (2019) <doi:10.18637/jss.v089.i03>.

URL <https://www.uva.nl/profile/a.mohammadi>

Imports igraph

Suggests ssgraph, huge, pROC, ggplot2, tmvtnorm

License GPL (>= 2)

Repository CRAN

NeedsCompilation yes

Author Reza Mohammadi [aut, cre] (<<https://orcid.org/0000-0001-9538-0648>>), Ernst Wit [aut] (<<https://orcid.org/0000-0002-3671-9610>>), Adrian Dobra [ctb] (<<https://orcid.org/0000-0001-7793-2197>>)

Maintainer Reza Mohammadi <a.mohammadi@uva.nl>

Date/Publication 2021-05-03 04:50:12 UTC

R topics documented:

BDgraph-package	2
adj2link	4
bdgraph	5
bdgraph.mpl	8
bdgraph.npn	10
bdgraph.sim	12
bf	14
churn	15
compare	17
conf.mat	18

conf.mat.plot	20
covariance	21
Discrete Weibull	22
geneExpression	24
gnorm	24
graph.sim	26
link2adj	27
pgraph	28
plinks	30
plot.bdgraph	31
plot.graph	32
plot.sim	33
plotcoda	34
plotroc	36
precision	37
print.bdgraph	38
print.sim	39
reinis	40
rgwish	41
rmvnorm	43
roc	44
rwish	46
select	47
sparsity	49
summary.bdgraph	50
surveyData	51
traceplot	52
transfer	53
Index	55

Description

The R package **BDgraph** provides statistical tools for Bayesian structure learning in undirected graphical models for continuous, count, binary, and mixed data. The package is implemented the recent improvements in the Bayesian graphical models literature, including Mohammadi and Wit (2015), Mohammadi et al. (2017), Dobra and Mohammadi (2018), and Letac et al. (2018). The computationally intensive tasks of the package are implemented in parallel using **OpenMP** in C++ and interfaced with R, to speed up the computations. Besides, the package contains several functions for simulation and visualization, as well as several multivariate datasets taken from the literature.

How to cite this package

To cite **BDgraph** in publications use:

Mohammadi, R. and Wit, E. C. (2019). BDgraph: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30. doi:10.18637/jss.v089.i03

Author(s)

Reza Mohammadi
Amsterdam Business School
University of Amsterdam
Maintainer: Reza Mohammadi <a.mohammadi@uva.nl>

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30.
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Mohammadi, A., et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845
- Dobra, A. and Lenkoski, A. (2011). Copula Gaussian graphical models and their application to modeling functional disability data, *The Annals of Applied Statistics*, 5(2A):969-93
- Dobra, A., et al. (2011). Bayesian inference for general Gaussian graphical models with application to multivariate lattice data. *Journal of the American Statistical Association*, 106(496):1418-33
- Mohammadi, A. and Dobra, A. (2017). The R Package **BDgraph** for Bayesian Structure Learning in Graphical Models, *ISBA Bulletin*, 24(4):11-16
- Lenkoski, A. (2013). A direct sampler for G-Wishart variates, *Stat*, 2(1):119-28
- Pensar, J. et al (2017) Marginal pseudo-likelihood learning of discrete Markov network structures, *Bayesian Analysis*, 12(4):1195-215

See Also

[bdgraph](#), [bdgraph.mpl](#), [bdgraph.sim](#), [compare](#), [rgwish](#)

Examples

```
## Not run:  
library( BDgraph )  
  
# Generating multivariate normal data from a 'random' graph
```

```

data.sim <- bdgraph.sim( n = 70, p = 6, size = 7, vis = TRUE )

# Running algorithm based on GGMS
bdgraph.obj <- bdgraph( data = data.sim, iter = 5000 )

summary( bdgraph.obj )

# To compare the result with true graph
compare( data.sim, bdgraph.obj, main = c( "Target", "BDgraph" ), vis = TRUE )

# Running algorithm based on GGMS and marginal pseudo-likelihood
bdgraph.obj_mpl <- bdgraph.mpl( data = data.sim, iter = 5000 )

summary( bdgraph.obj_mpl )

# To compare the results of both algorithms with true graph
compare( data.sim, bdgraph.obj, bdgraph.obj_mpl,
        main = c( "Target", "BDgraph", "BDgraph_mpl" ), vis = TRUE )

## End(Not run)

```

adj2link

Extract links from an adjacency matrix

Description

Extract links from an adjacency matrix or an object of classes "sim" from function [bdgraph.sim](#) and "graph" from function [graph.sim](#).

Usage

```
adj2link( adj )
```

Arguments

adj An adjacency matrix corresponding to a graph structure in which $a_{ij} = 1$ if there is a link between nodes i and j , otherwise $a_{ij} = 0$. It can be an object with S3 class "sim" from function [bdgraph.sim](#). It can be an object with S3 class "graph" from function [graph.sim](#).

Value

A matrix corresponding to the extracted links from graph structure.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845
- Pensar, J. et al (2017) Marginal pseudo-likelihood learning of discrete Markov network structures, *Bayesian Analysis*, 12(4):1195-215

See Also

[link2adj, graph.sim](#)

Examples

```
# Generating a 'random' graph
adj <- graph.sim( p = 6, vis = TRUE )

adj2link( adj )
```

bdgraph

Search algorithm in graphical models

Description

As the main function of the **BDgraph** package, this function consists of several MCMC sampling algorithms for Bayesian model determination in undirected graphical models. To speed up the computations, the birth-death MCMC sampling algorithms are implemented in parallel using **OpenMP** in C++.

Usage

```
bdgraph( data, n = NULL, method = "ggm", algorithm = "bdmcmc", iter = 5000,
  burnin = iter / 2, not.cont = NULL, g.prior = 0.5, df.prior = 3,
  g.start = "empty", jump = NULL, save = FALSE,
  cores = NULL, threshold = 1e-8 )
```

Arguments

data	There are two options: (1) an $(n \times p)$ matrix or a data.frame corresponding to the data, (2) an $(p \times p)$ covariance matrix as $S = X'X$ which X is the data matrix (n is the sample size and p is the number of variables). It also could be an object of class "sim", from function <code>bdgraph.sim</code> . The input matrix is automatically identified by checking the symmetry.
n	The number of observations. It is needed if the "data" is a covariance matrix.
method	A character with two options "ggm" (default) and "gcgM". Option "ggm" is for Gaussian graphical models based on Gaussianity assumption. Option "gcgM" is for Gaussian copula graphical models for the data that not follow Gaussianity assumption (e.g. continuous non-Gaussian, count, or mixed dataset).
algorithm	A character with two options "bdmcmc" (default) and "rjmcM". Option "bdmcmc" is based on birth-death MCMC algorithm. Option "rjmcM" is based on reversible jump MCMC algorithm.
iter	The number of iteration for the sampling algorithm.
burnin	The number of burn-in iteration for the sampling algorithm.
not.cont	For the case method = "gcgM", a vector with binary values in which 1 indicates not continuous variables.
g.prior	For determining the prior distribution of each edge in the graph. There are two options: a single value between 0 and 1 (e.g. 0.5 as a noninformative prior) or an $(p \times p)$ matrix with elements between 0 and 1.
df.prior	The degree of freedom for G-Wishart distribution, $W_G(b, D)$, which is a prior distribution of the precision matrix.
g.start	Corresponds to a starting point of the graph. It could be an $(p \times p)$ matrix, "empty" (default), or "full". Option "empty" means the initial graph is an empty graph and "full" means a full graph. It also could be an object with S3 class "bdgraph" of R package <code>BDgraph</code> or the class "ssgraph" of R package <code>ssgraph::ssgraph()</code> ; this option can be used to run the sampling algorithm from the last objects of previous run (see examples).
jump	It is only for the BDMCMC algorithm (algorithm = "bdmcmc"). It is for simultaneously updating multiple links at the same time to update graph in the BDMCMC algorithm.
save	Logical: if FALSE (default), the adjacency matrices are NOT saved. If TRUE, the adjacency matrices after burn-in are saved.
cores	The number of cores to use for parallel execution. The case cores="all" means all CPU cores to use for parallel execution.
threshold	The threshold value for the convergence of sampling algorithm from G-Wishart for the precision matrix.

Value

An object with S3 class "bdgraph" is returned:

p_links	An upper triangular matrix which corresponds the estimated posterior probabilities of all possible links.
---------	---

K_hat The posterior estimation of the precision matrix.

