

Package ‘BDgraph’

September 24, 2014

Type Package

Title Graph estimation based on birth-death MCMC approach

Version 2.12

Date 2013-02-04

Author Abdolreza Mohammadi and Ernst Wit

Depends R (>= 3.0.0), Matrix, igraph

Maintainer Abdolreza Mohammadi <a.mohammadi@rug.nl>

Description This package provides a Bayesian methodology for structure learning in undirected graphical models. Our Bayesian methodology is based on birth-death Markov chain Monte Carlo (BDMCMC) algorithm which is the main function with the name 'bdgraph'. The main target of this package is high-dimensional data analysis wherein usually $p \gg n$. The computation is memory-optimized using the sparse matrix output.

Acknowledgements The help of the CRAN team to integrate C++ code with R is gratefully acknowledged.

License GPL (>= 3)

LazyLoad yes

URL <http://www.math.rug.nl/stat/Main/Research>

Repository CRAN

Date/Publication 2014-09-24 14:22:58

NeedsCompilation yes

R topics documented:

BDgraph-package	2
bdgraph	3
bdgraph.npn	5
bdgraph.sim	6
CellSignal	8
compare	9
geneExpression	10
I.g	11
phat	12
plot.bdgraph	13
plot.simulate	14
plotcoda	15
plotroc	16
print.bdgraph	17
print.simulate	18
prob	19
rgwish	20
select	21
summary.bdgraph	22
surveyData	23
traceplot	24
Index	26

BDgraph-package	<i>Graph selection based on birth-death MCMC</i>
-----------------	--

Description

The R package BDgraph provides a statistical tool for Bayesian structure learning in undirected graphical models based on birth-death MCMC methodology.

Details

This package provides an implementation of the procedures described in Mohammadi and Wit (2013). The main function is 'bdgraph' which is a birth-death MCMC algorithm for Bayesian inference in graphical models.

Functions:

bdgraph	Graph selection based on birth-death MCMC algorithm
bdgraph.sim	Synthetic graph data generator
bdgraph.npt	Nonparametric transfer
CellSignal	A flow cytometry dataset
compare	Comparing the result
geneExpression	Human gene expression dataset
I.g	Normalizing constant of G-Wishart distribution

phat	Posterior edge inclusion probabilities
plot.bdgraph	Plot function for "bdgraph" output
plotcoda	Convergence plot
plotroc	ROC plot
print.bdgraph	Print function for "bdgraph" output
prob	Posterior probabilities of the graphs
rgwish	Sampling from G-Wishart distribution
select	Selecting the best graph
surveyData	Labor force survey data
summary.bdgraph	Summary function for "bdgraph" output
traceplot	Trace plot of graph size from "bdgraph" output

Author(s)

Abdolreza Mohammadi <a.mohammadi@rug.nl> and Ernst Wit

References

- Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>
- Lenkoski, A. (2013). A direct sampler for G-Wishart variates, Stat 2, 119-128.
- Wang, H. and S. Li (2012). Efficient Gaussian graphical model determination under G-Wishart prior distributions. Electronic Journal of Statistics 6, 168-198.

bdgraph

Graph selection based on birth-death MCMC algorithm

Description

This function is the BD-MCMC algorithm for model selection in undirected Gaussian graphical models based on birth-death MCMC methodology.

Usage

```
bdgraph( data, n = NULL, method = "exact", iter = 5000, burnin = iter / 2,
         b = 3, D = NULL, Gstart = "empty" )
```

Arguments

- data** It could be a matrix or a data.frame of the data ($n \times p$) or it could be a covariance matrix as $S = x'x$ which x is the data matrix. It also could be an object of class "simulate".
- n** The number of observations. It is needed if the "data" is a covariance matrix.