For the case "save = TRUE" is returned:

sample_graphs A vector of strings which includes the adjacency matrices of visited graphs after burn-in.

graph_weights A vector which includes the waiting times of visited graphs after burn-in.

all_graphs A vector which includes the identity of the adjacency matrices for all iterations after burn-in. It is needed for monitoring the convergence of the BD-MCMC algorithm.

all_weights A vector which includes the waiting times for all iterations after burn-in. It is needed for monitoring the convergence of the BD-MCMC algorithm.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Ernst Wit

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845
- Mohammadi, A. and Dobra A. (2017). The R Package **BDgraph** for Bayesian Structure Learning in Graphical Models, *ISBA Bulletin*, 24(4):11-16

See Also

[bdgraph.mpl](#), [bdgraph.sim](#), [summary.bdgraph](#), [compare](#)

Examples

```
## Not run:
# --- Example 1
# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 20, p = 6, size = 7, vis = TRUE )

bdgraph.obj <- bdgraph( data = data.sim, iter = 1000 )

summary( bdgraph.obj )

# To compare our result with true graph
```

```

compare( data.sim, bdgraph.obj, main = c( "Target", "BDgraph" ) )

# Running algorithm with starting points from previous run
bdgraph.obj2 <- bdgraph( data = data.sim, g.start = bdgraph.obj )

compare( data.sim, bdgraph.obj, bdgraph.obj2,
        main = c( "Target", "Frist run", "Second run" ) )

# --- Example 2
# Generating mixed data from a 'scale-free' graph
data.sim <- bdgraph.sim( n = 50, p = 6, type = "mixed", graph = "scale-free", vis = TRUE )

bdgraph.obj <- bdgraph( data = data.sim, method = "gcmg", iter = 10000 )

summary( bdgraph.obj )

compare( data.sim, bdgraph.obj )

## End(Not run)

```

bdgraph.mpl

Search algorithm in graphical models using marginal pseudo-likelihood

Description

This function consists of several sampling algorithms for Bayesian model determination in undirected graphical models based on marginal pseudo-likelihood. To speed up the computations, the birth-death MCMC sampling algorithms are implemented in parallel using **OpenMP** in C++.

Usage

```

bdgraph.mpl( data, n = NULL, method = "ggm", transfer = TRUE,
             algorithm = "bdmcmc", iter = 5000, burnin = iter / 2,
             g.prior = 0.5, g.start = "empty",
             jump = NULL, alpha = 0.5, save = FALSE,
             cores = NULL, operator = "or" )

```

Arguments

data	There are two options: (1) an $(n \times p)$ matrix or a data.frame corresponding to the data, (2) an $(p \times p)$ covariance matrix as $S = X'X$ which X is the data matrix (n is the sample size and p is the number of variables). It also could be an object of class "sim", from function bdgraph.sim . The input matrix is automatically identified by checking the symmetry.
n	The number of observations. It is needed if the "data" is a covariance matrix.

method	A character with two options "ggm" (default), "dgm" and "dgm-binary". Option "ggm" is for Gaussian graphical models based on Gaussianity assumption. Option "dgm" is for discrete graphical models for the count data. Option "dgm-binary" is for discrete graphical models for the data that are binary.
transfer algorithm	For only count data which method = "dgm" or method = "dgm-binary". A character with two options "bdmcmc" (default) and "rjmc". Option "bdmcmc" is based on birth-death MCMC algorithm. Option "rjmc" is based on reversible jump MCMC algorithm. Option "hc" is based on hill-climbing algorithm; this algorithm is only for count data which method = "dgm" or method = "dgm-binary".
iter	The number of iteration for the sampling algorithm.
burnin	The number of burn-in iteration for the sampling algorithm.
g.prior	For determining the prior distribution of each edge in the graph. There are two options: a single value between 0 and 1 (e.g. 0.5 as a noninformative prior) or an $(p \times p)$ matrix with elements between 0 and 1.
g.start	Corresponds to a starting point of the graph. It could be an $(p \times p)$ matrix, "empty" (default), or "full". Option "empty" means the initial graph is an empty graph and "full" means a full graph. It also could be an object with S3 class "bdgraph" of R package BDgraph or the class "ssgraph" of R package ssgraph::ssgraph() ; this option can be used to run the sampling algorithm from the last objects of previous run (see examples).
jump	It is only for the BDMCMC algorithm (algorithm = "bdmcmc"). It is for simultaneously updating multiple links at the same time to update graph in the BDMCMC algorithm.
alpha	Value of the hyper parameter of Dirichlet, which is a prior distribution.
save	Logical: if FALSE (default), the adjacency matrices are NOT saved. If TRUE, the adjacency matrices after burn-in are saved.
cores	The number of cores to use for parallel execution. The case cores="all" means all CPU cores to use for parallel execution.
operator	A character with two options "or" (default) and "and". It is for hill-climbing algorithm.

Value

An object with S3 class "bdgraph" is returned:

`p_links` An upper triangular matrix which corresponds the estimated posterior probabilities of all possible links.

For the case "save = TRUE" is returned:

`sample_graphs` A vector of strings which includes the adjacency matrices of visited graphs after burn-in.

`graph_weights` A vector which includes the waiting times of visited graphs after burn-in.

`all_graphs` A vector which includes the identity of the adjacency matrices for all iterations after burn-in. It is needed for monitoring the convergence of the BD-MCMC algorithm.

`all_weights` A vector which includes the waiting times for all iterations after burn-in. It is needed for monitoring the convergence of the BD-MCMC algorithm.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>, Adrian Dobra, and Johan Pensar

References

Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845

Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138

Mohammadi, A. and Dobra, A. (2017). The R Package **BDgraph** for Bayesian Structure Learning in Graphical Models, *ISBA Bulletin*, 24(4):11-16

Pensar, J. et al (2017) Marginal pseudo-likelihood learning of discrete Markov network structures, *Bayesian Analysis*, 12(4):1195-215

Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30

See Also

[bdgraph](#), [bdgraph.sim](#), [summary.bdgraph](#), [compare](#)

Examples

```
# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 70, p = 5, size = 7, vis = TRUE )

bdgraph.obj <- bdgraph.mpl( data = data.sim, iter = 500 )

summary( bdgraph.obj )

# To compare the result with true graph
compare( data.sim, bdgraph.obj, main = c( "Target", "BDgraph" ) )
```

bdgraph.npn

Nonparametric transfer

Description

Transfers non-Gaussian data to Gaussian.

Usage

```
bdgraph.npn( data, npn = "shrinkage", npn.thresh = NULL )
```

Arguments

data	An $(n \times p)$ matrix or a data.frame corresponding to the data (n is the sample size and p is the number of variables).
npn	A character with three options "shrinkage" (default), "truncation", and "skeptical". Option "shrinkage" is for the shrunken transformation, option "truncation" is for the truncated transformation and option "skeptical" is for the non-paranormal skeptical transformation. For more details see references.
npn.thresh	The truncation threshold; it is only for the truncated transformation (npn= "truncation"). The default value is $1/(4n^{1/4}\sqrt{\pi \log(n)})$.

Value

An $(n \times p)$ matrix of transferred data, if npn = "shrinkage" or "truncation", and a non-paranormal correlation $(p \times p)$ matrix, if npn = "skeptical".

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

Liu, H., et al (2012). High Dimensional Semiparametric Gaussian Copula Graphical Models, *Annals of Statistics*, 40(4):2293-2326

Zhao, T. and Liu, H. (2012). The **huge** Package for High-dimensional Undirected Graph Estimation in R, *Journal of Machine Learning Research*, 13:1059-1062

See Also

[bdgraph.sim](#), [bdgraph](#), [bdgraph.mpl](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 6, p = 4, size = 4 )
data      <- ( data.sim $ data - 3 ) ^ 4
data

# Transfer the data by truncation
bdgraph.npn( data, npn = "truncation" )

# Transfer the data by shrunken
bdgraph.npn( data, npn = "shrunken" )

# Transfer the data by skeptical
bdgraph.npn( data, npn = "skeptical" )

## End(Not run)
```

bdgraph.sim

*Graph data simulation***Description**

Simulating multivariate distributions with different types of underlying graph structures, including "random", "cluster", "scale-free", "lattice", "hub", "star", "circle", "AR(1)", and "AR(2)". Based on the underlying graph structure, it generates different types of *multivariate* data, including *multivariate Gaussian, non-Gaussian, count, mixed, binary, or discrete Weibull* data. This function can be used also for only simulating graphs by option $n=0$, as a default.

Usage