method	A character with three options "exact" (default), "approx" and "copula" to determine a type of BD-MCMC algorithm. Option "exact" is a exact sampling from joint posterior distributions, without any approximation. Option "approx" is an approximation approach for sampling from joint posterior distributions. See first reference for more details. Option "copula" is for non-Gaussian, discrete, or mixed data sets.
iter	The number of iteration for the BD-MCMC algorithm.
burnin	The number of burn-in iteration for the BD-MCMC algorithm.
b	The degree of freedom for G-Wishart distribution, $W_G(b, D)$. G-Wishart distribution is the prior distribution of precision matrix. The default is 3.
D	The positive definite matrix for G-Wishart distribution, $W_G(b, D)$. G-Wishart distribution is the prior distribution of precision matrix. The default is an identity matrix.
Gstart	It is corresponding to a starting point for graph. It can be either "full" (default), "empty", or an object with S3 class "bdgraph". Option "full" means the initial graph is a full graph and "empty" means a empty graph. Gstart also could be an object with S3 class "bdgraph"; With this option we could run the bdgraph algorithm from last objects of previous bdgraph run (see examples).

Value

an object with S3 class "bdgraph" as follow:

sampleGraphs	A vector which includes the adjacency matrices for all iteration after burn-in.
graphWeights	A vector which includes the waiting times for all iteration after burn-in.
allGraphs	A vector which includes the adjacency matrices for all iteration (includes burn-in iteration). It is needed to check the convergency of the BD-MCMC algorithm.
allWeights	A vector which includes the waiting times for all iteration (includes burn-in iteration).
Khat	Estimation for precision matrix which is a mean of all samples from precision matrices.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph.sim](#)

Examples

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

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

summary(output)

# To compare our result with true graph
compare( data.sim, output, colnames = c("True graph", "BDgraph") )

output2 <- bdgraph( data = data.sim, iter = 5000, Gstart = output )

compare( data.sim, output, output2, colnames = c("True graph", "Frist run", "Second run") )

## End(Not run)
```

bdgraph.npn

Nonparametric transfer

Description

This function transfer non-Gaussian data to Gaussian models.

Usage

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

Arguments

data	It could be a matrix or a data.frame corresponding to the data ($n \times p$).
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, 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 (when npn = "truncation"). The default value is $1/(4n^{1/4}\sqrt{\pi \log(n)})$.

Value

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

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

- Liu, H., Han, F., Yuan, M., Lafferty, J., and Wasserman, L., (2012). High Dimensional Semiparametric Gaussian Copula Graphical Models. *Annals of Statistics*
- Zhao, T. and H. Liu. (2012). The huge Package for High-dimensional Undirected Graph Estimation in R. *Journal of Machine Learning Research*
- Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, *Bayesian Analysis*, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph.sim](#) and [bdgraph](#)

Examples

```
## Not run:
# generating synthetic 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 skeptic
bdgraph.npn( data, npn = "skeptic" )

## End(Not run)
```

bdgraph.sim

Synthetic graph data generator

Description

Implements a synthetic graph data generation with different graph structures, including "random", "cluster", "scale-free", "hub" "fixed", and "circle". The default is "random". Based on the graph structure, it generate four different data sets which is multivariate Gaussian, non-Gaussian, discrete, or mixed data.

Usage

```
bdgraph.sim( n = 2, p = 10, graph = "random", size = NULL, prob = 0.2,
             class = NULL, type = "Gaussian", cut = 4, b = 3, D = diag(p),
             K = NULL, sigma = NULL, mean = 0, vis = FALSE )
```

Arguments

n	The number of samples required. The default value is 2.
p	The number of variables (nodes). The default value is 10.
graph	The graph structure with option "random" (as a default), "cluster", "scale-free", "hub" "fixed", and "circle". It also could be adjacency matrix corresponding to the graph structure (an upper triangular matrix in which $g_{ij} = 1$ if there is a link between notes i and j , otherwise $g_{ij} = 0$).
size	The number of links in true graph (graph size).
prob	For "random" graph, it is the probability that a pair of nodes has a link. The default value is 0.2.
class	Is number of the classes for "cluster" option.
type	Type of data which has four options "Gaussian" (as a default), "non-Gaussian", "discrete", and "mixed". For option "Gaussian", data is generated from multivariate normal distribution. For option "non-Gaussian", data is transferred multivariate normal distribution to continuous multivariate non-Gaussian distribution. For option "discrete", data is transferred from multivariate normal distribution to discrete multivariate distribution. For option "mixed", data is transferred from multivariate normal distribution to mixture of 'count', 'ordinal', 'non-Gaussian', 'binary' and 'Gaussian', respectively.
cut	For simulated discrete data (type = "discrete"), it is number of categories for discrete variables. The default value is 4.
b	The degree of freedom for G-Wishart distribution, $W_G(b, D)$. The default is 3.
D	The positive definite ($p \times p$) "scale" matrix for G-Wishart distribution, $W_G(b, D)$. The default is an identity matrix.
K	A positive-definite symmetric matrix specifying the precision matrix of the variables. It is for "fixed" option of graph.
sigma	A positive-definite symmetric matrix specifying the covariance matrix of the variables. It is for "fixed" option of graph.
mean	A vector specifying the mean of the variables. The default value is zero.
vis	visualize the true graph pattern. The default value is FALSE.

Value

an object with S3 class "simulate" as follow:

data	Generated data which is a $n \times p$ 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)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph](#)

Examples

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

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

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

## End(Not run)
```

CellSignal

A flow cytometry dataset

Description

This data set contains flow cytometry of $p = 11$ proteins measured on $n = 11672$ cells.

Usage

```
data(CellSignal)
```

Format

The format is a list with 11672 rows (number of cells) and 11 column (number of proteins).

Source

Data are publicly available at <http://www.sciencemag.org/content/308/5721/523/suppl/DC1>

Examples

```
data( CellSignal )
head( CellSignal )
boxplot( CellSignal )
```

compare	<i>Comparing the result</i>
---------	-----------------------------

Description

With this function we can compare the result from BD-MCMC algorithm or other approaches, usually based to the true graph structure.

Usage

```
compare( G, est, est2 = NULL, colnames = NULL, vis = FALSE )
```

Arguments

G	The adjacency matrix corresponding to the true graph in which $G_{ij} = 1$ if there is a link between notes i and j , otherwise $G_{ij} = 0$. It also can be an object with S3 class "simulate".
est	Adjacency matrix corresponding to the estimated graph. It also can be an object with S3 class "bdgraph" or "select" (from huge package).
est2	Adjacency matrix corresponding to the estimated graph. It also can be an object with S3 class "bdgraph" or "select" (from huge package).
colnames	A character vector giving the column names for the result table.
vis	Logical: if TRUE you will see the plot result. The default is FALSE.

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.
Accuracy	the number of true results (both true positives and true negatives) divided by the total number of true and false results.
Balanced F-score	a weighted average of the "positive predictive" and "true positive rate". F-score value reaches its best value at 1 and worst score at 0.
Positive predictive	the number of correctly estimated links divided by the total number of links in the estimated graph.
True positive rate	the number of correctly estimated links divided by the total number of links in the true graph.
False positive rate	the false positive value divided by the total number of links in the true graph.

Author(s)

Abdolreza Mohammadi, Antonio Abbruzzo, Ivan Vujacic, and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph](#) and [select](#)

Examples

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

# selecting the best graph according to BD-MCMC algorithm
output <- bdgraph( data = data.sim, iter = 10000 )
# comparing the result
compare( data.sim, output, colnames = c("true", "BDgraph"), vis = TRUE )

# selecting the best graph by 'huge' package
require( huge )
huge.g <- huge( data.sim $ data, method = "mb" )
huge.g <- huge.select( huge.g )

# comparing the result
compare( data.sim, output, huge.g, colnames = c("true", "BDgraph", "huge") , vis = T )

## End(Not run)
```

geneExpression

Human gene expression dataset

Description

This data set contains human gene expression of $p = 100$ transcripts (with unique Illumina TargetID) measured on $n = 60$ unrelated individuals.

Usage

```
data(geneExpression)
```

Format

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

Source

The genotypes of those 60 unrelated individuals are available from the Sanger Institute website at <ftp://ftp.sanger.ac.uk/pub/genevar>

Examples