```
bdgraph.sim( p = 10, graph = "random", n = 0, type = "Gaussian", prob = 0.2,
  size = NULL, mean = 0, class = NULL, cut = 4, b = 3,
  D = diag( p ), K = NULL, sigma = NULL,
  q = exp(-1), beta = 1, vis = FALSE )
```

Arguments

p	The number of variables (nodes).
graph	The graph structure with options "random", "cluster", "scale-free", "lattice", "hub", "star", "circle", "AR(1)", and "AR(2)". It also could be an adjacency matrix corresponding to a graph structure (an upper triangular matrix in which $g_{ij} = 1$ if there is a link between notes i and j , otherwise $g_{ij} = 0$).
n	The number of samples required. Note that for the case $n = 0$, only graph is generated.
type	Type of data with four options "Gaussian" (default), "non-Gaussian", "count", "mixed", "binary", and "dw". For the option "Gaussian", data are generated from multivariate normal distribution. For the option "non-Gaussian", data are transferred multivariate normal distribution to continuous multivariate non-Gaussian distribution. For the option "count", data are transferred from multivariate normal distribution to multivariate count data. For the option "mixed", data are transferred from multivariate normal distribution to mixture of 'count', 'ordinal', 'non-Gaussian', 'binary' and 'Gaussian', respectively. For the option "binary", data are generated directly from the joint distribution, in this case p must be less than 17. For the option "dw", data are transferred from multivariate normal distribution to the discrete Weibull distribution with parameters q and β .
prob	If graph="random", it is the probability that a pair of nodes has a link.
size	The number of links in the true graph (graph size).
mean	A vector specifies the mean of the variables.
class	If graph="cluster", it is the number of classes.
cut	If type="count", it is the number of categories for simulating count data.

b	The degree of freedom for G-Wishart distribution, $W_G(b, D)$.
D	The positive definite ($p \times p$) "scale" matrix for G-Wishart distribution, $W_G(b, D)$. The default is an identity matrix.
K	If graph="fixed", it is a positive-definite symmetric matrix specifies as a true precision matrix.
sigma	If graph="fixed", it is a positive-definite symmetric matrix specifies as a true covariance matrix.
q, beta	If type="dw", they are the parameters of the discrete Weibull distribution with density $p(x, q, \beta) = q^{x^\beta} - q^{(x+1)^\beta}, \quad \forall x = \{0, 1, 2, \dots\}.$
vis	Visualize the true graph structure.

Value

An object with S3 class "sim" is returned:

data	Generated data as an ($n \times p$) matrix.
sigma	The covariance matrix of the generated data.
K	The precision matrix of the generated data.
G	The adjacency matrix corresponding to the true graph structure.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>, Pariya Behrouzi, Veronica Vinciotti, and Ernst Wit

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Pensar, J. et al (2017) Marginal pseudo-likelihood learning of discrete Markov network structures, *Bayesian Analysis*, 12(4):1195-215

See Also

[graph.sim](#), [bdgraph](#), [bdgraph.mpl](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( p = 10, n = 50, prob = 0.3, vis = TRUE )
print( data.sim )

# Generating multivariate normal data from a 'hub' graph
data.sim <- bdgraph.sim( p = 6, n = 3, graph = "hub", vis = FALSE )
round( data.sim $ data, 2 )

# Generating mixed data from a 'hub' graph
data.sim <- bdgraph.sim( p = 8, n = 10, graph = "hub", type = "mixed" )
round( data.sim $ data, 2 )

# Generating only a 'scale-free' graph (with no data)
graph.sim <- bdgraph.sim( p = 8, graph = "scale-free" )
plot( graph.sim )
graph.sim $ G

## End(Not run)
```

 bf

Bayes factor between two graphs

Description

Compute the Bayes factor between the structure of two graphs.

Usage

```
bf( num, den, bdgraph.obj, log = TRUE )
```

Arguments

num, den	An adjacency matrix corresponding to the true graph structure in which $a_{ij} = 1$ if there is a link between notes i and j , otherwise $a_{ij} = 0$. It can be an object with S3 class "graph" from function graph.sim . It can be an object with S3 class "sim" from function bdgraph.sim .
bdgraph.obj	An object of S3 class "bdgraph", from function bdgraph . It also can be an object of S3 class "ssgraph", from the function ssgraph::ssgraph() of R package ssgraph::ssgraph() .
log	A character value. If TRUE the Bayes factor is given as log(BF).

Value

A single numeric value, the Bayes factor of the two graph structures num and den.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845

See Also

[bdgraph](#), [bdgraph.mpl](#), [compare](#), [bdgraph.sim](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'circle' graph
data.sim <- bdgraph.sim( n = 50, p = 6, graph = "circle", vis = TRUE )

# Running sampling algorithm
bdgraph.obj <- bdgraph( data = data.sim )

graph_1 <- graph.sim( p = 6, vis = TRUE )
graph_2 <- graph.sim( p = 6, vis = TRUE )

bf( num = graph_1, den = graph_2, bdgraph.obj = bdgraph.obj )

## End(Not run)
```

churn

Churn data set

Description

The data can be downloaded from IBM Sample Data Sets. Customer *churn* occurs when customers stop doing business with a company, also known as customer attrition. The data set contains 3333 rows (customers) and 20 columns (features). The "Churn" column is our target which indicate whether customer churned (left the company) or not.

Usage

```
data( churn )
```

Format

The churn dataset, as a data frame, contains 3333 rows (customers) and 20 columns (variables/features). The 20 variables are:

- State: Categorical, for the 50 states and the District of Columbia.
- Account.Length: count, how long account has been active.
- Area.Code: Categorical.
- Int.l.Plan: Categorical, yes or no, international plan.
- VMail.Plan: Categorical, yes or no, voice mail plan.
- VMail.Message: Count, number of voice mail messages.
- Day.Mins: Continuous, minutes customer used service during the day.
- Day.Calls: Count, total number of calls during the day.
- Day.Charge: Continuous, total charge during the day.
- Eve.Mins: Continuous, minutes customer used service during the evening.
- Eve.Calls: Count, total number of calls during the evening.
- Eve.Charge: Continuous, total charge during the evening.
- Night.Mins: Continuous, minutes customer used service during the night.
- Night.Calls: Count, total number of calls during the night.
- Night.Charge: Continuous, total charge during the night.
- Intl.Mins: Continuous, minutes customer used service to make international calls.
- Intl.Calls: Count, total number of international calls.
- Intl.Charge: Continuous, total international charge.
- CustServ.Calls: Count, number of calls to customer service.
- Churn: Categorical, True or False. Indicator of whether the customer has left the company (True or False).

References

Larose, D. T. and Larose, C. D. (2014). Discovering knowledge in data: an introduction to data mining. *John Wiley & Sons*.

Examples

```
data( churn )
```

```
summary( churn )
```


compare

*Graph structure comparison***Description**

This function provides several measures to assess the performance of the graphical structure learning.

Usage

```
compare( target, est, est2 = NULL, est3 = NULL, est4 = NULL, main = NULL,
         vis = FALSE )
```

Arguments

target	An adjacency matrix corresponding to the true graph structure in which $a_{ij} = 1$ if there is a link between notes i and j , otherwise $a_{ij} = 0$. It can be an object with S3 class "sim" from function bdgraph.sim . It can be an object with S3 class "graph" from function graph.sim .
est, est2, est3, est4	An adjacency matrix corresponding to an estimated graph. It can be an object with S3 class "bdgraph" from function bdgraph . It can be an object of S3 class "ssgraph", from the function ssgraph::ssgraph() of R package ssgraph::ssgraph() . It can be an object of S3 class "select", from the function huge.select of R package huge . Options est2, est3 and est4 are for comparing two or more different approaches.
main	A character vector giving the names for the result table.
vis	Visualize the true graph and estimated graph structures.

Value

True positive	The number of correctly estimated links.
True negative	The number of true non-existing links which is correctly estimated.
False positive	The number of links which they are not in the true graph, but are incorrectly estimated.
False negative	The number of links which they are in the true graph, but are not estimated.
F1-score	A weighted average of the "positive predictive" and "true positive rate". The F1-score value reaches its best value at 1 and worst score at 0.
Specificity	The Specificity value reaches its best value at 1 and worst score at 0.
Sensitivity	The Sensitivity value reaches its best value at 1 and worst score at 0.
MCC	The Matthews Correlation Coefficients (MCC) value reaches its best value at 1 and worst score at 0.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>, Antonio Abbruzzo, and Ivan Vujacic

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845

See Also

[bdgraph](#), [bdgraph.mpl](#), [bdgraph.sim](#), [plotroc](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 50, p = 6, size = 7, vis = TRUE )

# Running sampling algorithm based on GGMs
sample.ggm <- bdgraph( data = data.sim, method = "ggm", iter = 10000 )

# Comparing the results
compare( data.sim, sample.ggm, main = c( "True", "GGM" ), vis = TRUE )

# Running sampling algorithm based on GCGMs
sample.gcgm <- bdgraph( data = data.sim, method = "gcgm", iter = 10000 )

# Comparing GGM and GCGM methods
compare( data.sim, sample.ggm, sample.gcgm, main = c( "True", "GGM", "GCGM" ), vis = TRUE )

## End(Not run)
```

conf.mat

Confusion Matrix

Description

Create a Confusion Matrix.