```
data( geneExpression )
dim( geneExpression )
head( geneExpression )
boxplot( geneExpression )
```

I.g

*Normalizing constant of G-Wishart distribution***Description**

Monte Carlo method for approximating the normalizing constant of G-Wishart distribution. The function uses the Monte Carlo method of Atay-Kayis and Massam (2005).

Usage

```
I.g( G, b = 3, D = diag( ncol(G) ), mc = 100 )
```

Arguments

- | | |
|----|---|
| G | Adjacency matrix corresponding to the graph structure. It is an upper triangular matrix in which $g_{ij} = 1$ if there is a link between notes i and j , otherwise $g_{ij} = 0$. |
| b | The degree of freedom for G-Wishart distribution, $W_G(b, D)$. The default is 3. |
| D | The positive definite ($p \times p$) "scale" matrix for G-Wishart distribution, $W_G(b, D)$. The default is identity matrix. |
| mc | The number of iteration for the Monte Carlo approximation. The default is 100. |

Details

Normalizing constant approximation using Monte Carlo method for a G-Wishart(b,D):

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

Value

the normalizing constant of G-Wishart distribution.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

Atay-Kayis, A. and H. Massam (2005). A monte carlo method for computing the marginal likelihood in nondecomposable Gaussian graphical models. Biometrika 92(2), 317-335.

Examples

```
G <- matrix( c(0,0,1,
              0,0,1,
              0,0,0), 3, 3, byrow = TRUE )

# matrix G shows a graph with 3 nodes and 2 links
I.g( G, b = 3, D = diag(3) )
```

phat	<i>Posterior edge inclusion probabilities</i>
------	---

Description

According to the output of BD-MCMC algorithm, this function gives us the posterior edge inclusion probabilities for all possible edges.

Usage

```
phat( output, round = 3 )
```

Arguments

output	An object with S3 class "bdgraph".
round	A number for rounding all probabilities to the specified number of decimal places (default 3).

Value

phat	Upper triangular matrix which shows the posterior inclusion probabilities for all possible edges.
------	---

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph](#) and [bdgraph.sim](#)

Examples

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

output <- bdgraph( data = data.sim )

phat( output, round = 2 )

## End(Not run)
```

plot.bdgraph

Plot function for "bdgraph" output

Description

According to the output of our BD-MCMC algorithm, this function gives us the plot.

Usage

```
## S3 method for class 'bdgraph'
plot( x, g = 1, layout = layout.circle, ... )
```

Arguments

x	An object with S3 class "bdgraph".
g	The number of graphical models with highest probabilities (default is 1).
layout	The vertex placement algorithm which is according to igraph package. The default is "layout.circle".
...	System reserved (no specific usage).

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph](#)

Examples

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

output <- bdgraph( data = data.sim )

plot(output)

plot( output, g = 4 )

## End(Not run)
```

plot.simulate

Plot function for "bdgraph.sim" output

Description

Visualize the structure of true graph.

Usage

```
## S3 method for class 'simulate'
plot( x, main = NULL, layout = layout.circle, ... )
```

Arguments

x	An object with S3 class "simulate".
main	Graphical parameter (see plot).
layout	The vertex placement algorithm which is according to igraph package. The default is "layout.circle".
...	System reserved (no specific usage).

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph](#)

Examples

```
## Not run:  
# generating synthetic multivariate normal data from a 'random' graph  
data.sim <- bdgraph.sim( n = 10, p = 15 )  
  
plot( data.sim )  
  
## End(Not run)
```

plotcoda *Convergence plot*

Description

This function according to "bdgraph" output gives us several plots. These plots can be used to check the convergence of the BD-MCMC algorithm.

Usage

```
plotcoda(output, thin = NULL, trace = TRUE, main = NULL, ...)
```

Arguments

output	An object with S3 class "bdgraph"
thin	An option for getting fast result for a cumulative plot according to part of the iteration.
trace	Logical: if TRUE (default), tracing information is printed.
main	Graphical parameter (see plot).
...	System reserved (no specific usage).

Details

Note that a spending time for this function depends on the graph. It should be slow for the high-dimensional graphs. To make it faster you can choose bigger value for 'thin'.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph](#)

Examples

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

output <- bdgraph( data = data.sim )

plotcoda(output)

## End(Not run)
```

plotroc

ROC plot

Description

Plot the ROC curve according to the true graph structure for the "bdgraph" output.