Usage

```
conf.mat( pred, actual, cutoff = 0.5, proportion = FALSE,
          dnn = c( "Prediction", "Actual" ), ... )
```

Arguments

pred	An adjacency matrix corresponding to an estimated graph. It can be an object with S3 class "bdgraph" from function <code>bdgraph</code> . It can be an object of S3 class "ssgraph", from the function <code>ssgraph::ssgraph()</code> of R package <code>ssgraph::ssgraph()</code> .
actual	An adjacency matrix corresponding to the actual graph structure in which $a_{ij} = 1$ if there is a link between notes i and j , otherwise $a_{ij} = 0$. It can be an object with S3 class "sim" from function <code>bdgraph.sim</code> . It can be an object with S3 class "graph" from function <code>graph.sim</code> . It can be a factor, numeric or character vector of responses (true class), typically encoded with 0 (controls) and 1 (cases). Only two classes can be used in a ROC curve.
cutoff	cutoff value for the case that pred is vector of probabilities. The default is 0.5.
proportion	Logical: FALSE (default) for a confusion matrix with number of cases. TRUE, for a confusion matrix with the proportion of cases.
dnn	the names to be given to the dimensions in the result (the dimnames names).
...	further arguments to be passed to table.

Value

the results of table on pred and actual.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

See Also

[conf.mat.plot](#), [compare](#), [roc](#), [bdgraph](#)

Examples

```
## Not run:
set.seed( 100 )

# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 50, p = 6, size = 7, vis = TRUE )

# Running sampling algorithm based on GGMs
sample.ggm <- bdgraph( data = data.sim, method = "ggm", iter = 10000 )

# Confusion Matrix for GGM method
conf.mat( pred = sample.ggm, actual = data.sim )
```

```
## End(Not run)
```

conf.mat.plot *Plot Confusion Matrix*

Description

Plot a Confusion Matrix.

Usage

```
conf.mat.plot( pred, actual, cutoff = 0.5, conf.level = 0, margin = 1,
              color = c( "#ff83a8", "#83ff9b" ), ... )
```

Arguments

pred	An adjacency matrix corresponding to an estimated graph. It can be an object with S3 class "bdgraph" from function <code>bdgraph</code> . It can be an object of S3 class "ssgraph", from the function <code>ssgraph::ssgraph()</code> of R package <code>ssgraph::ssgraph()</code> .
actual	An adjacency matrix corresponding to the actual graph structure in which $a_{ij} = 1$ if there is a link between notes i and j , otherwise $a_{ij} = 0$. It can be an object with S3 class "sim" from function <code>bdgraph.sim</code> . It can be an object with S3 class "graph" from function <code>graph.sim</code> . It can be a factor, numeric or character vector of responses (true class), typically encoded with 0 (controls) and 1 (cases). Only two classes can be used in a ROC curve.
cutoff	cutoff value for the case that pred is vector of probabilities. The default is 0.5.
conf.level	confidence level used for the confidence rings on the odds ratios. Must be a single nonnegative number less than 1; if set to 0 (the default), confidence rings are suppressed.
margin	a numeric vector with the margins to equate. Must be one of 1 (the default), 2, or <code>c(1, 2)</code> , which corresponds to standardizing the row, column, or both margins in each 2 by 2 table. Only used if <code>std</code> equals "margins".
color	a vector of length 2 specifying the colors to use for the smaller and larger diagonals of each 2 by 2 table.
...	options to be passed to <code>fourfoldplot</code> .

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

See Also

[conf.mat](#), [compare](#), [roc](#), [bdgraph](#)

Examples

```
## Not run:
set.seed( 100 )

# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 50, p = 6, size = 7, vis = TRUE )

# Running sampling algorithm based on GGMs
sample.ggm <- bdgraph( data = data.sim, method = "ggm", iter = 10000 )

# Confusion Matrix for GGM method
conf.mat.plot( pred = sample.ggm, actual = data.sim )

## End(Not run)
```

 covariance

Estimated covariance matrix

Description

Provides the estimated covariance matrix.

Usage

```
covariance( bdgraph.obj, round = 2 )
```

Arguments

bdgraph.obj	An object of S3 class "bdgraph", from function bdgraph . It also can be an object of S3 class "ssgraph", from the function ssgraph::ssgraph() of R package ssgraph::ssgraph() .
round	A value for rounding all probabilities to the specified number of decimal places.

Value

A matrix which corresponds the estimated covariance matrix.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30

Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138

Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*

Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845

Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

See Also

[bdgraph](#), [precision](#), [plinks](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'circle' graph
data.sim <- bdgraph.sim( n = 70, p = 6, graph = "circle", vis = TRUE )

bdgraph.obj <- bdgraph( data = data.sim )

covariance( bdgraph.obj ) # Estimated covariance matrix

data.sim $ sigma # True covariance matrix

## End(Not run)
```

Discrete Weibull

The Discrete Weibull Distribution (Type I)

Description

Density, distribution function, quantile function and random generation for the discrete Weibull distribution (type I) with parameters q and β .

Usage

```
ddweibull( x, q = exp( -1 ), beta = 1, zero = TRUE )
pdweibull( x, q = exp( -1 ), beta = 1, zero = TRUE )
qdweibull( p, q = exp( -1 ), beta = 1, zero = TRUE )
rdweibull( n, q = exp( -1 ), beta = 1, zero = TRUE )
```

Arguments

x	vector of quantiles.
p	vector of probabilities.
q, beta	shape and scale parameters, the latter defaulting to 1.
zero	logical; if TRUE (default), the support contains 0; FALSE otherwise.
n	number of observations. If $\text{length}(n) > 1$, the length is taken to be the number required.

Details

The discrete Weibull distribution has density given by

$$f(x) = q^{x^\beta} - q^{(x+1)^\beta}, x = 0, 1, 2, \dots$$

And for the case zero = FALSE:

$$f(x) = q^{(x-1)^\beta} - q^{x^\beta}, x = 1, 2, \dots$$

Cumulative distribution function

$$F(x) = 1 - q^{(x+1)^\beta}$$

And for the case zero = FALSE, x+1 should be replaced by x.

Value

ddweibull gives the density, pdweibull gives the distribution function, qdweibull gives the quantile function, and rdweibull generates random values.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>, Pariya Behrouzi, Veronica Vinciotti

References

Nakagawa, T. and Osaki, S. (1975). The Discrete Weibull Distribution. *IEEE Transactions on Reliability*, R-24, 300-301.

Examples

```
n <- 1000
q <- 0.4
beta <- 0.8

rdw <- rdweibull( n = n, q = q, beta = beta )

plot( prop.table( table( rdw ) ), type = "h", col = "gray50" )

x <- 0:max( rdw )

lines( x, ddweibull( x = x, q = q, beta = beta ), type = "o", col = "blue", lwd = 2 )

hist( pdweibull( x = rdw, q = q, beta = beta ) )

plot( ecdf( rdw ) )
lines( x, pdweibull( x, q = q, beta = beta ), col = "blue", lwd = 2, type = "s" )
```

geneExpression	<i>Human gene expression dataset</i>
----------------	--------------------------------------

Description

The dataset contains human gene expression of 100 transcripts (with unique Illumina TargetID) measured on 60 unrelated individuals.

Usage

```
data( geneExpression )
```

Format

The format is a matrix with 60 rows (number of individuals) and 100 column (number of transcripts).

References

Bhadra, A. and Mallick, B. K. (2013). Joint High Dimensional Bayesian Variable and Covariance Selection with an Application to eQTL Analysis, *Biometrics*, 69(2):447-457.

Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138.

Examples

```
data( geneExpression )
```

```
dim( geneExpression )
```

```
head( geneExpression )
```

gnorm	<i>Normalizing constant for G-Wishart</i>
-------	---

Description

Calculates log of the normalizing constant of G-Wishart distribution based on the Monte Carlo method, developed by Atay-Kayis and Massam (2005).

Usage

```
gnorm( adj, b = 3, D = diag( ncol( adj ) ), iter = 100 )
```


Arguments

adj	The adjacency matrix corresponding to the graph structure. It is an upper triangular matrix in which $a_{ij} = 1$ if there is a link between nodes i and j , otherwise $a_{ij} = 0$.
b	The degree of freedom for G-Wishart distribution, $W_G(b, D)$.
D	The positive definite ($p \times p$) "scale" matrix for G-Wishart distribution, $W_G(b, D)$. The default is an identity matrix.
iter	The number of iteration for the Monte Carlo approximation.

Details

Log of the normalizing constant approximation using Monte Carlo method for a G-Wishart distribution, $K \sim W_G(b, D)$, with density:

$$Pr(K) = \frac{1}{I(b, D)} |K|^{(b-2)/2} \exp \left\{ -\frac{1}{2} \text{trace}(K \times D) \right\}.$$

Value

Log of the normalizing constant of G-Wishart distribution.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

- Atay-Kayis, A. and Massam, H. (2005). A monte carlo method for computing the marginal likelihood in nondecomposable Gaussian graphical models, *Biometrika*, 92(2):317-335
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Uhler, C., et al (2018) Exact formulas for the normalizing constants of Wishart distributions for graphical models, *The Annals of Statistics* 46(1):90-118
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138

See Also

[rgwish](#), [rwish](#)

Examples