Usage

```
plotroc( G, prob, prob2 = NULL, cut = 20, smooth = FALSE )
```

Arguments

G	The adjacency matrix corresponding to the true graph structure in which $G_{ij} = 1$ if there is a link between notes i and j , otherwise $G_{ij} = 0$. It also can be an object with S3 class "simulate".
prob	Upper triangular matrix which shows the stimated posterior inclusion probabilities for all possible edges. It also can be an object with S3 class "bdgraph".
prob2	This option is for comparing two different estimations. Upper triangular matrix which shows the stimated posterior inclusion probabilities for all possible edges. It also can be an object with S3 class "bdgraph".
cut	Number of cut point. Default is 20.
smooth	Logical: for smoothing the ROC curve. The default is FALSE.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph](#) and [compare](#)

Examples

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

# selecting the best graph according to BD-MCMC algorithm
output <- bdgraph( data = data.sim, iter = 10000 )
# comparing the result
plotroc( data.sim, output )

plotroc( data.sim, output, smooth = TRUE )

## End(Not run)
```

print.bdgraph

Print function for "bdgraph" output

Description

According to output of the BD-MCMC algorithm, this function gives us the print of the result.

Usage

```
## S3 method for class 'bdgraph'
print( x, round = 3, Khat = FALSE, phat = FALSE, ... )
```

Arguments

x	An object with S3 class "bdgraph".
round	A number for rounding all probabilities to the specified number of decimal places (default 3).
Khat	Logical: if TRUE you will see the estimation of precision matrix. The default is FALSE.
phat	Logical: if TRUE you will see the posterior edge inclusion probabilities. The default is FALSE.
...	System reserved (no specific usage).

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph](#)

Examples

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

output <- bdgraph( data = data.sim )

print(output)

print( output, Khat = TRUE, phat = TRUE )

## End(Not run)
```

print.simulate	<i>Print function for "bdgraph.sim" output</i>
----------------	--

Description

Print the sample size, the number of nodes, number of links, type and sparsity of the true graph.

Usage

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

Arguments

x	An object with S3 class "simulate".
...	System reserved (no specific usage).

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph.sim](#)

Examples

```
## Not run:
# generating synthetic multivariate normal data from a 'random' graph
data.sim <- bdgraph.sim( n = 10, p = 8, vis = TRUE )

print(data.sim)

## End(Not run)
```

prob	<i>Posterior probabilities of the graphs</i>
------	--

Description

According to the output of the BD-MCMC algorithm, this function gives the posterior probabilities for the 'g' most likely graphs or a specific graph, 'G'.

Usage

```
prob( output, g = 4, G = NULL )
```

Arguments

output	An object with S3 class "bdgraph".
g	The number of graphs with the highest probabilities to be shown (default is 4). This option is ignored if 'G' is specified.
G	Adjacency matrix corresponding to a graph structure. It is an upper triangular matrix in which $G_{ij} = 1$ if there is a link between notes i and j , otherwise $G_{ij} = 0$. It also can be an object with S3 class "simulate".

Value

best.G	The graphs with the highest posterior probabilities.
prob.G	A vector which includes posterior probabilities of the graphs in 'best.G'.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also[bdgraph](#)**Examples**

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

output <- bdgraph( data = data.sim )

# Estimated posterior probability of the true graph
prob( output, G = data.sim )

# Estimated posterior probability of the first and second graphs with highest probabilities
prob( output, g = 2 )

## End(Not run)
```

rgwish

*Sampling from G-Wishart distribution***Description**

Generate n random matrices, distributed according to the G-Wishart distribution with parameters b and D, $W_G(b, D)$.

Usage

```
rgwish( n = 1, G = NULL, b = 3, D = NULL )
```

Arguments

n	The number of samples required. The default is 1.
G	Adjacency matrix corresponding to the graph structure. It is an upper triangular matrix in which $G_{ij} = 1$ if there is a link between notes i and j , otherwise $G_{ij} = 0$.
b	the degree of freedom for G-Wishart distribution, $W_G(b, D)$. The default is 3.
D	the positive definite ($p \times p$) "scale" matrix for G-Wishart distribution, $W_G(b, D)$. The default is an identity matrix.

Details

Sampling from G-Wishart(b,D) distribution with density:

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

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 $W_G(b, D)$.

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

Lenkoski, A. (2013). A direct sampler for G-Wishart variates, Stat 2, 119-128.

Examples

```
## Not run:
G <- toeplitz( c( 0, 1, rep( 0, 3 ) ) )
G # graph with 5 nodes and 4 links

sample <- rgwish( n = 3, G = G, b = 3, D = diag(5) )
sample

## End(Not run)
```

 select

Selecting the best graph

Description

According to output of the BD-MCMC algorithm, this function gives the graphs with the highest posterior probabilities. For more specific selection of graphs consult the 'prob' function.

Usage

```
select( output, vis = FALSE )
```

Arguments

`output` An object with S3 class "bdgraph".

`vis` Logical: if TRUE you will see the plot of best graph. The default is FALSE.

Value

`G` Adjacency matrix corresponding to the best graph (graph with the highest posterior probability).

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph](#)

Examples

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

output <- bdgraph( data = data.sim )

select(output)

## End(Not run)
```

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

Description

This function gives us a summary of the result from BD-MCMC algorithm according to output from "bdgraph" function.

Usage

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

Arguments

object	An object with S3 class "bdgraph".
vis	Logical: if TRUE (default) you will see the plot result.
layout	The vertex placement algorithm which is according to igraph package. The default is "layout.circle".
...	System reserved (no specific usage).

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph](#)

Examples

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

output <- bdgraph( data = data.sim )

summary(output)

summary( output, vis = FALSE )

## End(Not run)
```

surveyData

Labor force survey data

Description

This survey data concerns 1002 males in the U.S labor force, described in Hoff (2007). The seven observed variables which have been measured on various scales are as follow: the income (income), degree (degree), 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.

Examples

```
data( surveyData )
dim( surveyData )
head( surveyData )
boxplot( surveyData )
```

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

Description

Trace plot of graph size from all bdgraph output. It is a tool for checking the convergency of the BD-MCMC algorithm.

Usage

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

Arguments

output	An object with S3 class "bdgraph".
acf	Logical: if TRUE you will see the plot of autocorrelation functions for graph sizes. The default is FALSE.
pacf	Logical: if TRUE you will see the plot of partial autocorrelations for graph sizes. The default is FALSE.
main	Graphical parameter (see plot).
...	System reserved (no specific usage).

Author(s)

Abdolreza Mohammadi and Ernst Wit

References

Mohammadi, A. and Wit, E. C. (2014). Bayesian structure learning in sparse Gaussian graphical models, Bayesian Analysis, accepted. <http://arxiv.org/abs/1210.5371v6>

See Also

[bdgraph](#)

Examples

```
## Not run:  
# generating synthetic multivariate normal data from a 'random' graph  
data.sim <- bdgraph.sim( n = 50, p = 6, size = 7, vis = TRUE )  
  
output <- bdgraph( data = data.sim, iter = 10000 )  
  
traceplot(output)  
  
traceplot( output, acf = TRUE, pacf = TRUE )  
  
## End(Not run)
```

Index

*Topic **datasets**

CellSignal, 8
geneExpression, 10
surveyData, 23

*Topic **package**

BDgraph-package, 2

BDgraph (BDgraph-package), 2
bdgraph, 3, 6, 8, 10, 13–15, 17, 18, 20, 22–24
BDgraph-package, 2
bdgraph.npn, 5
bdgraph.sim, 4, 6, 6, 13, 19

CellSignal, 8
compare, 9, 17

geneExpression, 10

I.g, 11

phat, 12
plot.bdgraph, 13
plot.simulate, 14
plotcoda, 15
plotroc, 16
print.bdgraph, 17
print.simulate, 18
prob, 19

rgwish, 20

select, 10, 21
summary.bdgraph, 22
surveyData, 23

traceplot, 24