```
## Not run:
# adj: adjacency matrix of graph with 3 nodes and 2 links
adj <- matrix( c( 0, 0, 1,
                0, 0, 1,
                0, 0, 0 ), 3, 3, byrow = TRUE )
```

```
gnorm( adj, b = 3, D = diag( 3 ) )
## End(Not run)
```

graph.sim

Graph simulation

Description

Simulating undirected graph structures, including "random", "cluster", "scale-free", "lattice", "hub", "star", and "circle".

Usage

```
graph.sim( p = 10, graph = "random", prob = 0.2, size = NULL, class = NULL, vis = FALSE )
```

Arguments

p	The number of variables (nodes).
graph	The undirected graph with options "random", "cluster", "scale-free", "lattice", "hub", "star", and "circle". It also could be an adjacency matrix corresponding to a graph structure (an upper triangular matrix in which $g_{ij} = 1$ if there is a link between nodes i and j , otherwise $g_{ij} = 0$).
prob	If graph="random", it is the probability that a pair of nodes has a link.
size	The number of links in the true graph (graph size).
class	If graph="cluster", it is the number of classes.
vis	Visualize the true graph structure.

Value

The adjacency matrix corresponding to the simulated graph structure, as an object with S3 class "graph".

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30

Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138

Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*

Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845

Pensar, J. et al (2017) Marginal pseudo-likelihood learning of discrete Markov network structures, *Bayesian Analysis*, 12(4):1195-215

See Also

[bdgraph.sim](#), [bdgraph](#), [bdgraph.mpl](#)

Examples

```
# Generating a 'hub' graph
adj <- graph.sim( p = 8, graph = "scale-free" )

plot( adj )

adj
```

link2adj

Extract links from an adjacency matrix

Description

Extract links from an adjacency matrix or an object of class "sim" from function [bdgraph.sim](#) and "graph" from function [graph.sim](#).

Usage

```
link2adj( link, p = NULL )
```

Arguments

link	An $(2 \times p)$ matrix or a <code>data.frame</code> corresponding to the links from the graph structure.
p	The number of nodes of the graph.

Value

An adjacency matrix corresponding to a graph structure in which $a_{ij} = 1$ if there is a link between nodes i and j , otherwise $a_{ij} = 0$.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845
- Pensar, J. et al (2017) Marginal pseudo-likelihood learning of discrete Markov network structures, *Bayesian Analysis*, 12(4):1195-215

See Also

[adj2link](#), [graph.sim](#)

Examples

```
# Generating a 'random' graph
adj <- graph.sim( p = 6, vis = TRUE )

link <- adj2link( adj )

link2adj( link, p = 6 )
```

pgraph

Posterior probabilities of the graphs

Description

Provides the estimated posterior probabilities for the most likely graphs or a specific graph.

Usage

```
pgraph( bdgraph.obj, number.g = 4, adj = NULL )
```

Arguments

- | | |
|-------------|--|
| bdgraph.obj | An object of S3 class "bdgraph", from function bdgraph . |
| number.g | The number of graphs with the highest posterior probabilities to be shown. This option is ignored if 'adj' is specified. |
| adj | An adjacency matrix corresponding to a graph structure. It is an upper triangular matrix in which $a_{ij} = 1$ if there is a link between notes i and j , otherwise $a_{ij} = 0$. It also can be an object of S3 class "sim", from function bdgraph.sim . |

Value

selected_g	the adjacency matrices which corresponding to the graphs with the highest posterior probabilities.
prob_g	A vector of the posterior probabilities of the graphs corresponding to 'selected_g'.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Ernst Wit

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

See Also

[bdgraph](#), [bdgraph.mpl](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 50, p = 6, size = 6, vis = TRUE )

bdgraph.obj <- bdgraph( data = data.sim, save = TRUE )

# Estimated posterior probability of the true graph
pgraph( bdgraph.obj, adj = data.sim )

# Estimated posterior probability of first and second graphs with highest probabilities
pgraph( bdgraph.obj, number.g = 2 )

## End(Not run)
```

plinks *Estimated posterior link probabilities*

Description

Provides the estimated posterior link probabilities for all possible links in the graph.

Usage

```
plinks( bdgraph.obj, round = 2, burnin = NULL )
```

Arguments

bdgraph.obj	An object of S3 class "bdgraph", from function bdgraph . It also can be an object of S3 class "ssgraph", from the function ssgraph::ssgraph() of R package ssgraph::ssgraph() .
round	A value for rounding all probabilities to the specified number of decimal places.
burnin	The number of burn-in iteration to scape.

Value

An upper triangular matrix which corresponds the estimated posterior probabilities for all possible links.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Ernst Wit

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

See Also

[bdgraph](#), [bdgraph.mpl](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'circle' graph
data.sim <- bdgraph.sim( n = 70, p = 6, graph = "circle", vis = TRUE )

bdgraph.obj <- bdgraph( data = data.sim, iter = 10000 )

plinks( bdgraph.obj, round = 2 )

## End(Not run)
```

plot.bdgraph *Plot function for S3 class "bdgraph"*

Description

Visualizes structure of the selected graphs which could be a graph with links for which their estimated posterior probabilities are greater than 0.5 or graph with the highest posterior probability.

Usage

```
## S3 method for class 'bdgraph'
plot( x, cut = 0.5, number.g = NULL, ... )
```

Arguments

x	An object of S3 class "bdgraph", from function bdgraph .
cut	Threshold for including the links in the selected graph based on the estimated posterior probabilities of the links; See the examples.
number.g	The number of graphs with the highest probabilities. This option works for the case running function <code>bdgraph()</code> with option <code>save = TRUE</code> ; See the examples.
...	System reserved (no specific usage).

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Ernst Wit

References

Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30

Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138

Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*

Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845

Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

See Also

[bdgraph](#), [bdgraph.mpl](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 50, p = 6, size = 7, vis = TRUE )

bdgraph.obj <- bdgraph( data = data.sim )

plot( bdgraph.obj )

bdgraph.obj <- bdgraph( data = data.sim, save = TRUE )

plot( bdgraph.obj, number.g = 4 )

plot( bdgraph.obj, cut = 0.4 )

## End(Not run)
```

plot.graph

Plot function for S3 class "graph"

Description

Visualizes structure of the graph.

Usage

```
## S3 method for class 'graph'
plot( x, cut = 0.5, mode = "undirected", diag = FALSE, main = NULL,
      vertex.color = "white", vertex.label.color = 'black', ... )
```

Arguments

x	An object of S3 class "graph", from function graph.sim .
cut	This option is for the case where input 'x' is the object of class "bdgraph" or "ssgraph". Threshold for including the links in the selected graph based on the estimated posterior probabilities of the links.
mode	Type of graph which is according to R package igraph .
diag	Logical which is according to R package igraph .

Arguments

x An object of S3 class "sim", from function [bdgraph.sim](#).
 ... System reserved (no specific usage).

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Ernst Wit

References

Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30

Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138

Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*

Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845

Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

See Also

[graph.sim](#), [bdgraph.sim](#)

Examples

```
# Generating synthetic multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 10, p = 15 )

plot( data.sim )
```

plotcoda

Convergence plot

Description

Visualizes the cumulative occupancy fractions of all possible links in the graph. It can be used for monitoring the convergence of the sampling algorithms, BDMCMC and RJMCMC.

Usage

```
plotcoda( bdgraph.obj, thin = NULL, control = TRUE, main = NULL, ... )
```

Arguments

<code>bdgraph.obj</code>	An object of S3 class "bdgraph", from function <code>bdgraph</code> . It also can be an object of S3 class "ssgraph", from the function <code>ssgraph::ssgraph()</code> of R package <code>ssgraph::ssgraph()</code> .
<code>thin</code>	An option for getting fast result for a cumulative plot according to part of the iteration.
<code>control</code>	Logical: if TRUE (default) and the number of nodes is greater than 15, then 100 links randomly is selected for visualization.
<code>main</code>	Graphical parameter (see plot).
<code>...</code>	System reserved (no specific usage).

Details

Note that a spending time for this function depends on the number of nodes.

For fast result, you can choose bigger value for the 'thin' option.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

See Also

[bdgraph](#), [bdgraph.mpl](#), [traceplot](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'circle' graph
data.sim <- bdgraph.sim( n = 50, p = 6, graph = "circle", vis = TRUE )

bdgraph.obj <- bdgraph( data = data.sim, iter = 10000, burnin = 0 , save = TRUE )

plotcoda( bdgraph.obj )

## End(Not run)
```

plotroc

*ROC plot***Description**

Draws the receiver operating characteristic (ROC) curve according to the true graph structure for object of S3 class "bdgraph", from function [bdgraph](#).

Usage

```
plotroc( target, est, est2 = NULL, est3 = NULL, est4 = NULL,
         cut = 20, smooth = FALSE, label = TRUE, main = "ROC Curve" )
```

Arguments

target	An adjacency matrix corresponding to the true graph structure in which $a_{ij} = 1$ if there is a link between nodes i and j , otherwise $a_{ij} = 0$. It can be an object with S3 class "sim" from function bdgraph.sim . It can be an object with S3 class "graph" from function graph.sim .
est, est2, est3, est4	An upper triangular matrix corresponding to the estimated posterior probabilities for all possible links. It can be an object with S3 class "bdgraph" from function bdgraph . It can be an object of S3 class "ssgraph", from the function ssgraph::ssgraph() of R package ssgraph::ssgraph() . It can be an object of S3 class "select", from the function huge.select of R package huge . Options est2, est3 and est4 are for comparing two or more different approaches.
cut	Number of cut points.
smooth	Logical: for smoothing the ROC curve.
label	Logical: for adding legend to the ROC plot.
main	An overall title for the plot.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30

Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138

Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*

Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845

Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

See Also

[bdgraph](#), [bdgraph.mpl](#), [compare](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 30, p = 6, size = 7, vis = TRUE )

# Runing sampling algorithm
bdgraph.obj <- bdgraph( data = data.sim, iter = 10000 )
# Comparing the results
plotroc( data.sim, bdgraph.obj )

# To compare the results based on CGGMs approach
bdgraph.obj2 <- bdgraph( data = data.sim, method = "gcgm", iter = 10000 )
# Comparing the resultss
plotroc( data.sim, bdgraph.obj, bdgraph.obj2, label = FALSE )
legend( "bottomright", c( "GGMs", "GCGMs" ), lty = c( 1, 2 ), col = c( "black", "red" ) )

## End(Not run)
```

precision

Estimated precision matrix

Description

Provides the estimated precision matrix.

Usage

```
precision( bdgraph.obj, round = 2 )
```

Arguments

bdgraph.obj An object of S3 class "bdgraph", from function [bdgraph](#). It also can be an object of S3 class "ssgraph", from the function [ssgraph::ssgraph\(\)](#) of R package [ssgraph::ssgraph\(\)](#).

round A value for rounding all probabilities to the specified number of decimal places.

Value

A matrix which corresponds the estimated precision matrix.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30

Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138

Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*

Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845

Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

See Also

[bdgraph](#), [covariance](#), [plinks](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'circle' graph
data.sim <- bdgraph.sim( n = 70, p = 6, graph = "circle", vis = TRUE )

bdgraph.obj <- bdgraph( data = data.sim )

precision( bdgraph.obj ) # Estimated precision matrix

data.sim $ K # True precision matrix

## End(Not run)
```

```
print.bdgraph
```

```
Print function for S3 class "bdgraph"
```

Description

Prints the information about the selected graph which could be a graph with links for which their estimated posterior probabilities are greater than 0.5 or graph with the highest posterior probability. It provides adjacency matrix, size and posterior probability of the selected graph.

Usage

```
## S3 method for class 'bdgraph'
print( x, ... )
```

Arguments

x An object of S3 class "bdgraph", from function [bdgraph](#).
 ... System reserved (no specific usage).

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Ernst Wit

References

Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30

Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138

Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*

Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845

Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

See Also

[bdgraph](#), [bdgraph.mpl](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 50, p = 6, vis = TRUE )

bdgraph.obj <- bdgraph( data = data.sim )

print( bdgraph.obj )

## End(Not run)
```

print.sim

Print function for S3 class "sim"

Description

Prints the information about the type of data, the sample size, the graph type, the number of nodes, number of links and sparsity of the true graph.

Usage

```
## S3 method for class 'sim'  
print( x, ... )
```

Arguments

x An object of S3 class "sim", from function [bdgraph.sim](#).
... System reserved (no specific usage).

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Ernst Wit

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*

See Also

[graph.sim](#), [bdgraph.sim](#)

Examples

```
# Generating multivariate normal data from a 'random' graph  
data.sim <- bdgraph.sim( n = 20, p = 10, vis = TRUE )  
  
print( data.sim )
```

reinis

Risk factors of coronary heart disease

Description

The dataset consist of 6 discrete variables as the potential risk factors of coronary heart disease. The data collected from 1841 men employed of a car factory in Czechoslovakia (Reinis et al. 1981).

Usage

```
data( reinis )
```

Format

The format is a matrix with 1841 rows (number of individuals) and 6 column (number of variables).

References

Edwards and Havranek (1985). A fast procedure for model search in multidimensional contingency tables, *Biometrika*, 72:339-351

Reinis et al (1981). Prognostic significance of the risk profile in the prevention of coronary heart disease, *Bratis. lek. Listy*, 76:137-150

Mohammadi, A. and Dobra, A. (2017). The R Package **BDgraph** for Bayesian Structure Learning in Graphical Models, *ISBA Bulletin*, 24(4):11-16

Examples

```
data( reinis )
```

```
summary( reinis )
```

 rgwish

Sampling from G-Wishart distribution

Description

Generates random matrices, distributed according to the G-Wishart distribution with parameters b and D , $W_G(b, D)$ with respect to the graph structure G . Note this function works for both non-decomposable and decomposable graphs.

Usage

```
rgwish( n = 1, adj = NULL, b = 3, D = NULL, threshold = 1e-8 )
```

Arguments

n	The number of samples required.
adj	The adjacency matrix corresponding to the graph structure which can be non-decomposable or decomposable. It should be an upper triangular matrix in which $a_{ij} = 1$ if there is a link between nodes i and j , otherwise $a_{ij} = 0$. adj could be an object of class "graph", from function graph.sim . It also could be an object of class "sim", from function bdgraph.sim . It also could be an object of class "bdgraph", from functions bdgraph.mpl or bdgraph .
b	The degree of freedom for G-Wishart distribution, $W_G(b, D)$.
D	The positive definite ($p \times p$) "scale" matrix for G-Wishart distribution, $W_G(b, D)$. The default is an identity matrix.
threshold	The threshold value for the convergence of sampling algorithm from G-Wishart.

Details

Sampling from G-Wishart distribution, $K \sim W_G(b, D)$, with density:

$$Pr(K) \propto |K|^{(b-2)/2} \exp \left\{ -\frac{1}{2} \text{trace}(K \times D) \right\},$$

which $b > 2$ is the degree of freedom and D is a symmetric positive definite matrix.

Value

A numeric array, say A, of dimension $(p \times p \times n)$, where each $A[, , i]$ is a positive definite matrix, a realization of the G-Wishart distribution, $W_G(b, D)$. Note, for the case $n = 1$, the output is a matrix.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

- Lenkoski, A. (2013). A direct sampler for G-Wishart variates, *Stat*, 2:119-128
- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

See Also

[gnorm](#), [rwish](#)

Examples

```
# Generating a 'circle' graph as a non-decomposable graph
adj <- graph.sim( p = 5, graph = "circle" )
adj   # adjacency of graph with 5 nodes

sample <- rgwish( n = 1, adj = adj, b = 3, D = diag( 5 ) )
round( sample, 2 )

sample <- rgwish( n = 5, adj = adj )
round( sample, 2 )
```

`rmvnorm`*Generate data from the multivariate Normal distribution*

Description

Random generation function from the multivariate Normal distribution with mean equal to *mean* and covariance matrix *sigma*.

Usage

```
rmvnorm( n = 10, mean = rep( 0, length = ncol( sigma ) ),
         sigma = diag( length( mean ) ) )
```

Arguments

<code>n</code>	Number of observations.
<code>mean</code>	Mean vector, default is <code>rep(0, length = ncol(sigma))</code> .
<code>sigma</code>	positive definite covariance matrix, default is <code>diag(length(mean))</code> .

Value

A numeric matrix with rows equal to *n* and columns equal to `length(mean)`.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

See Also

[bdgraph.sim](#), [rwish](#), [rgwish](#)

Examples

```
mean <- c( 5, 20 )
sigma <- matrix( c( 4, 2,
                  2, 5 ), 2, 2 ) # covariance matrix

sample <- rmvnorm( n = 500, mean = mean, sigma = sigma )
plot( sample )
```

roc

*Build a ROC curve***Description**

This function builds a ROC curve specifically for graph structure learning and returns a “roc” object, a list of class “roc”. This object can be printed, plotted, or passed to the functions `pROC::roc()`, `pROC::ci()`, `pROC::smooth.roc()` and `pROC::coords()`. Additionally, two roc objects can be compared with `pROC::roc.test()`. This function is based on the `roc` function of R package `pROC`.

Usage

```
roc( pred, actual, auc = TRUE, smooth = FALSE, plot = FALSE, ... )
```

Arguments

pred	An adjacency matrix corresponding to an estimated graph. It can be an object with S3 class “bdgraph” from function <code>bdgraph</code> . It can be an object of S3 class “ssgraph”, from the function <code>ssgraph::ssgraph()</code> of R package <code>ssgraph::ssgraph()</code> . It can be a <code>numeric</code> or <code>ordered</code> vector of the same length than <code>actual</code> , containing the predicted value of each observation.
actual	An adjacency matrix corresponding to the actual graph structure in which $a_{ij} = 1$ if there is a link between notes i and j , otherwise $a_{ij} = 0$. It can be an object with S3 class “sim” from function <code>bdgraph.sim</code> . It can be an object with S3 class “graph” from function <code>graph.sim</code> . It can be a factor, numeric or character vector of responses (true class), typically encoded with 0 (controls) and 1 (cases). Only two classes can be used in a ROC curve.
smooth	if TRUE, the ROC curve is passed to <code>smooth</code> to be smoothed.
auc	compute the area under the curve (AUC)? If TRUE (default), additional arguments can be passed to <code>pROC::auc()</code> .
plot	plot the ROC curve? If TRUE, additional arguments can be passed to <code>pROC::plot.roc()</code> .
...	further arguments to be passed to <code>pROC::roc()</code> .

Value

If the data contained any NA value and `na.rm=FALSE`, NA is returned. Otherwise, if `smooth=FALSE`, a list of class “roc” with the following fields:

auc	if called with <code>auc=TRUE</code> , a numeric of class “auc” as defined in <code>pROC::auc()</code> .
ci	if called with <code>ci=TRUE</code> , a numeric of class “ci” as defined in <code>pROC::ci()</code> .
response	the response vector. Patients whose response is not <code>%in%</code> levels are discarded. If NA values were removed, a <code>na.action</code> attribute similar to <code>na.omit</code> stores the row numbers.

predictor	the predictor vector converted to numeric as used to build the ROC curve. Patients whose response is not %in% levels are discarded. If NA values were removed, a <code>na.action</code> attribute similar to <code>na.omit</code> stores the row numbers.
<code>original.predictor</code> , <code>original.response</code>	the response and predictor vectors as passed in argument.
levels	the levels of the response as defined in argument.
controls	the predictor values for the control observations.
cases	the predictor values for the cases.
percent	if the sensitivities, specificities and AUC are reported in percent, as defined in argument.
direction	the direction of the comparison, as defined in argument.
<code>fun.sesp</code>	the function used to compute sensitivities and specificities. Will be re-used in bootstrap operations.
sensitivities	the sensitivities defining the ROC curve.
specificities	the specificities defining the ROC curve.
thresholds	the thresholds at which the sensitivities and specificities were computed. See below for details.
call	how the function was called. See <code>match.call</code> for more details.

If `smooth=TRUE` a list of class “smooth.roc” as returned by `pROC::smooth()`, with or without additional elements `auc` and `ci` (according to the call).

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

Tom Fawcett (2006) “An introduction to ROC analysis”. *Pattern Recognition Letters* **27**, 861–874. DOI: doi: [10.1016/j.patrec.2005.10.010](https://doi.org/10.1016/j.patrec.2005.10.010).

Xavier Robin, Natacha Turck, Alexandre Hainard, *et al.* (2011) “pROC: an open-source package for R and S+ to analyze and compare ROC curves”. *BMC Bioinformatics*, **7**, 77. DOI: doi: [10.1186/147121051277](https://doi.org/10.1186/147121051277).

See Also

`pROC::auc()`, `pROC::plot.roc()`, `pROC::print.roc()`, `bdgraph`

Examples

```
## Not run:
set.seed( 100 )

# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 50, p = 6, size = 7, vis = TRUE )

# Running sampling algorithm based on GGMS
```

```

sample.ggm <- bdgraph( data = data.sim, method = "ggm", iter = 10000 )

# ROC curve for GGM method
roc.ggm <- roc( pred = sample.ggm, actual = data.sim, plot = TRUE )

# Running sampling algorithm based on GCGMs
sample.gcgm <- bdgraph( data = data.sim, method = "gcgm", iter = 10000 )

# ROC curve for GGM and GCGM methods
roc.gcgm <- roc( pred = sample.gcgm, actual = data.sim, plot = TRUE )

ggroc( list( roc.ggm = roc.ggm, roc.gcgm = roc.gcgm ) )

## End(Not run)

```

 rwish

Sampling from Wishart distribution

Description

Generates random matrices, distributed according to the Wishart distribution with parameters b and D , $W(b, D)$.

Usage

```
rwish( n = 1, p = 2, b = 3, D = diag( p ) )
```

Arguments

n	The number of samples required.
p	The number of variables (nodes).
b	The degree of freedom for Wishart distribution, $W(b, D)$.
D	The positive definite ($p \times p$) "scale" matrix for Wishart distribution, $W(b, D)$. The default is an identity matrix.

Details

Sampling from Wishart distribution, $K \sim W(b, D)$, with density:

$$Pr(K) \propto |K|^{(b-2)/2} \exp \left\{ -\frac{1}{2} \text{trace}(K \times D) \right\},$$

which $b > 2$ is the degree of freedom and D is a symmetric positive definite matrix.

Value

A numeric array, say A , of dimension $(p \times p \times n)$, where each $A[, , i]$ is a positive definite matrix, a realization of the Wishart distribution $W(b, D)$. Note, for the case $n = 1$, the output is a matrix.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

- Lenkoski, A. (2013). A direct sampler for G-Wishart variates, *Stat*, 2:119-128
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30

See Also

[gnorm](#), [rgwish](#)

Examples

```
sample <- rwish( n = 3, p = 5, b = 3, D = diag( 5 ) )
round( sample, 2 )
```

select

Graph selection

Description

Provides the selected graph which, based on input, could be a graph with links for which their estimated posterior probabilities are greater than 0.5 (default) or a graph with the highest posterior probability; see examples.

Usage

```
select( bdgraph.obj, cut = NULL, vis = FALSE )
```

Arguments

- `bdgraph.obj` A matrix in which each element response to the weight of the links. It can be an object of S3 class "bdgraph", from function `bdgraph`. It can be an object of S3 class "ssgraph", from the function `ssgraph::ssgraph()` of R package `ssgraph::ssgraph()`.
- `cut` Threshold for including the links in the selected graph based on the estimated posterior probabilities of the links; see the examples.
- `vis` Visualize the selected graph structure.

Value

An adjacency matrix corresponding to the selected graph.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Ernst Wit

References

Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30

Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138

Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*

Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845

Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

See Also

[bdgraph](#), [bdgraph.mpl](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 50, p = 6, size = 7, vis = TRUE )

bdgraph.obj <- bdgraph( data = data.sim )

select( bdgraph.obj )

bdgraph.obj <- bdgraph( data = data.sim, save = TRUE )

select( bdgraph.obj )

select( bdgraph.obj, cut = 0.5, vis = TRUE )

## End(Not run)
```

`sparsity`*Compute the sparsity of a graph*

Description

Compute the sparsity of a graph (or a network) or an object of classes "sim" from function `bdgraph.sim` and "graph" from function `graph.sim`.

Usage

```
sparsity( adj )
```

Arguments

`adj` An adjacency matrix corresponding to a graph structure in which $a_{ij} = 1$ if there is a link between nodes i and j , otherwise $a_{ij} = 0$. It can be an object with S3 class "sim" from function `bdgraph.sim`. It can be an object with S3 class "graph" from function `graph.sim`.

Value

A value corresponding to the sparsity of a graph.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl>

References

Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30

See Also

`graph.sim`, `adj2link`, `link2adj`

Examples

```
# Generating a 'random' graph
adj <- graph.sim( p = 10, vis = TRUE )

sparsity( adj )
```

summary.bdgraph	<i>Summary function for S3 class "bdgraph"</i>
-----------------	--

Description

Provides a summary of the results for function `bdgraph`.

Usage

```
## S3 method for class 'bdgraph'
summary( object, round = 2, vis = TRUE, ... )
```

Arguments

object	An object of S3 class "bdgraph", from function <code>bdgraph</code> .
round	A value for rounding all probabilities to the specified number of decimal places.
vis	Visualize the results.
...	System reserved (no specific usage).

Value

selected_g	The adjacency matrix corresponding to the selected graph which has the highest posterior probability.
p_links	An upper triangular matrix corresponding to the posterior probabilities of all possible links.
K_hat	The estimated precision matrix.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Ernst Wit

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Letac, G., Massam, H. and Mohammadi, R. (2018). The Ratio of Normalizing Constants for Bayesian Graphical Gaussian Model Selection, *arXiv preprint arXiv:1706.04416v2*
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645

See Also

[bdgraph](#), [bdgraph.mpl](#)

Examples

```
## Not run:  
# Generating multivariate normal data from a 'random' graph  
data.sim <- bdgraph.sim( n = 50, p = 6, size = 7, vis = TRUE )  
  
bdgraph.obj <- bdgraph( data = data.sim )  
  
summary( bdgraph.obj )  
  
bdgraph.obj <- bdgraph( data = data.sim, save = TRUE )  
  
summary( bdgraph.obj )  
  
summary( bdgraph.obj, vis = FALSE )  
  
## End(Not run)
```

surveyData

Labor force survey data

Description

The survey dataset concerns 1002 males in the U.S labor force, described by Hoff (2007). The seven observed variables which have been measured on various scales are as follow: the income (income), degree (degree), the number of children (children), parents income (pincome), parents degree (pdegree), number of parents children (pchildren), and age (age).

Usage

```
data( surveyData )
```

Format

The format is a matrix with 1002 rows (number of individuals) and 7 column (number of variables).

References

Hoff, P. (2007). Extending the rank likelihood for semiparametric copula estimation, *The Annals of Applied Statistics*, 1(1), 265-283.

Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30

Examples

```
data( surveyData )

summary( surveyData )
```

traceplot	<i>Trace plot of graph size</i>
-----------	---------------------------------

Description

Trace plot for graph size for the objects of S3 class "bdgraph", from function [bdgraph](#). It is a tool for monitoring the convergence of the sampling algorithms, BDMCMC and RJMCMC.

Usage

```
traceplot ( bdgraph.obj, acf = FALSE, pacf = FALSE, main = NULL, ... )
```

Arguments

bdgraph.obj	An object of S3 class "bdgraph", from function bdgraph . It also can be an object of S3 class "ssgraph", from the function ssgraph::ssgraph() of R package ssgraph::ssgraph() .
acf	Visualize the autocorrelation functions for graph size.
pacf	Visualize the partial autocorrelations for graph size.
main	Graphical parameter (see plot).
...	System reserved (no specific usage).

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Ernst Wit

References

- Mohammadi, R. and Wit, E. C. (2019). **BDgraph**: An R Package for Bayesian Structure Learning in Graphical Models, *Journal of Statistical Software*, 89(3):1-30
- Mohammadi, A. and Wit, E. C. (2015). Bayesian Structure Learning in Sparse Gaussian Graphical Models, *Bayesian Analysis*, 10(1):109-138
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845
- Mohammadi, A. et al (2017). Bayesian modelling of Dupuytren disease by using Gaussian copula graphical models, *Journal of the Royal Statistical Society: Series C*, 66(3):629-645
- Mohammadi, A. and Dobra, A. (2017). The R Package **BDgraph** for Bayesian Structure Learning in Graphical Models, *ISBA Bulletin*, 24(4):11-16

See Also

[plotcoda](#), [bdgraph](#), [bdgraph.mpl](#)

Examples

```
## Not run:
# Generating multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 50, p = 6, size = 7, vis = TRUE )

bdgraph.obj <- bdgraph( data = data.sim, iter = 10000, burnin = 0, save = TRUE )

traceplot( bdgraph.obj )

traceplot( bdgraph.obj, acf = TRUE, pacf = TRUE )

## End(Not run)
```

transfer

transfer for count data

Description

Transfers count data, by counting the duplicated rows.

Usage

```
transfer( r_data )
```

Arguments

`r_data` An $(n \times p)$ matrix or a data frame corresponding to the data (n is the sample size and p is the number of variables).

Value

An $(n \times p + 1)$ matrix of transferred data, in which the last column is the frequency of duplicated rows.

Author(s)

Reza Mohammadi <a.mohammadi@uva.nl> and Adrian Dobra

References

- Mohammadi, A. and Dobra, A. (2017). The R Package **BDgraph** for Bayesian Structure Learning in Graphical Models, *ISBA Bulletin*, 24(4):11-16
- Dobra, A. and Mohammadi, R. (2018). Loglinear Model Selection and Human Mobility, *Annals of Applied Statistics*, 12(2):815-845

See Also

[bdgraph.mpl](#), [bdgraph.sim](#)

Examples

```
# Generating multivariate binary data from a 'random' graph
data.sim <- bdgraph.sim( n = 12, p = 4, size = 4, type = "count", cut = 2 )
r_data   <- data.sim $ data
r_data

# Transfer the data
transfer( r_data )
```

Index

- * **Wishart**
 - gnorm, 24
 - rgwish, 41
 - rwish, 46
- * **data preprocessing**
 - bdgraph.npn, 10
 - transfer, 53
- * **datagen**
 - bdgraph.sim, 12
- * **datasets**
 - churn, 15
 - geneExpression, 24
 - reinis, 40
 - surveyData, 51
- * **distribution**
 - Discrete Weibull, 22
 - rgwish, 41
 - rmvnorm, 43
 - rwish, 46
- * **graphs**
 - adj2link, 4
 - graph.sim, 26
 - link2adj, 27
 - sparsity, 49
- * **hplot**
 - plot.bdgraph, 31
 - plot.graph, 32
 - plot.sim, 33
 - plotcoda, 34
 - plotroc, 36
 - traceplot, 52
- * **iteration**
 - bdgraph, 5
 - bdgraph.mpl, 8
 - gnorm, 24
- * **package**
 - BDgraph-package, 2
- * **parameter learning**
 - covariance, 21
 - precision, 37
- * **print**
 - print.bdgraph, 38
 - print.sim, 39
- * **sampling algorithms**
 - bdgraph, 5
 - bdgraph.mpl, 8
- * **simulation**
 - bdgraph.sim, 12
 - graph.sim, 26
- * **structure learning**
 - bdgraph, 5
 - bdgraph.mpl, 8
 - bf, 14
 - compare, 17
 - conf.mat, 18
 - conf.mat.plot, 20
 - pgraph, 28
 - plinks, 30
 - roc, 44
 - select, 47
 - %in%, 44, 45
- calc_joint_dist (BDgraph-package), 2
- compute_measures (BDgraph-package), 2
- compute_tp_fp (BDgraph-package), 2
- detect_cores (BDgraph-package), 2
- generate_clique_factors (BDgraph-package), 2
- get_cores (BDgraph-package), 2
- get_g_prior (BDgraph-package), 2
- get_g_start (BDgraph-package), 2
- get_graph (BDgraph-package), 2
- get_K_start (BDgraph-package), 2
- get_S_n_p (BDgraph-package), 2
- global_hc (BDgraph-package), 2
- global_hc_binary (BDgraph-package), 2
- hill_climb_mpl (BDgraph-package), 2
- hill_climb_mpl_binary (BDgraph-package), 2

- local_mb_hc (BDgraph-package), 2
- local_mb_hc_binary (BDgraph-package), 2
- log_mpl_binary (BDgraph-package), 2
- log_mpl_discrete (BDgraph-package), 2
- sample_ug (BDgraph-package), 2
- adj2link, 4, 28, 49
- BDgraph, 6, 9
- BDgraph (BDgraph-package), 2
- bdgraph, 3, 5, 10, 11, 13–15, 17–22, 27–32, 35–39, 41, 44, 45, 47, 48, 50–53
- BDgraph-package, 2
- bdgraph.mpl, 3, 7, 8, 11, 13, 15, 18, 27, 29, 30, 32, 35, 37, 39, 41, 48, 51, 53, 54
- bdgraph.npn, 10
- bdgraph.sim, 3, 4, 6–8, 10, 11, 12, 14, 15, 17–20, 27, 28, 33, 34, 36, 40, 41, 43, 44, 49, 54
- bf, 14
- churn, 15
- compare, 3, 7, 10, 15, 17, 19, 20, 37
- conf.mat, 18, 20
- conf.mat.plot, 19, 20
- covariance, 21, 38
- ddweibull (Discrete Weibull), 22
- Discrete Weibull, 22
- geneExpression, 24
- gnorm, 24, 42, 47
- graph.sim, 4, 5, 13, 14, 17, 19, 20, 26, 27, 28, 32–34, 36, 40, 41, 44, 49
- huge, 17, 36
- huge.select, 17, 36
- igraph, 32, 33
- link2adj, 5, 27, 49
- match.call, 45
- na.omit, 44, 45
- numeric, 44
- ordered, 44
- pdweibull (Discrete Weibull), 22
- pgraph, 28
- plinks, 22, 30, 38
- plot.bdgraph, 31
- plot.graph, 32
- plot.sim, 33
- plotcoda, 34, 53
- plotroc, 18, 36
- precision, 22, 37
- print.bdgraph, 38
- print.sim, 39
- pROC::auc(), 44, 45
- pROC::ci(), 44
- pROC::coords(), 44
- pROC::plot.roc(), 44, 45
- pROC::print.roc(), 45
- pROC::roc(), 44
- pROC::roc.test(), 44
- pROC::smooth(), 45
- pROC::smooth.roc(), 44
- qdweibull (Discrete Weibull), 22
- rdweibull (Discrete Weibull), 22
- reinis, 40
- rgwish, 3, 25, 41, 43, 47
- rmvnorm, 43
- roc, 19, 20, 44, 44
- rwish, 25, 42, 43, 46
- select, 47
- smooth, 44
- sparsity, 49
- ssgraph::ssgraph(), 6, 9, 14, 17, 19–21, 30, 35–37, 44, 47, 52
- summary.bdgraph, 7, 10, 50
- surveyData, 51
- traceplot, 35, 52
- transfer, 53